

Received December 12, 2021, accepted December 23, 2021, date of publication January 4, 2022, date of current version January 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140373

ConvAE-LSTM: Convolutional Autoencoder Long Short-Term Memory Network for Smartphone-Based Human Activity Recognition

DIPANWITA THAKUR¹, (Member, IEEE), SUPARNA BISWAS², (Senior Member, IEEE), EDMOND S. L. HO³, AND SAMIRAN CHATTOPADHYAY^{4,5}, (Senior Member, IEEE)

¹Department of Computer Science, Banasthali Vidyapith, Jaipur, Rajasthan 304022, India

²Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, Kolkata, West Bengal 700064, India

³Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K.

⁴Institute for Advancing Intelligence, TCG Centres for Research and Education in Science and Technology, Kolkata 700091, India

⁵Department of Information Technology, Jadavpur University, Kolkata 700032, India

Corresponding author: Samiran Chattopadhyay (samirancju@gmail.com)

This work was supported by the Royal Society International Exchanges Grant under Grant IES/R1 191147.

ABSTRACT The self-regulated recognition of human activities from time-series smartphone sensor data is a growing research area in smart and intelligent health care. Deep learning (DL) approaches have exhibited improvements over traditional machine learning (ML) models in various domains, including human activity recognition (HAR). Several issues are involved with traditional ML approaches; these include handcrafted feature extraction, which is a tedious and complex task involving expert domain knowledge, and the use of a separate dimensionality reduction module to overcome overfitting problems and hence provide model generalization. In this article, we propose a DL-based approach for activity recognition with smartphone sensor data, i.e., accelerometer and gyroscope data. Convolutional neural networks (CNNs), autoencoders (AEs), and long short-term memory (LSTM) possess complementary modeling capabilities, as CNNs are good at automatic feature extraction, AEs are used for dimensionality reduction and LSTMs are adept at temporal modeling. In this study, we take advantage of the complementarity of CNNs, AEs, and LSTMs by combining them into a unified architecture. We explore the proposed architecture, namely, “ConvAE-LSTM”, on four different standard public datasets (WISDM, UCI, PAMAP2, and OPPORTUNITY). The experimental results indicate that our novel approach is practical and provides relative smartphone-based HAR solution performance improvements in terms of computational time, accuracy, F1-score, precision, and recall over existing state-of-the-art methods.

INDEX TERMS Deep Learning, human activity recognition, smartphone sensors, autoencoder.

I. INTRODUCTION

Human activity recognition (HAR) has been a popular research area for several decades due to its wide applications in smart health care, ambient assisted living, disease prediction, video surveillance, remote health care and so on [1], [2]. According to a report released by the United Nations (UN) [3], the expected worldwide population of elderly people is expected to reach 2 billion by 2050. Elderly

people need special attention and care, as most elderly people suffer from many diseases. Moreover, the doctor-to-patient population ratio was determined to be 1:1800. The monitoring of real-time human physical activities, particularly the daily living activities (DLAs) [4] of elderly people, is an indispensable aspect in smart health care and can effectively enhance medical rehabilitation and elderly care. Daily lifestyles have significant impacts on several critical diseases. Therefore, daily physical activity monitoring provides an important health indicator [5]. The identification and classification of human physical activity are popularly used to

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato¹.

monitor, analyze and understand various postures across a variety of applications and systems.

Various sensor-based HAR frameworks have been proposed in the literature, such as smartphone sensor-based, body-worn sensor-based and audio/video data-based frameworks. Body-worn sensors are not comfortable for users, and audio/video data have several privacy concerns. Moreover, both body-worn sensor signals and audio/video signals require complex signal processing techniques [6]. The audio recording of any long-term activity becomes noisy due to background noise or white noise. Therefore, an audio signal alone at any given moment does not provide valuable information. It may also be difficult to differentiate between two pieces of audio information. Therefore, audio data are insufficient on their own for recognizing some basic activities. The collection of video data turns out to be difficult in populated locations, in locations where many physical obstacles exist, or when brightness is low [7]. To infer the descriptions of human behaviors and transport modes, sensor data are also obtained by using smartphones. In human activity monitoring and understanding, the utility of tactile information provided by smartphones affects analyses because of their distinct centers of attention over other sensor modalities. Normally, smartphone sensor-based physical HAR systems are motivated by their ubiquity, discretion, inexpensive installation procedures, noninvasive properties and ease of use [8]. By utilizing a smartphone, continuous data can be collected while performing any type of physical activity. Moreover, mobile health-related data monitoring becomes more elegant and accurate due to the variety of built-in smartphone sensors [9]. Several built-in smartphone sensors can be used to collect data for HAR. However, the most commonly used built-in sensors are accelerometers and gyroscopes [10]–[13].

Smartphone sensor-based datasets contain multivariate time-series data. Local dependency is the intrinsic nature of time-series data. Moreover, the natures of human activity signals are hierarchical and translation-invariant [14], and they are rich in dynamic information regarding the underlying system. As a result, the need to accurately model such high-dimensional datasets is increasing. Physical activities consist of some special distinctive features. Hence, several methodological challenges are involved in HAR (except for handcrafted feature extraction), such as imbalanced datasets, intraclass variability, interclass similarity, the empty class problem [15] and the multiclass window problem [16].

Recently, smartphone-based HAR systems utilizing conventional machine learning (ML) algorithms or deep learning (DL) approaches have gained popularity [17]. Feature engineering is a dominant phase in traditional ML methods because it extracts the relevant features that are responsible for differentiating various activity patterns. The accurate performance of HAR solutions greatly depends on the feature engineering of raw signals [18]. Then, the features are fed to classifiers to recognize human activities [6], [19]. Without an adequate feature engineering process, conventional classifiers fail to competently and accurately identify physical

activities. Hence, to provide sensory data in an appropriate form, complex data preprocessing techniques are required, and handcrafted features are extracted from the acquired sensory data based on expert domain knowledge [11]. Finally, the handcrafted feature vector is fed to the conventional classifiers to recognize various human physical activities. However, past research has shown that some of these handcrafted features are good at distinguishing one activity but not as good at recognizing others [20]. Moreover, different research domains require different handcrafted feature vectors to properly handle classification problems.

A. MOTIVATION

Automatic feature learning capabilities make DL algorithms more popular than conventional ML algorithms. DL algorithms can extract relevant features efficiently without any manual assistance while simultaneously identifying human activities [21]–[23]. DL approaches have proven their outstanding predictive capabilities in speech and image recognition, intelligent gamification, and natural language processing. In the HAR literature, numerous DL approaches have been investigated and applied. Convolutional neural networks (CNNs) are popular supervised DL methods that are used in the HAR domain (see Figure 1) to overcome the problem of handcrafted feature extraction. CNN layers are used to automatically extract features from raw time-series data without human intervention. In a CNN, the number of feature maps often increases with the network depth, causing the computational complexity of the architecture to also increase. To overcome this problem, a dimensionality reduction technique can be employed to reduce the number of feature maps. Moreover, existing DL-based HAR systems require large quantities of labeled training data to achieve good performance. However, in a real scenario, a large volume of labeled training data is not easy to obtain because the task of creating such a volume is tedious, time-consuming, laborious, and expensive. Moreover, in real-time HAR applications, the availability of labeled data is quite poor. To overcome these issues, in this research work, we take advantage of an ‘autoencoder (AE), which possesses the property of unsupervised feature learning and enables dimensionality reduction with convolutional layers [24].

A convolutional AE (ConvAE) is a type of AE in which nonlinear transformation is performed by a CNN [25]. This is the motivation behind the use of a combination of a CNN and an AE in our proposed architecture. Convolutional layers are less computationally complex than connected layers [26]. Since CNNs primarily work in vector spaces, learning the high-dimensional properties of input time series is more difficult. As a result, CNN architectures alone cannot efficiently predict time-series signal data [27]. Moreover, CNNs cannot extract temporal features, leading to a reduction in activity recognition accuracy. ‘Long short-term memory (LSTM) networks are adept at sequential learning because they carry signal information across time steps [27]. By leveraging the complementary strengths of a CNN and an LSTM neural

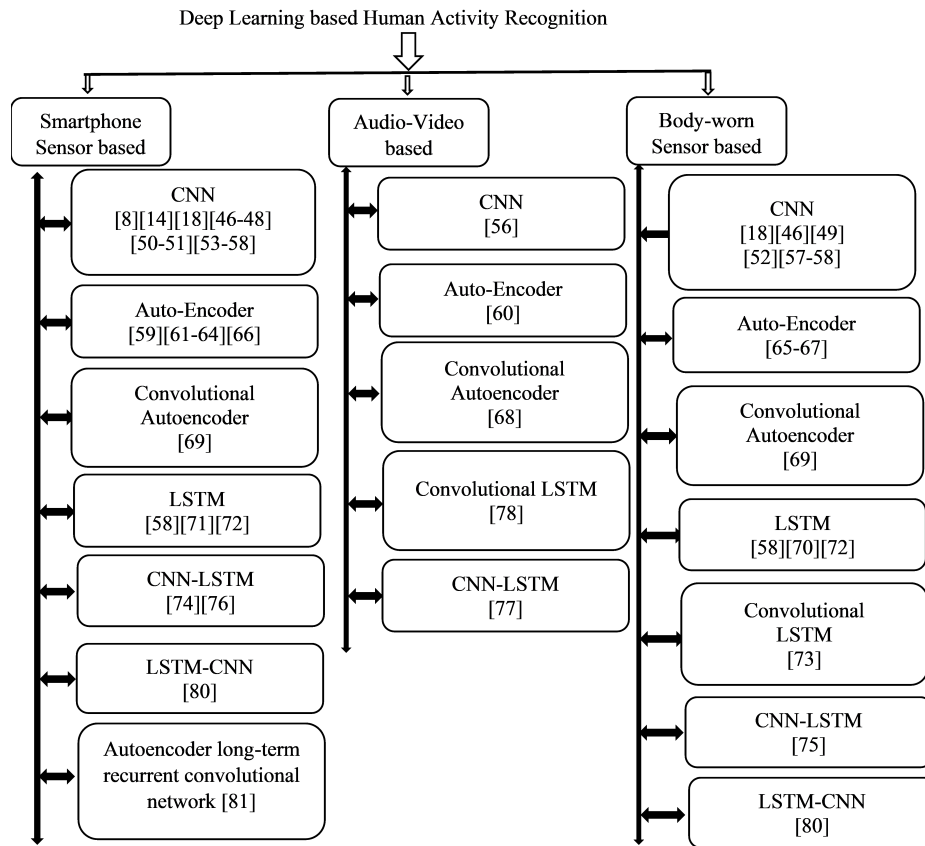


FIGURE 1. Taxonomy of DL-based HAR.

network, the combination of CNN and LSTM models preserves both spatial and temporal information and performs better in terms of sequential learning [27], [28].

Motivated by the architectures of CNNs, AEs, and LSTM networks, this work proposes an integrated architecture using ConvAE and LSTM for recognizing human activities. Our exhaustive literature study also suggests that this combination is novel. The end-to-end model in the proposed work consists of the following three distinct modules.

- A convolutional AE module.
- An LSTM module.
- A fully connected layer followed by a softmax function.

B. CONTRIBUTIONS

The contributions provided in this paper are as follows.

- 1) We perform an extensive literature survey regarding DL-based HAR, which can be helpful for readers to understand and compare the state-of-the-art methods in this domain.
- 2) We propose ConvAE-LSTM, which is a novel DL architecture that (a) can automatically extract features from unlabeled raw sensory data, (b) uses fewer parameters due to the presence of a convolution layer that minimizes the risk of overfitting, (c) reduces the

required computational time and d) enhances the accuracy of HAR.

- 3) We demonstrate the effectiveness of our proposed ConvAE-LSTM network through empirical experiments on two different standard public smartphone sensor-based HAR datasets the same experimental environment.
- 4) Additionally, we compare the experimental results obtained using our proposed method with those of some state-of-the-art methods drawn from the HAR literature.

II. RELATED WORK

A substantial number of reviews have been conducted regarding the recognition of human physical activities using various approaches, as elucidated in [11], [29], [30], which include both ML and DL approaches. Various shallow machine learning approaches have been used in HAR solutions. For instance, in [6], [12], [13], [31]–[36], the authors emphasized only accuracy based on various ML algorithms by using 17 different baseline time-frequency domain features mentioned in [11]. Previously, the researchers in [29], [32], [37]–[45] used various feature extraction and/or selection methods before feeding the obtained data to a classification algorithm for recognizing diverse human activities.

ML models rely on handcrafted features in the HAR domain, and such features require expert domain knowledge. Moreover, they increase the time complexity of the resulting model. To overcome these issues, researchers have started exploring DL approaches, as DL models possess automatic feature extraction capabilities. In this section, we establish an underlying basis for DL-based HAR while looking at some of the earlier works that are relevant to our approach and how their methods differ from ours.

A. DEEP LEARNING FOR HAR

Relevant features can be automatically learned by DL algorithms. As a result, in the case of smartphone-based HAR models, the performance of DL algorithms is astonishingly high. Because of their hierarchical feature extraction capacities, CNNs are gaining prominence in the HAR domain. CNNs containing at least one convolutional layer and one pooling layer followed by at least one fully connected layer have acquired fame because of their capability of learning unique representations from images or speech while capturing local dependencies and distortion invariance [8]. In [18], the authors proposed a CNN-based HAR model by capturing the local dependencies and scale-invariant features of activity signals. To prove the efficiency of the proposed framework, the authors used three public datasets: OPPORTUNITY [82], [83], Skoda [84], and Actitracker [85]. Ronao and Cho [14] proposed another CNN method for HAR by using the UCI public dataset and handcrafted features. Another work proposed in [8] used a 1D-CNN to recognize human activities in the UCI public dataset [11] with extra temporal fast Fourier transformation (FFT) information.

Ravi *et al.* [46] presented a HAR framework using convolutional layers and shallow features obtained from smartphone sensors and wearable sensors. The WISDM dataset [10] and DAPHNet-FoG [86] datasets were used in this study. Bevilacqua *et al.* [87] suggested a CNN-based HAR that uses five distinct sensors, including an accelerometer and a gyroscope, to recognize 16 different lower-limb actions. In another work, Jiang *et al.* [47] described a CNN-based HAR framework using the UCI-HAR public dataset. The adaptive moment estimation (Adam) hyperparameter optimization technique was employed in this study. Another HAR framework proposed by Ignatov [48] used a combination of manually extracted features and automatically extracted features obtained from a CNN. To demonstrate the efficiency of the proposed framework, the authors used two popular public datasets (WISDM and UCI). They also experimented without extracting handcrafted features from the UCI dataset. A body-worn sensor-based HAR framework was proposed by Rueda *et al.* [49] using a CNN. Three different datasets were used in their experiments to prove the efficiency of the proposed model; two different public datasets, OPPORTUNITY and PAMAP2, were used in [88]. In [50], the authors proposed an HAR framework using a 2D-CNN and calculated both

the accuracy and computational time of the developed approach.

In [51], the authors suggested a HAR framework using a CNN architecture for the “UCI-HAR” public dataset. Moreover, the authors calculated the training and testing times of their approach as 3.4274 seconds and 372.6 ms, respectively. Zebin *et al.* [52] proposed an HAR model to recognize five different activities such as “walking on a level surface”, “walking upstairs”, “walking downstairs”, “remaining sedentary” and “sleeping” by utilizing a CNN. Waist-mounted inertial sensors such as an accelerometer and a gyroscope were used to collect the data. In [53], the authors proposed a CNN-based HAR solution using smartphone accelerometer, magnetometer, gyroscope, and barometer sensor data. In this work, the authors identified nine different activities. In [54], the authors proposed a 2D deep CNN architecture to solve an HAR problem. In this work, the authors used a separate data compression technique for smartphone sensor data. Gamble and Huang [55] described a 1D-CNN architecture with accelerometer and gyroscope smartphone sensors for HAR to identify human physical activities. Cruciani *et al.* [56] proposed a smartphone sensor-based and audio-based HAR method using a CNN. They used the UCI-HAR dataset, a real-world extrasensory dataset [89] and the DCASE 2017 dataset. Yen *et al.* [57] suggested a CNN-based HAR framework using the smartphone-derived UCI-HAR public dataset and self-collected data from wearable sensors. Wan *et al.* [58] suggested an HAR framework incorporating three different deep learning methods, namely, a CNN, LSTM, and bidirectional LSTM, with two different public datasets; one included the smartphone sensor-based UCI datasets, and the other was derived from the wearable sensor-based PAMAP2 datasets.

In [59], a sparse AE (SAE) was used to automatically learn features, and based on this concept, a smartphone-based HAR framework was proposed. Three different channels (an accelerometer, a gyroscope, and the magnitudes of both sensors) were used by the authors. Statistical metrics were used to demonstrate the achieved performance measures. In [60], the authors proposed an AE-based HAR system built on various video datasets. Utilizing a stacked autoencoder, a smartphone sensor-based HAR system was proposed by Almaslakh *et al.* [61]. In [62], the authors identified eight locomotion and transportation activities via an adversarial AE. Data were collected by using smartphone sensors with four different positions (torsos, bags, hips, and hands) to perform the experiment. Via a combination of stacking denoising AEs and LightGBM, an HAR solution was proposed in [63] using the UCI dataset. Ozcan and Basturk [64] proposed a stacked AE-based HAR system. The authors used both WISDM and UCI smartphone-based sensor data to perform their experiments. Recently, a regularized AE-based HAR framework was proposed in [65] using body-worn sensors. Based on the idea that one encoder is associated with one class, an ensemble of autonomous AE-based HAR solutions was proposed in [66]. In this study, the authors used three

TABLE 1. State-of-the-art DL-based HAR systems.

Reference	Sensor	Dataset	Classifier	Accuracy
[18]	Body-worn Body-worn Smartphone	OPPORTUNITY SKODA ACTITRACKER	CNN	88.19% 76.83% 96.88%
[14]	Smartphone	UCI	CNN CNN+Handcrafted	94.79% 95.75%
[8]	Smartphone	UCI	CNN CNN+FFT	94.79% 95.75%
[46]	Smartphone Body-worn	WISDM DAPHNet-FoG	CNN+Handcrafted CNN+Handcrafted	98.6% 95.8%
[47]	Smartphone	UCI	CNN	97.5%
[48]	Smartphone	WISDM UCI UCI	CNN CNN CNN+Handcrafted	90.42% 94.35% 95.32%
[49]	Body-worn Body-worn	OPPORTUNITY PAMAP2	CNN	92.24% 92.55%
[50]	Smartphone	UCI	CNN	95.69% 0.9063 s (Computational Time)
[51]	Smartphone	UCI	CNN	93.93% 3.4275 s (Training Time) 372.6 ms (Testing Time)
[52]	Body-worn (Acc.+Gyro.)	Self-collected	CNN	96.4%
[53]	Smartphone	Self-collected	CNN	98% 2s (Computational Time)
[54]	Smartphone (Acc.+Gyro.+Mag.)	Self-collected	CNN	95.2%
[55]	Smartphone (Acc.+Gyro.)	MotionSense	CNN	96.77%
[56]	Smartphone Smartphone+Smartwatch Audio	UCI Extrasensory DCASE 2017	CNN	67.51% 91.98% 92.30%
[57]	Smartphone Body-worn	UCI Self-collected	CNN	95.99% 97.19%
[58]	Smartphone Body-worn	UCI PAMAP2	CNN LSTM Bidirectional LSTM	93.21% 91% 89.14% 85.86% 89.94% 89.52%
[59]	Smartphone	Self-collected	SAE	97.55%
[60]	Video Data	KTH UCF11 VIRAT TRFCVID	AE	96.6% 59.73% 54.20% 63.75%
[61]	Smartphone	UCI	Stacked AE	97.5% 0.0375 ms (Computational Time)
[62]	Smartphone	SHL	Adversarial AE	90%
[63]	Smartphone	UCI	Stacking and denoising AE+LightGBM	95.99%
[64]	Smartphone	UCI WISDM	Stacked AE	0.0421(Error) 0.2104(Error)
[65]	Body-worn	OPPORTUNITY PAMAP2 DSA	Regularized AE	69.70%
[66]	Smartphone Body-worn Body-worn	WISDM MHealth PAMAP2	Ensemble of AE	80.8% 94.8% 56%
[67]	Body-worn	PAMAP2 OPPORTUNITY USC-HAD DAPHNet	AE	0.94 (F1-score) 0.43 (F1-score) 0.68 (F1-Score) 0.72 (F1-Score)
[68]	Video	KTH	Convolutional AE	92.49%
[69]	Smartphone Body-worn	WISDM OPPORTUNITY	Convolutional AE	94.9% 84.9%
[70]	Body-worn	Self-collected	Multilayer LSTM	92%
[71]	Smartphone	UCI	Bidirectional LSTM	93.79%

TABLE 1. (Continued.) State-of-the-art DL-based HAR systems.

[72]	Smartphone Body-worn	UCI OPPORTUNITY	Residual Bidirectional LSTM	93.6% 90.5%
[73]	Body-worn	OPPORTUNITY Skoda	Deep Convolutional LSTM	93% (F1-Score) 95.8% (F1-Score)
[74]	Smartphone	UCI	CNN-LSTM	93.40%
[75]	Body-worn	HAPT	CNN-LSTM	95.87%
[76]	Smartphone	UCI	CNN-LSTM	92.13%
[77]	Video	CAD-60	CNN-LSTM	97% (Precision) 98% (Recall)
[78]	Video	HMDB51 UCF101	Convolutional LSTM	69.4% 93.9%
[79]	Smartphone Body-worn Body-worn	UCI PAMAP2 OPPORTUNITY	InnoHAR	94.5% 93.5% 94.6%
[80]	Smartphone Smartphone Body-worn	UCI WISDM OPPORTUNITY	LSTM-CNN	95.78% 95.85% 92.63%
[81]	Smartphone	Self-collected	AE-based long- term recurrent convolutional network	97.4%

different datasets : WISDM, MHealth and PAMAP2. A typical AE-based HAR system was proposed in [67] using body-worn sensor data such as the PAMAP2, OPPORTUNITY, USC-HAD, and DAPHNet datasets.

Geng and Song [68] proposed an HAR solution using video data (KTH dataset). In this study, the authors used a CNN with a convolutional AE. Varamin *et al.* [69] proposed a deep convolutional AE-based approach to identify human activities using both a smartphone sensor and body-worn sensors with matching ratios of 94.9% and 84.9%, respectively. In this study, the authors emphasized only unsupervised feature learning concepts.

A context-aware HAR framework was proposed in [70]. A ‘multilayer LSTM with batch normalization was used by the authors to recognize static and dynamic physical activities using body-worn inertial sensors. However, the computational cost and memory requirements were quite high, as edge computing was used in this study. Yu and Qin [71] suggested an HAR framework using bidirectional LSTM with the UCI-HAR dataset. Zhao *et al.* [72] suggested a residual bidirectional LSTM architecture to identify different human activities using both UCI smartphone sensor and body-worn sensor (OPPORTUNITY) datasets.

In [73], the authors proposed CNN- and LSTM-based HAR solutions using two different public datasets collected by wearable sensors. In [74], the authors used a CNN followed by an LSTM-based DL architecture for HAR by using the UCI smartphone-based dataset. Wang *et al.* [75] suggested a HAR framework using a CNN in combination with LSTM. The HAPT public wearable sensor dataset was used by the authors to prove the effectiveness of their work. Mutegeki and Han [76] used the UCI smartphone sensor dataset to identify human activities using a CNN-LSTM architecture. Ercolano and Rossi [77] proposed a CNN-LSTM-based architecture using video data (the CAD-60 dataset) for HAR. In all

the aforementioned works, researchers used combinations of CNNs and LSTM to extract spatial and temporal features.

Ye *et al.* [78] suggested a two-stream convolutional network-based “convolutional LSTM” architecture to recognize various daily life activities. They used the HMDB51 and UCF101 video datasets and extracted features by using the convolution layer of the CNN.

Xia *et al.* [80] suggested an LSTM-CNN-based HAR framework to identify different daily life activities using three different datasets: UCI, WISDM, and OPPORTUNITY. In this study, a two-layer LSTM network, followed by a convolutional layer, was used. Two additional layers, global average pooling (GAP) and the batch normalization layers, were used to model parameters and speed up the convergence of the network, respectively. After convolution, the fully connected layer was replaced by a GAP layer.

Zou *et al.* [81] proposed an AE long-term recurrent convolutional network (AE-LRCN)-based HAR framework that consists of three different modules: an AE, a CNN, and LSTM. The proposed framework can identify five different activities : emptying, sitting, walking, running and standing.

Xu *et al.* [79] suggested “InnoHAR”, which is the combination of an inception neural network and an RNN with different scale-based convolution kernels, for HAR. Karim *et al.* [90] suggested ‘the use of multivariate LSTM-FCNs for time-series classification. The authors used an HAR dataset that included 34 other datasets obtained from different domains to demonstrate the performance of the proposed model. In this work, a squeeze-and-excitation block was incorporated to improve the performance of the proposed model.

We summarize the aforementioned state-of-the-art DL-based HAR methods in Table 1. In Table 1, we can easily verify that for smartphone-based HAR systems, UCI and WISDM are the most popular public standard smartphone

sensor data used in previous works. Moreover, we can also determine a research gap: convolutional AR with LSTM is a novel architecture by which we can obtain higher accuracy with permissible computational times in HAR domain applications. In [81], the authors used a combination of an AE, a CNN, and LSTM, although this approach exhibited several clear differences from our method. First, we introduce a combination of convolutional layers with an AE, and then the output of the convolutional AE passes through LSTM. Conversely, in [81], the authors used three different modules, where the input first passed through the AE, then through the CNN, and finally through LSTM. In our work, we take advantage of the convolutional layer of a CNN in combination with an AE and LSTM. Second, in [81], the authors used channel state information (CSI) frames as inputs. In contrast, in this study, we take time window segments of raw signals as inputs. Third, in [81], to prove the efficiency of the proposed AE-LRCN architecture, the authors used a self-collected dataset for activities such as emptying, sitting, walking, running and standing. In this paper, to exhibit the efficiency of our proposed architecture, we use four popular public datasets (UCI, WISDM, OPPORTUNITY, and PAMAP2).

III. PRELIMINARIES

The proposed DL architecture is a combination of a convolutional AE and LSTM. The Convolutional AE leverages the convolutional filtering performance of CNNs with unsupervised AE pretraining. Therefore, to understand the concept of a convolutional AE, it is necessary to separately understand the concepts of both the CNN and AE.

A. CNNs

Recently, CNNs have achieved great successes in various domains, such as image classification and speech recognition, due to their ability to learn locally connected features. Generally, CNNs consist of three different layers: convolution layers, pooling layers and fully connected layers [91]. The convolution layers are the fundamental concepts of a CNN architecture that perform feature extraction. Input feature map downsampling is performed by the pooling layers, and the fully connected layers are used for classification. Both max-pooling and average-pooling layers are commonly employed to perform local maximization and averaging operations on the input features, respectively. Motivated by [92], we present the max-pooling layer concept by using the following equation :

$$mp_i^{l+1} = \max_{(t-1)p+1 < k < tp} q_i^l(k), t = 1, 2, \dots, Q \quad (1)$$

An average-pooling layer is represented as follows :

$$ap_i^{l+1} = \text{avg}_{(t-1)p+1 < k < tp} q_i^l(k), t = 1, 2, \dots, Q \quad (2)$$

where $q_i^l(k)$ is the value of the k^{th} neuron in the i^{th} feature map of the l^{th} layer, t denotes the t^{th} moving step of the filter, p is the width of the pooling filter, and mp_i^{l+1} or ap_i^{l+1} represents

the corresponding output in the $(l+1)^{\text{th}}$ layer provided by the pooling operation.

In comparison with a fully connected layer, a convolution layer has much fewer parameters due to sparse connectivity and weight sharing, thereby minimizing the possibility of overfitting. However, due to the tremendous popularity of traditional CNNs in HAR, the recently proposed CNN-based HAR models adopt 1-2 fully connected layers as classifiers [57], [87]. Although fully connected layers can adequately perform classification, various parameters lead to the risk of overfitting.

B. CONVOLUTIONAL OPERATION

The given input data are processed by the convolution kernel, which produces processed features as outputs. These processed features are known as feature maps. The multiple kernels that reside in convolution layers are used to extract the relevant features. Motivated by [92], the ubiquitous convolutional operation is denoted by

$$y_i^{l+1}(j) = w_i^l * x^l(j) + b_i^l \quad (3)$$

where w_i^l and b_i^l represent the weight and bias of the i^{th} kernel in the l^{th} layer and $x^l(j)$ denotes the j^{th} local region of layer l .

Generally, after the convolution operation, a nonlinear transformation is subsequently employed by using an activation function. A commonly used activation function is the rectified linear unit (ReLU), which is represented by

$$a_i^{l+1}(j) = f(y_i^{l+1}(j)) = \max\{0, y_i^{l+1}(j)\} \quad (4)$$

C. AE

The basic principles of AEs were proposed in [93]. An AE is a feedforward neural network that accepts $x \in R^d$ an input and first maps it to a latent representation $h \in R^{d'}$ to produce an output under certain constraints.

An AE encodes and decodes the given inputs to produce unsupervised pretraining data. The deterministic encoding function used to construct a nonlinear mapping for the given input x is as follows :

$$d_i = \sigma(wx_i + b) \quad (5)$$

where σ is the nonlinear activation function and w and b are the weights and biases, respectively.

The decoding function used to reconstruct the input vector x with encoded features is as follows :

$$\hat{d}_i = \sigma(\hat{w}d_i + \hat{b}) \quad (6)$$

where \hat{w} and \hat{b} are the weights and biases of the decoder, respectively.

IV. THE PROPOSED MODEL

Multivariate time-series data are collected from built-in smartphone-based sensors to identify various human activities. Fine-grained information can be obtained by using sensory data from sensors such as triaxial accelerometers and gyroscopes. However, the data collected using smartphone

sensors are noisy and not in an appropriate form. It is not possible to recognize fundamental patterns with such noisy raw sensory data. To remove this noise, conventional filtering techniques such as low-pass, high-pass, and median filtering are used. After removing the noise, feature engineering is applied to extract relevant features, and ultimately, the extracted feature vector is fed to the classifier as its input. As mentioned earlier, conventional noise removal, feature engineering, and classification methods require substantial human expertise and intervention, and they fail to reveal the temporal interdependence of data [81]. CNNs are popularly used for automatic feature extraction in several domains, including HAR. CNNs, however, use backpropagation neural networks to train the kernels/weights used for convolution, which takes a long time.

In both ML and DL architectures, time window-based sensor data segmentation is required to assign a single activity class. Researchers approximate the size of the sliding window over the sensor data streams to extract features in cases involving labeled data [22]. Sometimes, this strategy leads to the loss of important activity information. Activity recognition accuracy may increase with an increase in the length of the segments, but a long window size causes response time delays in real-time HAR. Hence, an unsupervised feature learning approach such as an AE is beneficial in scenarios in which we do not have labeled data.

The proposed model consists of three modules, as represented in Figure 4. The first module is a convolutional AE, which consists of a convolution layer, a pooling layer and a deconvolution layer. The output of the convolutional AE is passed through a flattened layer to change it to the LSTM input format. The LSTM output is passed through a fully connected layer to obtain a high-level representation. Finally, a softmax layer is used for the final human physical activity recognition step. Thus, this model is capable of identifying the temporal dependencies among the time-series signals acquired through smartphone sensors.

As mentioned, a flattened layer is added after the deconvolution layer in the convolutional AE to format the feature data for the LSTM layer. This is because the data format of the convolution layer is different from the input data format of the LSTM layer. A time-distributed wrapper, which is provided by the Keras library in Python, takes a layer as an argument and applies convolutions to the signal while maintaining its temporal integrity for the LSTM layers [94]. As the time-distributed layer works with a 3D data format, we need to reshape the input signal from 128 time frames with an accurate number of signals. A total of 128 time frames are divided into four slices with 32 time frames each.

The three different aforementioned modules used in this proposed deep learning architecture are further explained as follows:

A. CONVOLUTIONAL AE

One of the AE variations is the convolutional AE [69], in which a fully connected layer is replaced by a

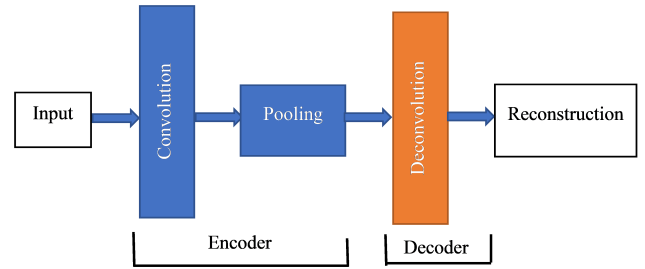


FIGURE 2. Convolutional AE architecture.

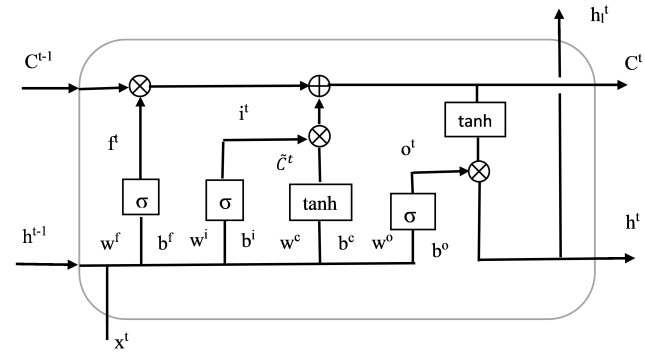


FIGURE 3. LSTM architecture [81].

convolutional layer. Convolutional AEs have the advantages of both convolutional layers and the unsupervised pretraining capability of an AE. In contrast to the conventional AE network, the convolutional AE contains convolutional layers in the encoder and deconvolution layers instead of a fully connected layer in the decoder. Our proposed convolutional AE includes convolution, pooling, and deconvolution layers, as presented in Figure 2. The encoder consists of one convolution layer and a pooling layer. The decoder consists of a deconvolution layer. Encoding the result of the convolution operation with a max-pooling layer permits higher-layer representations that are invariant to small input translations and reduce the computational cost of the proposed approach [26]. The convolution-deconvolution layer is followed by an activation function, which is represented as follows [95]:

$$h^k = \sigma \left(\sum_{l \in L} x^l \otimes w^k + b^k \right) \quad (7)$$

where

- h^k = the latent representation of the k^{th} feature map of the current layer
- σ = the activation function
- x^l = the l^{th} feature map of the group of feature maps L obtained from the previous layer
- \otimes = a 2D convolution operation
- w^k = the weights of the k^{th} feature map of the current layer
- b^k = the bias of the k^{th} feature map of the current layer

“Valid convolution” is performed by the convolution layer, and “full convolution” is performed by the deconvolution

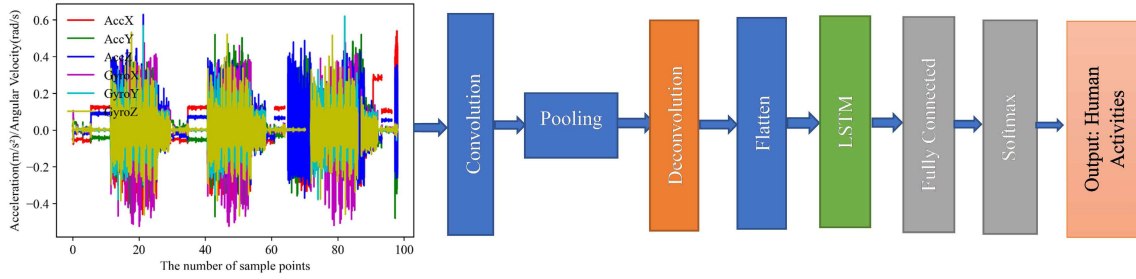


FIGURE 4. Convolutional AE-LSTM architecture.

layer. For instance, if the size of a feature map x^l is $p \times p$ and the size of the filter is $q \times q$, then after performing the valid convolution, the size becomes $(p-q+1) \times (p-q+1)$, and after performing the full convolution, the size becomes $(p+q-1) \times (p+q-1)$ [95].

By utilizing the maximum activity within the input feature maps, a max-pooling layer pools features, and according to the size of the pooling kernel, it constructs reduced-size output feature maps.

B. LSTM

Temporal features in time-series sensor data have great importance when modeling human movement [23]. Recently, recurrent neural networks (RNNs), most remarkably those that depend on LSTM [96], have achieved impressive performance in different domains, including HAR. The LSTM architecture is responsible for extracting the temporal features from sensory signals due to its temporal characteristics and long-term dependencies. The conventional architecture of LSTM [81] is represented in Figure 3.

In our proposed model, the convolutional AE, as explained in section IV-A, is followed by an LSTM model. The output of the convolutional AE and the compressed features are the inputs of the LSTM for deducing the latent temporal interactions throughout the timeframes. According to Figure 3, at time frame t , x^t is the input signal and h^t is the hidden state. At time frame $t-1$, C^{t-1} is the memory cell state. w^f , w_i , w^c , and w^o and b^f , b^i , b^c , b^o are the weights and biases, respectively. σ and \tanh are the activation functions. In the first step, the LSTM calculates the previous information from the cell state C^{t-1} by using a forget gate as follows :

$$f^t = \sigma(w^f[h^{t-1}, x^t] + b^f) \quad (8)$$

where f^t is either 0 or 1 to denote the total block and total transit of the information, respectively. In the next step, the LSTM calculates the upcoming information to be stored by using a two-step process. The first part regulates the parameters to be used via the following equation:

$$i^t = \sigma(w^i[h^{t-1}, x^t] + b^i) \quad (9)$$

The second part determines an optimal state value \tilde{C}^t by using the following equation:

$$\tilde{C}^t = \tanh(w^c[h^{t-1}, x^t] + b^c) \quad (10)$$

In the third step, the LSTM determines the current state C^t by using the following equation:

$$C^t = f^t * C^{t-1} + i^t * \tilde{C}^t \quad (11)$$

As exhibited in Figure 3, the filtered version of the compressed cell state $\tanh(C^t)$ is the hidden network output h^t . The part of the information that should be preserved is calculated by using the sigmoid layer o^t , which is determined according to the following equation:

$$o^t = \sigma(w^o[h^{t-1}, x^t] + b^o) \quad (12)$$

Ultimately, the final hidden output h^t is articulated as

$$h^t = o^t * \tanh(C^t) \quad (13)$$

C. FULLY CONNECTED AND SOFTMAX CLASSIFICATION LAYERS

Fully connected layers are used to follow high-level representations. In this work, the LSTM outputs are fed into two hidden layers, and ultimately, a softmax layer is used for the final activity identification step.

V. PERFORMANCE EVALUATION

We present the experimental results of our proposed method (ConvAE-LSTM) on two smartphone sensor-based public standard datasets (UCI [11] and WISDM [10]) and two body-worn, sensor-based public standard datasets (OPPORTUNITY [82] and PAMAP2 [88]) in this section. In this article, we mainly focus on smartphone sensor-based HAR. Hence, we present detailed experimental results obtained using the UCI and WISDM datasets. To demonstrate the efficiency of our proposed model, we also present the experimental results obtained by using OPPORTUNITY and PAMAP2 in terms of accuracy and F1-scores.

A. DATASET

To exhibit the efficiency of the proposed method, we use two popular public standard smartphone sensor-based HAR datasets and two body-worn, sensor-based public standard datasets that represent both static and dynamic activities. The standard datasets used are explained as follows:

- **UCI dataset** [11]: This standard dataset is taken from the publicly available ‘University of California Irvine

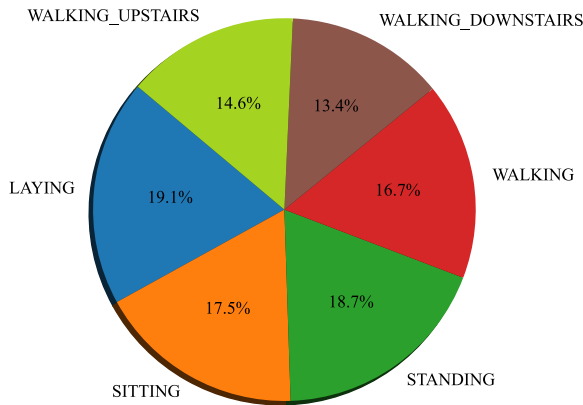


FIGURE 5. % of different activities in the UCI dataset.

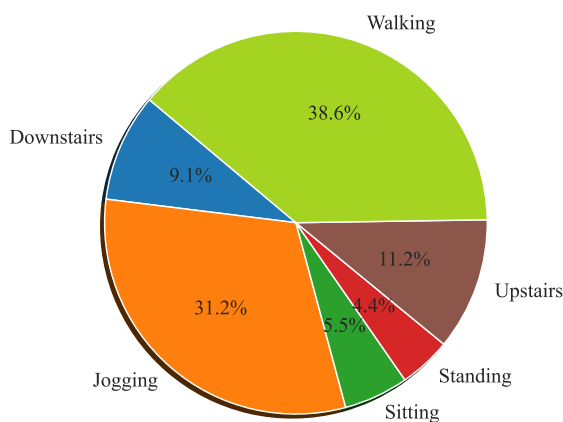


FIGURE 6. % of different activities in the WISDM dataset.

(UCI) Machine Learning” repository. This is a balanced dataset, as shown in Figure 5. This dataset was collected from 30 subjects aged between 19 and 48 years who performed 6 different daily life activities such as “sitting”, “standing”, “walking”, “lying”, “walking upstairs” and “walking downstairs”. To collect the data, a waist-mounted smartphone (*Samsung Galaxy SII*) with a built-in accelerometer and gyroscope was used. This dataset was also collected in a laboratory environment with proper surveillance. This collected dataset consists of 10,299 instances in total. Triaxial linear acceleration and angular velocity measurements were collected at a constant sampling rate of 50 Hz.

- **WISDM Actitracker dataset** [10]: This standard public dataset is provided by the ‘Wireless Sensor and Data Mining (WISDM) lab. The dataset was collected from 36 subjects using smartphone accelerometer sensors. Each subject was asked to perform 6 different human physical activities, such as “sitting”, “standing”, “walking”, “jogging”, “walking upstairs” and “walking downstairs”. This dataset was also collected in a laboratory environment with proper surveillance. In this dataset, the total number of instances is 1,098,207. The

3-axial linear acceleration measurements were collected at a constant sampling rate of 20 Hz.

- **OPPORTUNITY dataset** [82]: This dataset consists of complicated naturalistic activities including a large number of atomic activities (over 27,000) recorded in a sensory-rich environment at a constant sampling rate of 30 Hz. It includes recordings of 12 participants obtained using 15 networked sensor systems with 72 sensors from 10 different modalities that are embedded in the environment, in objects, and on the body. We only consider the on-body sensors, including the 3-axial accelerometer and inertial measurement units. This is an 18-class classification problem (including the null class).
- **PAMAP2 dataset** [88]: It contains recordings from 9 subjects (8 men and 1 woman) who were asked to perform 18 lifestyle tasks, including household chores, at a constant sampling rate of 100 Hz. Over the course of 10 hours, data from inertial measurement equipment on the hand, chest, and ankle were collected, including accelerometer, gyroscope, magnetometer, temperature, and heart rate data. The resulting dataset has 52 dimensions.

Using a continuous sequence of sensory data, an end-to-end HAR model is implemented in this work. During this process, from the raw sensory data, a sequence of short time-series data is extracted. To save the transient connections between the information focused on a given activity, a sliding window with a 50% overlapping rate is used to segment the collected raw sensory data. For the above datasets, the length of the sliding window is 128 with a step size of 64.

B. EXPERIMENTAL SETUP

In the training phase, forward calculation is used with the training set to obtain the network output. Afterward, in between the predicted outputs and actual outputs, the cross-entropy errors are calculated. Then, the Adam optimizer is used to backpropagate the errors in the sequence of layers to update the hyperparameters of our proposed network. After calculating the adaptive learning rate of each parameter, the hyperparameters of the objective function are optimized by Adam [97]. The Keras API permits one to move from the beginning to the end result with the least viable delay [94]. During this experiment, we build a sequential Keras model (version 2.4.3) with TensorFlow in the backend (version 2.3.1). For our experiments, we use a single GPU (NVIDIA GTX 1060 GPU with 6 GB of memory).

To perform the experiment, the first two datasets are divided into two different groups: 70% of the volunteers are selected for training, and 30% are used for testing the proposed HAR solution. Hence, the same subjects’ data are not included in both the training and testing sets. In our experiment, simple 5-fold cross validation is used to generate multiple training and validation splits from the training set, as cross validation is less computationally complex than other methods such as leave-one-out cross validation [98].

TABLE 2. Average training and testing times yielded for the UCI-HAR dataset.

Performance	Methods					
	CNN	LSTM	AE	CNN-LSTM	Convolutional AE	ConvAE-LSTM
Training Time	3.42 s	3.41 s	3.12 s	3.78 s	3.28 s	3.32 ms
Testing Time	372.5 ms	374.1 ms	245.3 ms	394.3 ms	256.4 ms	261.4 ms
Testing Accuracy	93.96%	89.45%	91.24%	94.52%	92.67%	98.14%

Leave-one-subject-out (LOSO) cross validation is also performed to provide a more comprehensive evaluation. We use data from one subject for testing and those from the remaining subjects for training. This cross-subject test is more rigorous because the test data are hidden from the models, making it a more realistic setting for validating the models' generalization abilities. By using all the datasets, in the input layer of the CNN, 1D convolution is performed. In our experiment, a ReLU is used as an activation function for the convolution layers with a kernel size of 3, a stride of 2, and a filter size of 64. Similarly, in the max-pooling layer, the pooling size and stride are both of size 2. The learning rate is set to 0.001. The optimizer updates and calculates the network parameters that affect the model training and output processes to approximate or reach the optimal value, thereby reducing the loss function.

C. EXPERIMENTAL RESULTS

In this section, we discuss the experimental results of the proposed method in terms of accuracy, precision, recall, computational complexity, and testing time by using the first two datasets. The experimental results obtained with the other two datasets are provided later. To prove the capability of our proposed model, we also compare the results of our proposed model with those of other commonly used DL approaches, such as a CNN, LSTM, an AE, CNN-LSTM, and a convolutional AE. In this experiment, we take the simple CNN and LSTM architectures as proposed in [58]. In the cases of the AE and convolutional AE, we utilize a max-pooling layer for encoding, which is similar to our proposed method (ConvAE-LSTM).

1) UCI DATASET

Utilizing the UCI-HAR dataset, we perform exhaustive experiments on various DL approaches, such as a CNN, LSTM, an AE, CNN-LSTM, a convolutional AE, and the proposed method (ConvAE-LSTM). Table 2 demonstrates the training time, testing time, and testing accuracy of the aforementioned DL approaches, including our proposed model. All the DL approaches are used in our experiment to demonstrate the effectiveness of our proposed method in the same experimental environment. The computational time of our proposed model is very competitive with those of the aforementioned DL approaches in the same computational environment. Moreover, the computational time of ConvAE-LSTM is very competitive with that of the state-of-the-art approach proposed in [51], where the computational times for training and testing are 3.4274 s and 372.6 ms,

TABLE 3. Classification report for ConvAE-LSTM with UCI.

Class	Precision	Recall	F1-score	Support
STANDING	1.00	0.96	0.98	516
SITTING	0.96	0.98	0.97	463
WALKING	0.98	1.00	0.99	413
LYING	0.90	0.98	0.98	433
UPSTAIRS	0.98	0.89	0.94	583
DOWNSTAIRS	1.00	1.00	1.00	539
accuracy			0.98	2947
macro avg	0.97	0.98	0.98	2947
weighted avg	0.98	0.98	0.98	2947

respectively, when using the CNN with the UCI dataset. The testing accuracy of ConvAE-LSTM is 98.14%, which is much higher than that of other popular DL approaches. However, the computational times of the AE and convolutional AE are the lowest among all the mentioned approaches in Table 2, whereas their accuracies are very poor in comparison with that of our proposed approach, as these two methods do not consider the temporal dependencies among the raw sensory time-series data.

Table 3 demonstrates the detailed classification results of our proposed model. In this proposed model, as we take both a convolutional AE and LSTM in combination, the F1-scores of activities such as "walking", "walking down-stairs" and "walking upstairs" are 99%, 100%, and 94%, respectively. Therefore, we can conclude that our proposed method can distinguish similar activity patterns very efficiently, which is not achieved by using only the CNN method, as mentioned in the HAR literature. Similarly, in cases with static features such as "sitting", "lying" and "standing", we achieve F1-scores of 97%, 98%, and 98%, respectively, which are much better than those of any CNN-based HAR solution. From the experimental results, we can easily conclude that our proposed method not only efficiently differentiates among the static and dynamic activities but can also efficiently identify similar activity patterns. Figure 7 depicts the confusion matrix of the different activities in the testing set. We also analyze the activity recognition accuracy of the proposed ConvAE-LSTM method and compare its performance with that of UCI data-based state-of-the-art HAR solutions, the CNN, a CNN +handcrafted features [8], LSTM with bidirectional LSTM [58], bidirectional LSTM alone [71], CNN-LSTM [74] and LSTM-CNN [80], as well as two popularly used shallow ML approaches (a random forest (RF) and a support vector machine (SVM)). Table 4 compares the average accuracy of ConvAE-LSTM with that of the aforementioned approaches. ConvAE-LSTM provides

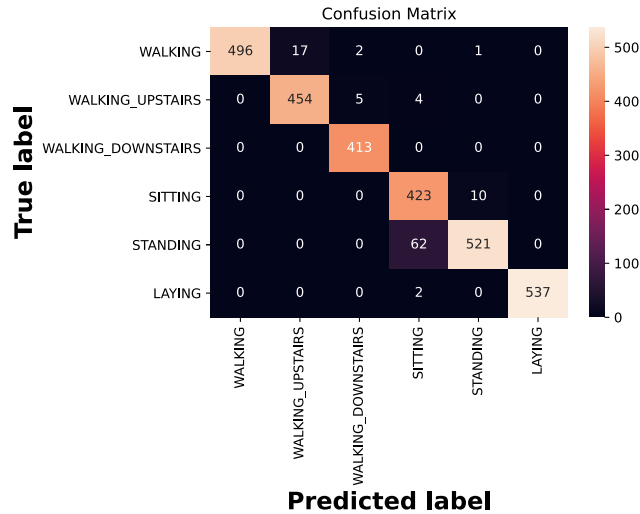


FIGURE 7. Confusion matrix for CAE-LSTM with UCI.

TABLE 4. Average accuracy comparison on the UCI-HAR dataset.

Methods	Testing accuracy
CNN [8]	94.79%
CNN + Handcrafted features [8]	95.75%
LSTM [58]	89.14%
Stacked-LSTM [99]	93.13%
MLSTM-FCN [90]	96.71%
InnoHAR [79]	94.6%
Bidirectional LSTM [58]	89.94%
Bidirectional LSTM [71]	93.79%
CNN-LSTM [74]	93.40%
LSTM-CNN [80]	95.78%(F1-Score)
RF	95.67%
SVM	96.89%
ConvAE-LSTM	98.14% & 97.67%(F1-Score)

the best activity recognition accuracy (98.14%) among all tested approaches.

2) WISDM DATASET

Utilizing the WISDM dataset, we perform exhaustive experiments on various DL approaches, such as the CNN, LSTM, the AE, CNN-LSTM, the convolutional AE, and the proposed method (ConvAE-LSTM). Table 5 demonstrates the training times, testing times, and testing accuracies of the aforementioned DL approaches, including our proposed model. The computational time of our proposed model is very competitive with those of the other DL approaches in the same computational environment. The testing accuracy of the proposed model is 97.76%, which is much higher than that of other popular DL approaches. However, the training and testing times of the AE and convolutional AE are the lowest among all the mentioned approaches in Table 5, whereas their accuracies are very poor in comparison with that of our proposed approach, as these two methods do not consider the temporal dependencies among the raw sensory time-series data. It is pertinent to mention that the WISDM dataset is imbalanced, as depicted in Figure 6. In the case of an imbalanced dataset, several techniques are required to balance the dataset according to the HAR literature, and

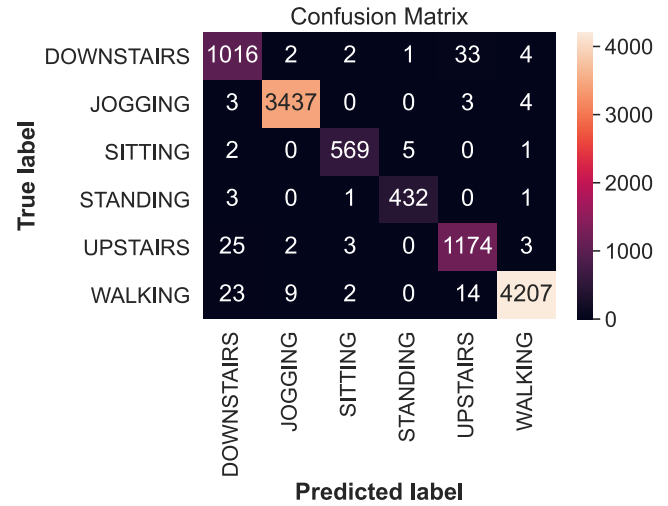


FIGURE 8. Confusion matrix for ConvAE-LSTM with WISDM.

most conventional ML algorithms fail to classify imbalanced datasets properly. However, while performing our experiment, none of these techniques are applied to convert the imbalanced dataset into a balanced dataset. The imbalanced WISDM dataset is employed in our proposed model directly, and we obtain very high identification performance for all the activities, as the F1-score of each activity is greater than 95% and the overall accuracy is 99%. Hence, we can conclude that our proposed method provides an added advantage to overcome the imbalanced dataset issue.

Table 6 shows the detailed classification results of our proposed method with WISDM. In this proposed method, as we utilize both a convolutional AE and LSTM in combination, the F1-scores for activities such as “jogging”, “walking”, “walking downstairs” and “walking upstairs” are 100%, 99%, 95%, and 97%, respectively. Therefore, we can conclude that our proposed method can distinguish similar activity patterns very efficiently, which is not achieved when using only the CNN method, as mentioned in the HAR literature. Moreover, for both static activities (“sitting” and “standing”), the F1-score is 99%, which is very remarkable according to the HAR literature. Figure 8 depicts the confusion matrix for the different activities in the testing set.

We also analyze the activity recognition accuracy of our proposed ConvAE-LSTM method, and compare its performance with that of WISDM dataset-based standard HAR models, the CNN [18], a CNN + handcrafted features [48], an AE ensemble [66], the convolutional AE [69] and LSTM-CNN [80], as well as two shallow ML approaches (an RF and an SVM). Table 7 compares the average accuracy of ConvAE-LSTM with that of the aforementioned approaches. ConvAE-LSTM provides the best activity recognition accuracy (98.67%) among all tested approaches.

D. PERFORMANCE EVALUATION OF THE PROPOSED MODEL USING LOSO CROSS VALIDATION

We also perform LOSO cross validation to provide a more comprehensive evaluation. We use the data from one subject

TABLE 5. Average training and testing time comparisons on the WISDM dataset.

Performance	Methods					
	CNN	LSTM	AE	CNN-LSTM	Convolutional AE	ConvAE-LSTM
Training Time	3.22 s	3.34 s	3.09 s	3.37 s	3.19 s	3.09 ms
Testing Time	342.3 ms	336.2 ms	233.5 ms	378.4 ms	247.2 ms	253.2 ms
Testing Accuracy	93.27%	88.34%	90.08%	92.78%	91.97%	98.67%

TABLE 6. Classification report for ConvAE-LSTM with WISDM.

Class	Precision	Recall	F1-score	Support
DOWNSTAIRS	0.95	0.96	0.95	1058
JOGGING	1.00	1.00	1.00	3447
SITTING	0.99	0.99	0.99	577
STANDING	0.99	0.99	0.99	437
UPSTAIRS	0.96	0.97	0.97	1207
WALKING	1.00	0.99	0.99	4255
accuracy			0.99	10981
macro avg	0.98	0.98	0.98	10981
weighted avg	0.99	0.99	0.99	10981

TABLE 7. Average accuracy comparison on the WISDM dataset.

Methods	Testing Accuracy
CNN [18]	96.88%
CNN + Handcrafted features [48]	90.42%
Ensemble of AE [66]	80.8%
Convolutional AE [69]	94.9%
Bidirectional LSTM [71]	93.79%
CNN-LSTM [74]	93.40%
LSTM-CNN [80]	95.85%(F1-Score)
RF	86.78%
SVM	91.12%
ConvAE-LSTM	98.67% & 98.17%(F1-Score)

for testing and the data from the remaining subjects for training. This cross-subject test is more difficult because the test data are hidden from the models, making it a more realistic setting for validating the models' generalization abilities. After testing the models with a unique subject for each fold, we obtain different evaluation metric values, one from each fold. To assess the accuracy of the models, we take the $mean \pm SD$ of all the accuracy metrics. We perform the LOSO cross-validation evaluation technique with a 95% confidence level as follows:

- In LOSO cross-validation, for each fold, we obtain different accuracy metrics. We calculate the mean and SD for each accuracy metric.
- After calculating the average accuracy, we calculate the error: $error = 1 - accuracy$.
- Next, we calculate the confidence interval for the classification error using

$$error \pm constant * \sqrt{\frac{error * (1 - error)}{n}},$$

where n is the number of samples used to evaluate the model and the value of the constant is 1.96, which is provided by the statistics for the 95% confidence level.

TABLE 8. Average accuracy comparison on the UCI-HAR dataset with LOSO cross validation.

Methods	95% Confidence Interval	True Accuracy
CNN [8]	(0.9089, 0.9123)	91.18%
CNN + Handcrafted features [8]	(0.9187, 0.9267)	92.45%
LSTM [58]	(0.8184, 0.8272)	82.31%
Stacked-LSTM [99]	(0.8690, 0.8771)	87.56%
MLSTM-FCN [90]	(0.9635, 0.9691)	96.71%
InnoHAR [79]	(0.8995, 0.9067)	90.48%
Bidirectional LSTM [58]	(0.8286, 0.8354)	83.27%
Bidirectional LSTM [71]	(0.8912, 0.8968)	89.56%
CNN-LSTM [74]	(0.9012, 0.9197)	90.78%
LSTM-CNN [80]	(0.9234, 0.9273)	92.45%
RF	(0.9088, 0.9147)	91.33%
SVM	(0.9232, 0.9279)	92.63%
ConvAE-LSTM	(0.9698, 0.9767)	97.13% & 97.56%(F1-Score)

TABLE 9. Average accuracy comparison on the WISDM dataset with LOSO cross validation.

Methods	95% Confidence Interval	True Accuracy
CNN [18]	(0.9223, 0.9291)	92.67%
CNN + Handcrafted features [48]	(0.8592, 0.8636)	86.12%
Ensemble of AE [66]	(0.7394, 0.7464)	74.37%
Convolutional AE [69]	(0.8997, 0.9058)	90.28%
Bidirectional LSTM [71]	(0.8912, 0.8978)	89.41%
CNN-LSTM [74]	(0.8688, 0.8742)	87.23%
LSTM-CNN [80]	(0.9095, 0.9154)	91.34%
RF	(0.8191, 0.8228)	82.19%
SVM	(0.8592, 0.8647)	86.23%
ConvAE-LSTM	(0.9686, 0.9717)	97.08% & 97.38%(F1-Score)

By using the aforementioned steps, we calculate the true accuracy and confidence interval with a 95% confidence level (the significance level is 0.05) for each of the models, as presented in Tables 8 and 9.

Even when using LOSO cross validation, which is more realistic and difficult, the proposed technique outperforms the aforementioned state-of-the-art methods on both the UCI and WISDM datasets. By utilizing LOSO cross validation, the accuracies of the proposed model are 97.13% and 97.56% on the UCI and WISDM datasets, respectively, with a 0.05 level of significance. Similarly, the F1-scores of the proposed model are 97.08% and 97.38% on the UCI and WISDM datasets, respectively, with a 0.05 level of significance. The performance of the different models according to LOSO cross validation with a 95% confidence level is presented in Tables 8 and 9 for the UCI and WISDM datasets, respectively.

TABLE 10. Average accuracy comparison on the OPPORTUNITY and PAMAP2 datasets.

Methods	OPPORTUNITY	PAMAP2
CNN [18]	88.19%	-
CNN [49]	92.24%	92.55%
LSTM [58]	-	85.86%
Regularize AE [65]	69.70%	-
Ensemble AE [66]	56%	-
AE [67]	43%(F1-Score)	94%(F1-Score)
Convolutional AE [69]	84.9%	-
Bidirectional LSTM [72]	90.5%	-
Deep Convolutional LSTM [73]	93%(F1-Score)	-
InnoHAR [79]	94.6%(F1-Score)	93.5%(F1-Score)
LSTM-CNN [80]	-	92.63%
RF	87.29%	-
SVM	85.11%	-
ConvAE-LSTM	95.69%	94.33%
	95.54%(F1-Score)	94.46%(F1-Score)

E. EXPERIMENTAL RESULTS OBTAINED ON THE OPPORTUNITY AND PAMAP2 DATASETS

We use LOSO cross validation to perform an experiment using these two body-worn sensor datasets. The performance of the different models according to LOSO cross validation with a 95% confidence level is presented in Table 10 for the OPPORTUNITY and PAMAP2 datasets. We achieve an accuracy of 95.69% and an F1-score of 95.54% on the OPPORTUNITY dataset and an accuracy of 94.33% and an F1-score of 94.46% on the PAMAP2 dataset. Our proposed method outperforms the existing methods on both datasets.

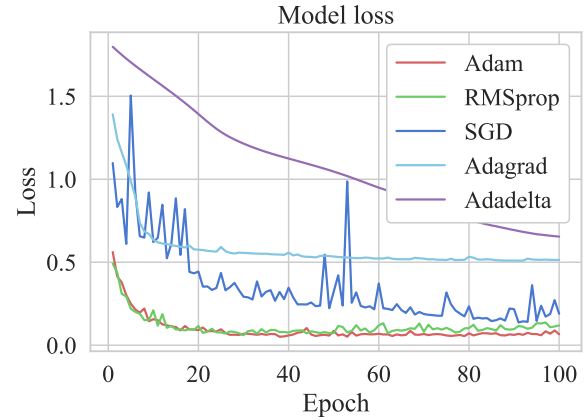
F. STATISTICAL ANALYSIS

To prove the generalization and robustness of our proposed technique, it is also necessary to perform a statistical test. In this study, we consider four different datasets: UCI [11], WISDM [10], PAMAP2 [88] and OPPORTUNITY [82]. We perform the Friedman test [100], [101], which is a nonparametric equivalent of the repeated-measures ANOVA technique. In our statistical test, we assume that the null hypothesis is as follows: “There are no significant differences among the model performances”. The alternate hypothesis is as follows: “There is a significant difference among the model performances”.

The following steps are executed to perform the Friedman test.

- First, represent the observed accuracy in a matrix x_{ij} with n rows and k columns, where the accuracies of 16 different models are presented corresponding to the 4 different datasets. In our experiment, $n=16$ and $k=4$.
- Then, for each dataset, separately calculate the ranks of the models.
- Replace the data with a new matrix $\{r_{ij}\}_{n \times k}$, where entry r_{ij} is the rank of x_{ij} within block i .
- Calculate the values $\bar{r}_{.j} = \frac{1}{n} \sum_{i=1}^n r_{ij}$.
- Then, calculate the test statistic (Friedman test statistic) using the following formula:

$$Q = \frac{12n}{k(k+1)} \sum_{j=1}^k \left(\bar{r}_{.j} - \frac{k+1}{2} \right)^2.$$

**FIGURE 9.** Effects of the optimizers on model performance with the UCI dataset.

- Finally, using a chi-squared distribution, approximate the probability distribution of Q . In this case, the p-value is given by $P(\chi_{k-1}^2 \geq Q)$.

After performing the abovementioned test with $\alpha = 0.05$, the observed level of significance $p \leq \alpha$. Hence, the result is statistically significant. However, this p-value is based on a single accuracy and thus may give an inappropriate result. Therefore, to adjust our statistical confidence measures based on the number of completed tests, we require multiple testing correction processes. The Bonferroni correction [102] is the simplest and most extensively used multiple testing correction method. If we use a significance threshold of α but run n independent tests, the Bonferroni adjustment only considers a score significant if the matching p-value $\leq \alpha/n$. A Bonferroni correction [102] is used to control the familywise Type-I error rate, resulting in an adjusted significance of 0.0031. In Table 11, the p-values are less than 0.31%, and we can statistically conclude that the proposed model outperforms the state-of-the-art models as the assumed null hypothesis is rejected. Hence, there are significant differences between the model performances.

G. EFFECTS OF THE HYPERPARAMETERS ON THE PERFORMANCE OF THE PROPOSED MODEL

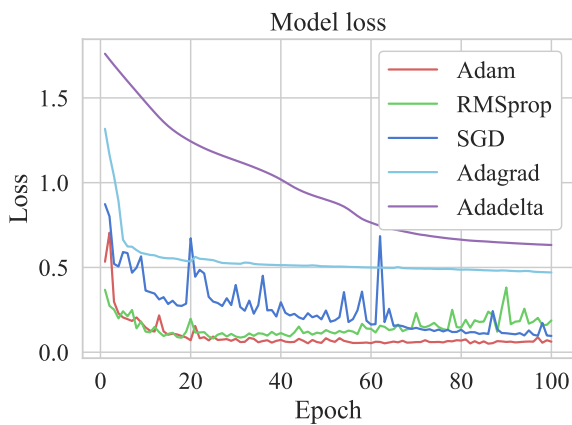
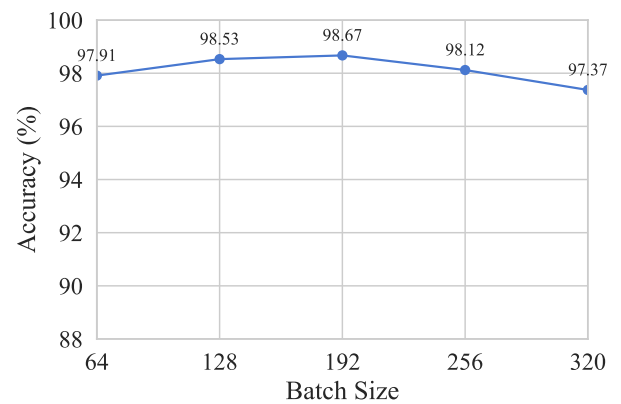
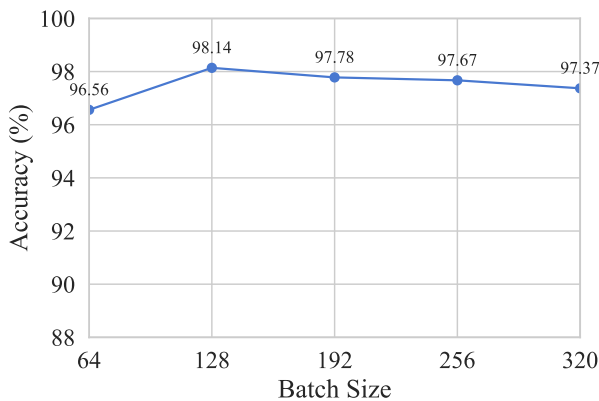
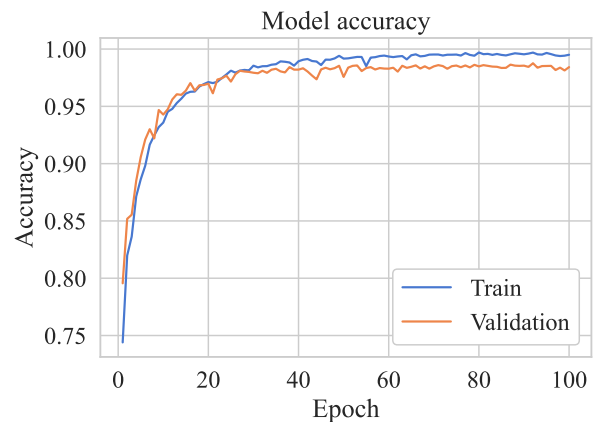
The performance of a classification model is heavily influenced by its hyperparameters. The impacts of two major hyperparameters, that is, the number of epochs and the batch size, on model performance are presented in this section. Tests are run on the first two datasets, and the performance of the model is assessed by tweaking a few model parameters. The accuracy is utilized as the criterion for evaluation.

1) IMPACT OF THE OPTIMIZER

An optimizer updates and estimates the network parameters that affect the model training and output processes to approximate or reach the optimal value, thereby reducing the loss function. This is an essential part of any DL approach. As a result, selecting an appropriate optimizer for DL training is

TABLE 11. Comparison of the models using the Friedman test.

	[8]	[18]	[49]	[48]	[66]	[69]	[58]	[99]	[90]	[79]	[58]	[71]	[74]	[80]	RF	SVM
[18]	5.42E-09															
[49]	4.56E-09	3.14E-08														
[48]	2.56E-08	1.56E-08	3.87E-09													
[66]	2.13E-09	3.23E-09	2.18E-09	3.16E-08												
[69]	1.17E-10	2.14E-9	1.67E-10	2.49E-9	1.67E-10											
[58]	1.15E-10	2.45E-09	1.77E-10	2.26E-09	2.34E-10	2.31E-09										
[99]	1.14E-10	2.13E-09	2.17E-09	2.22E-09	1.66E-10	1.43E-10	1.17E-10									
[90]	4.17E-09	4.21E-09	3.17E-09	3.38E-09	3.11E-09	4.11E-09	3.89E-09	3.67E-09								
[79]	1.15E-10	1.18E-10	1.19E-10	1.22E-10	1.21E-10	1.19E-10	1.14E-10	1.19E-10	1.23E-10							
[58]	2.34E-09	2.42E-09	2.27E-10	3.65E-09	2.78E-09	4.89E-10	3.23E-09	2.65E-09	2.37E-09	3.13E-09						
[71]	1.53E-10	2.40E-09	5.47E-09	2.29E-08	1.66E-10	3.11E-09	4.27E-09	3.27E-09	1.43E-10	4.21E-09	4.16E-09					
[74]	3.39E-09	2.29E-08	2.82E-08	2.61E-08	2.57E-09	1.47E-10	2.11E-0	1.22E-10	1.10E-10	2.23E-08	1.18E-10	2.28E-09				
[80]	3.59E-09	2.34E-09	2.54E-09	2.82E-09	3.11E-09	2.13E-09	2.56E-09	2.19E-09	2.12E-09	2.31E-09	3.14E-09	3.28E-09	2.37E-09			
RF	5.46E-09	4.78E-09	2.14E-08	2.17E-09	4.18E-09	3.22E-09	2.12E-08	3.15E-09	2.77E-09	2.41E-09	1.57E-10	3.24E-09	2.58E-08	4.17E-09		
SVM	4.12E-09	3.17E-09	1.12E-10	2.15E-09	2.27E-08	2.66E-08	2.59E-09	1.56E-10	4.32E-10	2.97E-09	3.14E-09	4.36E-10	3.38E-09	2.28E-08	3.15E-10	
ConvAE-LSTM	5.02E-10	4.76E-10	1.10E-10	2.78E-10	1.11E-10	2.17E-09	2.29E-09	1.18E-10	3.24E-10	2.55E-09	2.16E-09	2.14E-09	1.26E-10	4.84E-10	3.28E-10	2.33E-09

**FIGURE 10.** Effects of the optimizers on model performance with the WISDM dataset.**FIGURE 12.** Effect of the batch size on model performance for the WISDM dataset.**FIGURE 11.** Effect of the batch size on model performance for the UCI dataset.**FIGURE 13.** Effect of the number of epochs on the model accuracy with the UCI dataset.

critical. As illustrated in Figures 9 and 10, several common optimizers, such as Adam, RMSprop, SGD, AdaGrad, and AdaDelta, are empirically verified. The Adam optimizer-trained model has the best fitting effect and the steadiest gradient descent curve fluctuation. Hence, to train the CNN model, Adam is employed as the optimizer.

Figures 15 and 16 present the accuracies and losses induced by different numbers of epochs on the WISDM dataset.

2) IMPACT OF THE BATCH SIZE

In regard to DL, minibatch processing is a popular technique for training neural networks. The gradient descent process may slow down due to the optimization of the cumulative error over the entire training set and possibly lead to a local optimum for the corresponding model. If the error due to one sample only is optimized in one iteration, the gradient

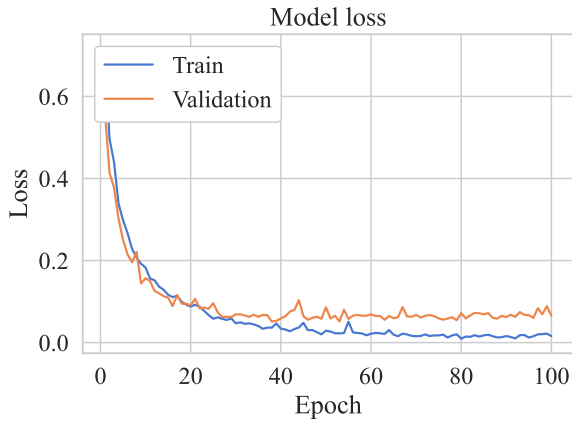


FIGURE 14. Effect of the number of epochs on the model loss with the UCI dataset.

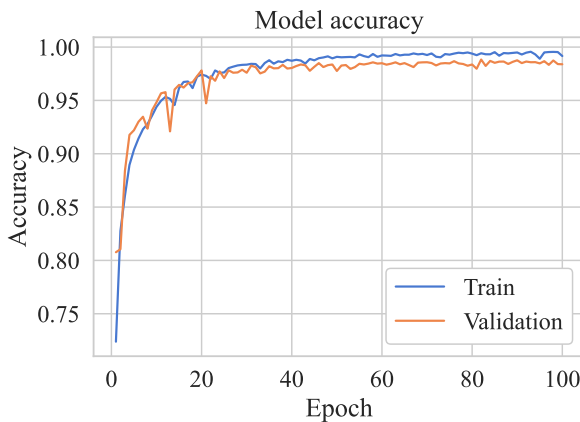


FIGURE 15. Effect of the number of epochs on the model accuracy with the WISDM dataset.

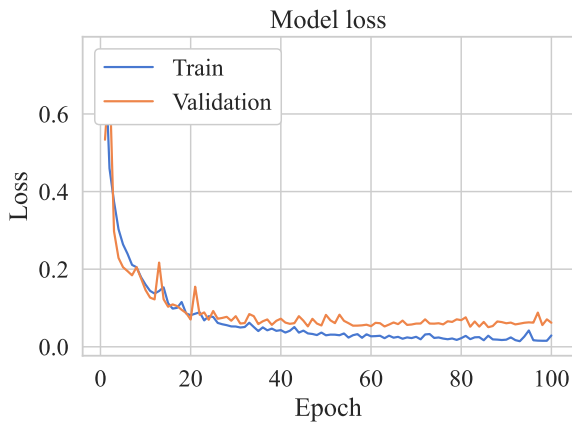


FIGURE 16. Effect of the number of epochs on the model loss with the WISDM dataset.

descent step may fluctuate dramatically, resulting in training difficulty. Figures 11 and 12 depict the accuracies obtained with the five different batch sizes. When the batch size is set to 192, the accuracy is at its maximum for the WISDM public dataset, and when the batch size is 128, the accuracy is at its maximum for our UCI dataset.

3) EFFECT OF THE NUMBER OF EPOCHS

The number of epochs is a type of hyperparameter that plays an important role in a DL model's training process. The total number of epochs to be used helps us determine whether the data have been overtrained. Figures 13 and 14 present the accuracies and losses obtained with different numbers of epochs on the UCI dataset. The validation set is used to minimize overfitting as much as possible. We stop the training procedure when the validation error is minimal.

H. COMPLEXITY ANALYSIS OF THE PROPOSED ARCHITECTURE

Suppose that n_0 = the number of input channels, n = the number of filters, $s_1 \times s_2$ = the size of each filter, and $m_1 \times m_2$ = Size of the output feature map. The complexity of the convolution layer = $\mathcal{O}(n_0 * s_1 * s_2 * n * m_1 * m_2)$. In the case of the deconvolutional layer in the AE, only the dimensionality is decreased by the downsampling factor, so there is no effect on the complexity. The polling layer is a fixed operation with no weighting factor. Fully connected layers - Input dimensions = m , and number of output dimensions = n . The number of parameters = $(m + 1) * n$. LSTM is local in terms of space and time; therefore, the overall complexity of an LSTM network per time step is equal to $\mathcal{O}(w)$, where w is the number of weights. Overall complexity = $\mathcal{O}(((n_0 * s_1 * s_2 * n * m_1 * m_2) + w) * i * e)$, where i is the input length and e is the number of epochs.

VI. DISCUSSIONS

In this study, we propose a DL framework by combining a convolutional AE and LSTM. The fully connected layer of the CNN increases the computational time of the model as the number of parameters increases. Additionally, CNNs are efficient in extracting features from labeled data, which are very rare in real scenarios. To overcome these shortcomings, we combine the convolution layer with an AE, and the output of the convolutional AE is given as the input of the LSTM module to extract temporal features and make the proposed DL architecture more accurate and effective in recognizing human activities. In the traditional ML method, feature engineering is a challenging and tedious job. In contrast, DL approaches are blessed with automatic feature learning characteristics. However, various DL approaches have their own merits and demerits. Hence, in this study, we consider the advantages of the convolution layer in combination with an AE for automatic feature extraction and for overcoming the overfitting issue. Moreover, sensor data streams are time series; hence, LSTM-based approaches with excellent sequential modeling capabilities are inherently appropriate. With the typical memory and computational resource restrictions, however, training an LSTM model on raw sensory data with a high sampling frequency is impossible. The suggested model not only avoids complicated data preprocessing and feature engineering techniques but also provides high recognition accuracy in an acceptable amount of computational time.

In this study, we mainly focus on smartphone-based HAR. Hence, for detailed experimentation, we consider smartphone-based public standard sensor data for exhaustive experiments. To prove the effectiveness of the proposed method, we also experiment with body-worn sensory data drawn from the “PAMAP2” and “OPPORTUNITY” public standard datasets and compare our results with those of state-of-the-art methods developed in other studies.

In the literature, convolutional AEs and LSTM are both popularly used for manifold data. Our proposed architecture is the combination of a convolutional AE and LSTM. However, in this paper, we do not show the experimental results obtained when using manifold data. In our future work, we can adopt manifold data to prove the effectiveness of our proposed framework.

VII. CONCLUSION

A novel DL approach in which a convolutional AE is followed by LSTM for HAR, namely, ConvAE-LSTM, is proposed in this paper. To establish the generalizability, potentiality, and efficacy of the suggested model, two standard smartphone sensor-based datasets (UCI and WISDM) and two standard body-worn sensor datasets (Opportunity and PAMAP2) are considered for experimentation. The proposed method achieves average precision, recall, F1-score, and accuracy values of 97%, 96.83%, 97.67%, and 98.14% on the UCI dataset and 98.17%, 98.33%, 98.17% and 98.67% on the WISDM dataset, respectively. Furthermore, we also explore the computational times of our proposed method and other commonly used DL approaches in the same experimental environment. The computational time of the proposed model is highly competitive with those of the other mentioned DL approaches. We also examine how several hyperparameters, such as the type of optimizer used, the number of epochs, and the batch size, affect the model performance. Finally, the model is trained with the best hyperparameters for the final design. To summarize, compared with the other tested DL approaches and two popularly used shallow ML approaches described in the HAR literature, the proposed ConvAE-LSTM model demonstrates consistently higher performance and has better generalization.

Despite the fact that much work has been done in this field, our findings show that many challenges remain unsolved, particularly in the area of activity recognition. Several aspects will be involved in future work. First, we will compare the proposed method with other recent DL-based methods and perform experiments on more available datasets by using other available classifiers. Second, in real-life applications, the applicability of the proposed method should be analyzed.

REFERENCES

- [1] J. Wannenburg and R. Malekian, “Physical activity recognition from smartphone accelerometer data for user context awareness sensing,” *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 12, pp. 3142–3149, Dec. 2017.
- [2] H. Bi, M. Perello-Nieto, R. Santos-Rodríguez, and P. Flach, “Human activity recognition based on dynamic active learning,” *IEEE J. Biomed. Health Informat.*, vol. 25, no. 4, pp. 922–934, Apr. 2021.
- [3] *World Population Ageing*, United Nation, New York, NY, USA, 2017.
- [4] H. Ghasemzadeh and R. Jafari, “Physical movement monitoring using body sensor networks: A phonological approach to construct spatial decision trees,” *IEEE Trans. Ind. Informat.*, vol. 7, no. 1, pp. 66–77, Feb. 2011.
- [5] P. Li, Y. Wang, Y. Tian, T.-S. Zhou, and J.-S. Li, “An automatic user-adapted physical activity classification method using smartphones,” *IEEE Trans. Biomed. Eng.*, vol. 64, no. 3, pp. 706–714, Mar. 2017.
- [6] Z. Chen, C. Jiang, and L. Xie, “A novel ensemble ELM for human activity recognition using smartphone sensors,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2691–2699, May 2019.
- [7] Y. Wang, X. Jiang, R. Cao, and X. Wang, “Robust indoor human activity recognition using wireless signals,” *Sensors*, vol. 15, no. 7, pp. 17195–17208, Jul. 2015.
- [8] C. A. Ronao and S.-B. Cho, “Human activity recognition with smartphone sensors using deep learning neural networks,” *Expert Syst. Appl.*, vol. 59, pp. 235–244, Oct. 2016.
- [9] F. Monteiro-Guerra, O. Rivera-Romero, L. Fernandez-Luque, and B. Caulfield, “Personalization in real-time physical activity coaching using mobile applications: A scoping review,” *IEEE J. Biomed. Health Informat.*, vol. 24, no. 6, pp. 1738–1751, Jun. 2020.
- [10] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, May 2011.
- [11] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones,” in *Proc. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn.*, 2013, pp. 1–3.
- [12] E. Bulbul, A. Cetin, and I. A. Dogru, “Human activity recognition using smartphones,” in *Proc. 2nd Int. Symp. Multidisciplinary Stud. Innov. Tech. (ISMSIT)*, Oct. 2018, pp. 1–6.
- [13] R.-A. Voicu, C. Dobre, L. Bajenaru, and R.-I. Ciobanu, “Human physical activity recognition using smartphone sensors,” *Sensors*, vol. 19, no. 3, p. 458, 2019.
- [14] C. Ronao and S.-B. Cho, “Deep convolutional neural networks for human activity recognition with smartphone sensors,” *Lect. Notes Comput. Sci.*, vol. 9492, pp. 46–53, Nov. 2015.
- [15] 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, p. 33, Jan. 2014.
- [16] L. Yao, Q. Z. Sheng, B. Benatallah, S. Dustdar, X. Wang, A. Shemshadi, and S. S. Kanhere, “WITS: An IoT-endowed computational framework for activity recognition in personalized smart Homes,” *Computing*, vol. 100, no. 4, pp. 369–385, Apr. 2018.
- [17] B. Cvetković, R. Szeklicki, V. Janko, P. Lutowski, and M. Luštrek, “Real-time activity monitoring with a wristband and a smartphone,” *Inf. Fusion*, vol. 43, pp. 77–93, Sep. 2018.
- [18] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, “Convolutional neural networks for human activity recognition using mobile sensors,” in *Proc. 6th Int. Conf. Mobile Comput., Appl. Services*, 2014, pp. 197–205.
- [19] O. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1192–1209, Jul. 2013.
- [20] T. Huynh and B. Schiele, “Analyzing features for activity recognition,” in *Proc. Joint Conf. Smart Objects Ambient Intell. Innov. Context-Aware Services, Usages Technol.*, New York, NY, USA, 2005, pp. 159–163.
- [21] J. B. 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. Conf. Artif. Intell.*, 2015, pp. 3995–4001.
- [22] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H. P. Tan, “Deep activity recognition models with triaxial accelerometers,” *CoRR*, vol. abs/1511.04664, pp. 1–15, Jan. 2015.
- [23] N. Y. Hammerla, S. Halloran, and T. Plötz, “Deep, convolutional, and recurrent models for human activity recognition using wearables,” in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1533–1540.
- [24] Y. Wang, H. Yao, and S. Zhao, “Auto-encoder based dimensionality reduction,” *Neurocomputing*, vol. 184, pp. 232–242, Apr. 2016.
- [25] S. Ryu, H. Choi, H. Lee, and H. Kim, “Convolutional autoencoder based feature extraction and clustering for customer load analysis,” *IEEE Trans. Power Syst.*, vol. 35, no. 2, pp. 1048–1060, Mar. 2020.

- [26] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 609–616.
- [27] A. Essien and C. Giannetti, "A deep learning model for smart manufacturing using convolutional LSTM neural network autoencoders," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6069–6078, Sep. 2020.
- [28] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, vol. 1. Cambridge, MA, USA: MIT Press, Dec. 2015, pp. 802–810.
- [29] O. S. Eyobu and D. Han, "Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network," *Sensors*, vol. 18, no. 9, p. 2892, Aug. 2018.
- [30] D. Thakur and S. Biswas, "Smartphone based human activity monitoring and recognition using ML and DL: A comprehensive survey," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 11, pp. 5433–5444, Mar. 2020.
- [31] 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.
- [32] Y. Tian and W. Chen, "MEMS-based human activity recognition using smartphone," in *Proc. 35th Chin. Control Conf. (CCC)*, Jul. 2016, pp. 3984–3989.
- [33] L. F. Mejia-Ricart, P. Helling, and A. Olmsted, "Evaluate action primitives for human activity recognition using unsupervised learning approach," in *Proc. 12th Int. Conf. Internet Technol. Secured Trans. (ICITST)*, Dec. 2017, pp. 186–188.
- [34] G. Ogbuabor and R. La, "Human activity recognition for healthcare using smartphones," in *Proc. Int. Conf. Mach. Learn. Comput.*, Feb. 2018, pp. 41–46.
- [35] F. Cruciani, C. Sun, S. Zhang, C. Nugent, C. Li, S. Song, C. Cheng, I. Cleland, and P. McCullagh, "A public domain dataset for human activity recognition in free-living conditions," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput.*, Oct. 2019, pp. 166–171.
- [36] A. Barna, A. K. M. Masum, M. E. Hossain, E. H. Bahadur, and M. S. Alam, "A study on human activity recognition using gyroscope, accelerometer, temperature and humidity data," in *Proc. Int. Conf. Electr., Comput. Commun. Eng. (ECCE)*, Feb. 2019, pp. 1–6.
- [37] J. Zhu, R. San-Segundo, and J. M. Pardo, "Feature extraction for robust physical activity recognition," *Hum.-Centric Comput. Inf. Sci.*, vol. 7, no. 1, p. 16, Dec. 2017.
- [38] H. Mazaar, E. Emary, and H. Onsi, "Evaluation of feature selection on human activity recognition," in *Proc. IEEE 7th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, Dec. 2015, pp. 591–599.
- [39] 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.
- [40] N. D. Nguyen, D. T. Bui, P. H. Truong, and G.-M. Jeong, "Position-based feature selection for body sensors regarding daily living activity recognition," *J. Sensors*, vol. 2018, Sep. 2018, Art. no. 9762098.
- [41] F. Attal, M. Dedabrishvili, L. Oukhellou, and Y. Amirat, "Physical human activity recognition using wearable sensors," *Sensors*, vol. 15, no. 12, pp. 31314–31338, 2015.
- [42] H. Ponce, M. Martínez-Villaseñor, and L. Miralles-Pechuán, "A novel wearable sensor-based human activity recognition approach using artificial hydrocarbon networks," *Sensors*, vol. 16, no. 7, p. 1033, Jul. 2016.
- [43] R. Jansi and R. Amutha, "A novel chaotic map based compressive classification scheme for human activity recognition using a tri-axial accelerometer," *Multimedia Tools Appl.*, vol. 77, no. 23, pp. 31261–31280, Dec. 2018.
- [44] S. Rosati, G. Balestra, and M. Knaflitz, "Comparison of different sets of features for human activity recognition by wearable sensors," *Sensors*, vol. 18, no. 12, p. 4189, Nov. 2018.
- [45] L. Köping, K. Shirahama, and M. Grzegorzec, "A general framework for sensor-based human activity recognition," *Comput. Biol. Med.*, vol. 95, pp. 248–260, Jan. 2018.
- [46] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 1, pp. 56–64, Jan. 2017.
- [47] X. Jiang, Y. Lu, Z. Lu, and H. Zhou, "Smartphone-based human activity recognition using CNN in frequency domain," in *Proc. APWeb/WAIM Workshops*, 2018, pp. 101–110.
- [48] I. Andrey, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Appl. Soft Comput.*, vol. 62, pp. 915–922, Jan. 2017.
- [49] F. Moya Rueda, R. Grzeszick, G. A. Fink, S. Feldhorst, and M. Ten Hompel, "Convolutional neural networks for human activity recognition using body-worn sensors," *Information*, vol. 5, no. 2, p. 26, 2018.
- [50] M. Gholamrezai and S. M. Taghi Almodarresi, "Human activity recognition using 2D convolutional neural networks," in *Proc. 27th Iranian Conf. Electr. Eng. (ICEE)*, Apr. 2019, pp. 1682–1686.
- [51] S. Dhanraj, S. De, and D. Dash, "Efficient smartphone-based human activity recognition using convolutional neural network," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Dec. 2019, pp. 307–312.
- [52] T. Zebin, P. J. Scully, N. Peek, A. J. Casson, and K. B. Ozanyan, "Design and implementation of a convolutional neural network on an edge computing smartphone for human activity recognition," *IEEE Access*, vol. 7, pp. 133509–133520, 2019.
- [53] B. Zhou, J. Yang, and Q. Li, "Smartphone-based activity recognition for indoor localization using a convolutional neural network," *Sensors*, vol. 19, no. 3, p. 621, Feb. 2019.
- [54] W. Qi, H. Su, C. Yang, G. Ferrigno, E. De Momi, and A. Aliverti, "A fast and robust deep convolutional neural networks for complex human activity recognition using smartphone," *Sensors*, vol. 19, p. 17, Jun. 2019.
- [55] J. A. Gamble and J. Huang, "Convolutional neural network for human activity recognition and identification," in *Proc. Int. Syst. Conf. (SysCon)*, 2020, pp. 1–7.
- [56] F. Cruciani, A. Vafeiadis, C. Nugent, I. M. P. Cleland, K. Votis, and R. Hamzaoui, "Feature learning for human activity recognition using convolutional neural networks," *CCF Trans. Pervasive Comput. Interact.*, vol. 2, no. 1, pp. 18–32, 2020.
- [57] C.-T. Yen, J.-X. Liao, and Y.-K. Huang, "Human daily activity recognition performed using wearable inertial sensors combined with deep learning algorithms," *IEEE Access*, vol. 8, pp. 174105–174114, 2020.
- [58] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones," *Mobile Netw. Appl.*, vol. 25, no. 2, pp. 743–755, Apr. 2020.
- [59] Y. Li, D. Shi, B. Ding, and D. Liu, "Unsupervised feature learning for human activity recognition using smartphone sensors," in *Mining Intelligence and Knowledge Exploration (Lecture Notes In Computer Science)*, vol. 8891. Cham, Switzerland: Springer, 2014, pp. 99–107.
- [60] M. Hasan and A. K. Roy-Chowdhury, "Continuous learning of human activity models using deep nets," in *Computer Vision*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 705–720.
- [61] B. Almaslukh, J. AlMuhtadi, and A. Artoli, "An effective deep autoencoder approach for online smartphonebased human activity recognition," *Int. J. Comput. Sci. Netw. Secur.*, vol. 17, no. 4, pp. 160–165, 2017.
- [62] D. Balabka, "Semi-supervised learning for human activity recognition using adversarial autoencoders," in *Proc. UbiComp/ISWC*, New York, NY, USA, 2019, pp. 685–688.
- [63] X. Gao, H. Luo, Q. Wang, F. Zhao, L. Ye, and Y. Zhang, "A human activity recognition algorithm based on stacking denoising autoencoder and lightGBM," *Sensors*, vol. 19, no. 4, p. 947, 2019.
- [64] T. Ozcan and A. Basturk, "Human action recognition with deep learning and structural optimization using a hybrid heuristic algorithm," *Cluster Comput.*, vol. 23, no. 4, pp. 2847–2860, Dec. 2020.
- [65] A. G. Prabono, B. N. Yahya, and S.-L. Lee, "Atypical sample regularizer autoencoder for cross-domain human activity recognition," *Inf. Syst. Frontiers*, vol. 23, no. 1, pp. 71–80, Feb. 2021.
- [66] K. D. Garcia, C. R. de Sá, M. Poel, T. Carvalho, J. Mendes-Moreira, J. M. P. Cardoso, A. C. P. L. F. de Carvalho, and J. N. Kok, "An ensemble of autonomous auto-encoders for human activity recognition," *Neurocomputing*, vol. 439, pp. 271–280, Jun. 2021.
- [67] M. T. H. Tonmoy, S. Mahmud, A. K. M. M. Rahman, M. A. Amin, and A. A. Ali, "Hierarchical self attention based autoencoder for open-set human activity recognition," in *Proc. Adv. Knowl. Discovery Data Mining*, P. K. Karlapalem, H. Cheng, N. Ramakrishnan, R. K. Agrawal, P. K. Reddy, D. J. Srivastava, and A. P. T. Chakraborty, Eds., 2021, pp. 1–12.

- [68] C. Geng and J. Song, "Human action recognition based on convolutional neural networks with a convolutional auto-encoder," in *Proc. 5th Int. Conf. Comput. Sci. Autom. Eng.*, 2016, pp. 933–938.
- [69] A. A. Varamin, E. Abbasnejad, Q. Shi, D. C. Ranasinghe, and H. Rezaatoghli, "Deep auto-set: A deep auto-encoder-set network for activity recognition using wearables," in *Proc. 15th EAI Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services*, New York, NY, USA, Nov. 2018, pp. 246–253.
- [70] T. Zebin, M. Sperrin, N. Peek, and A. J. Casson, "Human activity recognition from inertial sensor time-series using batch normalized deep LSTM recurrent networks," in *Proc. 40th Annu. Int. Conf. Eng. Med. Biol. Soc. (EMBC)*, 2018, pp. 1–4.
- [71] S. Yu and L. Qin, "Human activity recognition with smartphone inertial sensors using bidir-LSTM networks," in *Proc. 3rd Int. Conf. Mech., Control Comput. Eng. (ICMCCE)*, Sep. 2018, pp. 219–224.
- [72] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang, "Deep residual bidir-lstm for human activity recognition using wearable sensors," *Math. Problems Eng.*, vol. 2018, Dec. 2018, Art. no. 7316954.
- [73] F. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, Jan. 2016.
- [74] S. Deep and X. Zheng, "Hybrid model featuring CNN and LSTM architecture for human activity recognition on smartphone sensor data," in *Proc. 20th Int. Conf. Parallel Distrib. Comput., Appl. Technol. (PDCAT)*, Dec. 2019, pp. 259–264.
- [75] H. Wang, J. Zhao, J. Li, L. Tian, P. Tu, T. Cao, Y. An, K. Wang, and S. Li, "Wearable sensor-based human activity recognition using hybrid deep learning techniques," *Secur. Commun. Netw.*, vol. 2020, p. 12, Jan. 2020.
- [76] R. Mutegeki and D. S. Han, "A CNN-LSTM approach to human activity recognition," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIIC)*, Feb. 2020, pp. 362–366.
- [77] G. Ercolano and S. Rossi, "Combining CNN and LSTM for activity of daily living recognition with a 3D matrix skeleton representation," *Intell. Service Robot.*, vol. 14, no. 2, pp. 175–185, Apr. 2021.
- [78] W. Ye, J. Cheng, F. Yang, and Y. Xu, "Two-stream convolutional network for improving activity recognition using convolutional long short-term memory networks," *IEEE Access*, vol. 7, pp. 67772–67780, 2019.
- [79] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, "InnoHAR: A deep neural network for complex human activity recognition," *IEEE Access*, vol. 7, pp. 9893–9902, 2019.
- [80] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56855–56866, 2020.
- [81] H. Zou, Y. Zhou, J. Yang, H. Jiang, L. Xie, and C. J. Spanos, "DeepSense: Device-free human activity recognition via autoencoder long-term recurrent convolutional network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
- [82] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. R. Del 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, Jan. 2009.
- [83] D. Roggen, A. Calatroni, M. Rossi, T. Holleczeck, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirk, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. D. R. Millán, "Collecting complex activity datasets in highly rich networked sensor environments," in *Proc. 7th Int. Conf. Netw. Sens. Syst. (INSS)*, 2010, pp. 233–240.
- [84] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster, "Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection," in *Wireless Sensor Networks*, R. Verdone, Ed. Berlin, Germany: Springer, 2008, pp. 17–33.
- [85] J. W. Lockhart, G. M. Weiss, J. C. Xue, S. T. Gallagher, A. B. Grosner, and T. P. Pulickal, "Design considerations for the WISDM smart phone-based sensor mining architecture," in *Proc. 5th Int. Workshop Knowl. Discovery Sensor Data*, New York, NY, USA, 2011, pp. 25–33.
- [86] M. Bachlin, M. Plotnik, D. Roggen, I. Maidan, J. M. Hausdorff, N. Giladi, and G. Tröster, "Wearable assistant for Parkinson's disease patients with the freezing of gait symptom," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 436–446, Nov. 2010.
- [87] A. Bevilacqua, K. MacDonald, A. Rangrej, V. Widjaya, B. Caulfield, and T. Kechadi, "Human activity recognition with convolutional neural networks," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, Sep. 2018, pp. 541–552.
- [88] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.
- [89] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE Pervasive Comput.*, vol. 16, no. 4, pp. 62–74, Oct. 2017.
- [90] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for time series classification," *Neural Netw.*, vol. 116, pp. 237–245, Aug. 2019.
- [91] W. Wu and Y. Zhang, "Activity recognition from mobile phone using deep CNN," in *Proc. Chin. Control Conf. (CCC)*, Jul. 2019, pp. 7786–7790.
- [92] X. Liu, Q. Zhou, J. Zhao, H. Shen, and X. Xiong, "Fault diagnosis of rotating machinery under noisy environment conditions based on a 1-D convolutional autoencoder and 1-D convolutional neural network," *Sensors*, vol. 19, no. 4, p. 972, Feb. 2019.
- [93] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. 19th Int. Conf. Neural Inf. Process. Syst.*, Cambridge, MA, USA, USA, 2006, pp. 153–160.
- [94] F. Chollet. (2015). *Layer Wrappers*. [Online]. Available: <https://keras.io/layers/wrappers/timedistributed>
- [95] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Artificial Neural Networks and Machine Learning*, vol. 6791. Berlin, Germany: Springer-Verlag, 2011, pp. 52–59.
- [96] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [97] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [98] G. Cawley and N. Talbot, "Efficient approximation leave-one-out cross-validation for kernel logistic regression," *Mach. Learn.*, vol. 71, pp. 243–264, Jan. 2008.
- [99] M. Ullah, H. Ullah, S. D. Khan, and F. A. Cheikh, "Stacked LSTM network for human activity recognition using smartphone data," in *Proc. 8th Eur. Workshop Vis. Inf. Process. (EUVIP)*, Oct. 2019, pp. 175–180.
- [100] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Statist. Assoc.*, vol. 32, no. 200, pp. 675–701, Dec. 1937.
- [101] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *Ann. Math. Statist.*, vol. 11, no. 1, pp. 86–92, 1940.
- [102] R. A. Armstrong, "When to use the Bonferroni correction," *Ophthalmic Physiol. Opt.*, vol. 34, no. 5, pp. 502–508, 2014.



DIPANWITA THAKUR (Member, IEEE) has been an Assistant Professor with the Department of Computer Science and Engineering, Banasthali Vidyapith, Rajasthan, India, since July 2008. Prior to that, she was a Technical Support Engineer at Netgear Process, from 2007 to 2008. She has published various research papers in reputed conferences and journals. She is currently working in different aspects of human activity recognition. Her research interests include machine learning in healthcare, software defined networking, and network virtualization. She is a member of ACM.



SUPARNA BISWAS (Senior Member, IEEE) received the M.E. and Ph.D. degrees from Jadavpur University, India. She had been an ERASMUS MUNDUS Postdoctoral Research Fellow in cLINK project at Northumbria University, Newcastle, U.K., from 2014 to 2015. She is currently an Associate Professor and the Head of the Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, India. She has authored a number of research papers in reputed journals, conferences, and book chapters of international repute. She has delivered a number of invited lectures and tutorials in national and international conferences, workshops, and webinars. She is handling two funded research projects as a PI and a Co-PI on smart healthcare. She has successfully guided two Ph.D. scholars and guiding four more registered Ph.D. scholars. Her research interests include the Internet of Things, machine learning, security, and healthcare. Her recent activities include the session chair in international conferences, a lead editor in edited volumes of reputed publisher, and the lead organizing chair in international conference.



EDMOND S. L. HO received the B.Sc. degree (Hons.) in computer science from Hong Kong Baptist University, in 2003, the M.Phil. degree in computer science from the City University of Hong Kong, in 2006, and the Ph.D. degree from The University of Edinburgh, in 2010. He is currently an Associate Professor with the Department of Computer and Information Sciences, Northumbria University, Newcastle, U.K. Prior to this, he was a Research Assistant Professor with the Department of Computer Science, Hong Kong Baptist University. His research interests include computer graphics, computer vision, motion analysis, and machine learning.



SAMIRAN CHATTOPADHYAY (Senior Member, IEEE) received the B.Tech. and M.Tech. degrees from the Department of Computer Science and Engineering, IIT Kharagpur, in 1987 and 1989, respectively, and the Ph.D. degree from the Department of Computer Science and Engineering, Jadavpur University, in 1993. For more than 30 years, he worked as a Faculty Member at Jadavpur University. He is currently a professor at IAI, TCG CREST, a Professor (on lien) at Jadavpur University and a Visiting Fellow at the University of Northumbria, Newcastle upon Tyne, U.K. He has more than two decades of experience of serving reputed industry houses. He has published about 180 technical papers in international journals and conferences in the areas of wireless networks, network security, and machine learning applications. He has coauthored and edited more than ten books. His current research interests include network security, machine learning, wireless networks, and pervasive computing.

...