IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 8, 2021, accepted October 31, 2021, date of publication November 8, 2021, date of current version November 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3125733

Industry 4.0-Oriented Deep Learning Models for Human Activity Recognition

SAEED MOHSEN [1], **AHMED ELKASEER** [2,3], **AND STEFFEN G. SCHOLZ** [2,4]

[1] Department of Electronics and Communications Engineering, Al-Madina Higher Institute for Engineering and Technology, Giza 12947, Egypt
[2] Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, 76344 Karlsruhe, Germany
[3] Department of Production Engineering and Mechanical Design, Faculty of Engineering, Port Said University, Port Fuad 42526, Egypt
[4] College of Engineering, Future Manufacturing Research Institute, Swansea University, Swansea SA1 8EN, U.K.

Corresponding author: Ahmed Elkaseer (ahmed.elkaseer@kit.edu)

This work was supported in part by the Karlsruhe Nano Micro Facility (KNMFi, www.knmf.kit.edu), a Helmholtz Research Infrastructure at Karlsruhe Institute of Technology (KIT, www.kit.edu) and in part by the Helmholtz Research Programme MSE (Materials Systems Engineering) at KIT. The authors would like to acknowledge the support provided by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

ABSTRACT According to the Industry 4.0 vision, humans in a smart factory, should be equipped with formidable and seamless communication capabilities and integrated into a cyber-physical system (CPS) that can be utilized to monitor and recognize human activity via artificial intelligence (e.g., deep learning). Recent advances in the accuracy of deep learning have contributed significantly to solving the human activity recognition issues, but it remains necessary to develop high performance deep learning models that provide greater accuracy. In this paper, three models: long short-term memory (LSTM), convolutional neural network (CNN), and combined CNN-LSTM are proposed for classification of human activities. These models are applied to a dataset collected from 36 persons engaged in 6 classes of activities – downstairs, jogging, sitting, standing, upstairs, and walking. The proposed models are trained using TensorFlow framework with a hyper-parameter tuning method to achieve high accuracy. Experimentally, confusion matrices and receiver operating characteristic (ROC) curves are used to assess the performance of the proposed models. The results illustrate that the hybrid model CNN-LSTM provides a better performance than either LSTM or CNN in the classification of human activities. The CNN-LSTM model provides the best performance, with a testing accuracy of 97.76%, followed by the LSTM with a testing accuracy of 96.61%, while the CNN shows the least testing accuracy of 94.51%. The testing loss rates for the LSTM, CNN, and CNN-LSTM are 0.236, 0.232, and 0.167, respectively, while the precision, recall, $F1$-Measure, and the area under the ROC curves ($\mathrm{AUC_S}$) for the CNN-LSTM are 97.75%, 97.77%, 97.76%, and 100%, respectively.

INDEX TERMS Deep learning, convolutional neural network (CNN), long short-term memory (LSTM), human activity recognition, Industry 4.0.

I. INTRODUCTION

In the domain of Industry 4.0, the interaction between humans/workers in terms of their activities and the physical environment has changed substantially, but remains crucial for the synergetic integration of the intelligent manufacturing assets [1], [2]. Particularly in smart factories, recognizing and classifying human activities helps to evaluate human performance and thus their overall efficiency in production systems. From this perspective, in the era of Industry 4.0, artificial intelligence (AI) plays an important role in recognizing and evaluating human activities [3], [4]. In the last decade, human activity recognition has become a popular topic for research due to its importance in many fields, such as healthcare, sports, and fitness [5]–[10], interactive gaming, human-computer interaction, remote monitoring systems, and smart manufacturing [11].

In other applications, wearable accelerometers are used to measure human activity for remotely communicating between patients and hospitals [12]. However, the low accuracy of these accelerometers is a challenging problem yet to be fully overcome [13], [14]. Many traditional machine learning approaches have been proposed for accurately recognition of a human activity [15]–[18], but do not always achieve a satisfactory level of accuracy [19], [20].

Deep learning models such as recurrent neural networks ($\mathrm{RNN_S}$), long short-term memory networks ($\mathrm{LSTM_S}$), and convolutional neural networks ($\mathrm{CNN_S}$) present effective

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

solutions to overcome the problem of low accuracy. These models are already useful in such fields as speech recognition [21], image processing [22], and language modelling [23], and are applicable for recognizing human activities.

In the literature, several deep learning models have been introduced for classification of human activities. In [24], Pienaar and Malekian proposed a design of a LSTM-RNN model for daily life activities. The authors utilized a regularization method to improve the computations to process a huge WISDM dataset [25] and achieved an overall accuracy of 94%. However, the performance of the developed model was evaluated using only two evaluation metrics, namely confusion matrix and learning curve. In [26], Hammarela *et al.* introduced a bi-directional LSTM model using inertial sensors to classify a large number of human activities. This model was applied on the Opportunity dataset [27] and had a $F1$-Measure of 92.7%. Cruciani *et al.* [28] presented a CNN model for human activity recognition that was tested on the UCI-HAR dataset available in [29] with an achieved accuracy of 91.98%. This work was evaluated using a variety of evaluation metrics. In [30], Xia *et al.* proposed a LSTM-CNN model for human activity recognition. This model was also applied to the huge WISDM dataset [25] and achieved a maximum accuracy of 95.85%. However, the computational time consumed for the training phase was noticeable. Ordonez and Roggen [31] presented a model with slightly simple architecture to recognize human activities. This model utilized a combination of a ConvLSTM model based on seven inertial measurement units ($\mathrm{IMU_S}$) and twelve accelerometers. It classified five activities using the Skoda dataset [32] and achieved a $F1$-Measure of 95.8%.

Alani *et al.* [33] proposed LSTM, CNN, and CNN-LSTM models to classify imbalanced data for human activity recognition. These models were applied on the SPHERE dataset [34] and achieved accuracies of 92.98%, 93.55%, and 93.67%, respectively. This work dealt with twenty different human activities, but the performance evaluation of the models was limited to a single metric. In [35], Alzantot *et al.* applied a LSTM model to recognize human activities, which used to distinguish between synthesized and real data, but it had a high number of training parameters led to an architecture complexity, the accuracy achieved was quite low.

Researchers [36]–[39] have presented LSTM and CNN deep learning models to recognize human activities in daily living. In [36], Alsheikh *et al.* achieved an accuracy of 86.6%, but did not describe which dataset was used for the test. Shakya *et al.* [37] implemented RNN and CNN models, used the Actitracker, and Shoaib SA datasets, which were divided randomly, and achieved accuracies of 81.74% and 92.22%, respectively. An LSTM architecture was presented in [38], which achieved an accuracy of 92.1% on an unspecified test dataset split. Mekruksavanich *et al.* [39] also applied a LSTM model, achieving an accuracy of 96.2%, and $F1$-Measure of 96.3%.

Agarwal *et al.* [40] introduced a RNN-LSTM model to recognize human activity, using the WISDM dataset. The authors utilized only two response metrics for the performance evaluation of the model, and achieved an accuracy of 95.78%. Cipolla *et al.* [41] proposed a LSTM model to classify human activities using the SPHERE dataset. The developed model showed robust ability to deal with unbalanced classes. The model was applied to five activities and achieved a classification accuracy of 83.2%. Zhao *et al.* [42] implemented a bi-directional LSTM model to identify human activities, for which a number of sensors were used to collect the datasets. The main drawback of this model was the long time consumed for training the model, and thus it wasn't so convenient for real-time applications.

CNN gained a lot of attention over the years and is often used in applications of image classification [43], text analysis [44], and natural language processing [45]. Xu *et al.* [46] trained a CNN model on a randomly chosen 70% of a dataset, and then used the model to evaluate the remaining 30%, achieving an accuracy of 91.97%. In [47], Ignatov applied a CNN model for human activity recognition. The accuracy reached 93.32% and 90.42% for the training and testing datasets, respectively. Huang *et al.* [48] introduced an architecture of two cascaded $\mathrm{CNN_S}$ and used a cross validation method to achieve an $F1$-Measure of 84.6%.

Looking at the reviewed literature one can argue that although a number of research studies attempted to develop accurate models for human activities recognition, there is still a wide margin of improvement to obtain. In this context, in this paper, three models of deep learning such as LSTM, CNN, and CNN-LSTM are proposed to predict human activities with the overarching aim of improving/ increasing the classification accuracy of the proposed models by introducing a hyper-parameter tuning method. In this regard, the main objective of applying the proposed deep learning models is achieving high accuracy to recognize human activities. Therefore, a k-fold cross-validation technique is implemented to reach high performance of testing accuracy. The models are trained and tested on the dataset available from the Wireless Sensor Data Mining (WISDM) Lab [49].

The main contributions of the paper are the following:

- Implementing the LSTM, CNN, and CNN-LSTM models to classify human activities;
- Achieving maximum testing accuracy of the models with a hyper-parameter tuning method on a central processing unit (CPU);
- Evaluating the performance of the three proposed models using different evaluation metrics with the WISDM dataset. Besides, the performance of the LSTM was evaluated using the DaLiAc dataset, which entails a large number of human activities;
- Enhancing and validating the performance of the proposed deep learning models using the k-fold cross-validation technique.

The remainder of the paper is organized as follows: Section II introduces theoretical background concerning the

LSTM and CNN. Section III presents the proposed deep learning models. Section IV describes the evaluation metrics utilized with the proposed models. The experimental results are illustrated in section V. Section VI discusses the results. Finally, conclusions from the work are drawn in section VII.

II. THEORITICAL BACKGROUND

A. LONG SHORT-TERM MEMORY (LSTM)

As a special type of recurrent neural networks (RNN), LSTM is a popular deep learning approach [50]. RNN has the vanishing gradient problem [51], [52], which LSTM was devised to solve. LSTM is suitable for processing time sequences. LSTM layers include memory blocks recurrently connected in a memory cell. Figure 1 presents the architecture of an LSTM unit, which consists of a memory cell and three gates; a forget gate, an input gate, and an output gate [53]. The memory cell remembers values over arbitrary time intervals. The three gates accept and reject information passing through the cell. In Figure 1, the forget gate decides which information will be remembered from the previous cell state C_{t-1}. This decision is taken via a sigmoid activation function (σ). The output of this sigmoid is $f\ (t)$. If this output has a value of 1, the data will be passed into the model; if the output value is 0, the data will not be passed through the model. The input of this sigmoid is the current input x_t and previous hidden state h_{t-1}. The input gate decides what new information will be stored in the current cell state C_t.

The input gate has a sigmoid activation function to update the cell state. This function has a range from 0 to 1. The output of this sigmoid is $i\ (t)$, which is multiplied with a *tanh* activation function that outputs a new cell state $\hat{c}(t)$. The resultant of the multiplication is added to the current cell state C_t. Finally, the output gate presents the information to the next cell. It also has a sigmoid activation function to determine which parts of the current cell state require outputs; the cell state is processed by a *tanh* to obtain values between -1 and 1. Then, the values are multiplied by a sigmoid function with an output $o(t)$ to obtain the output values, where h_t is the current hidden state output of the current cell. The forget gate, input gate, and output gate are computed using Equations (1), (2), and (3), respectively. The $\hat{c}(t)$, C_t, and h_t parameters are estimated from Equations (4), (5), and (6), respectively. Whereby W_f, W_i, and W_o are the weights for the forget gate, input gate, and output gate, respectively. These weights will be learned in a training process. b_f, b_i, and b_o are the bias parameters for the forget gate, input gate, and output gate, respectively.

$$f\ (t) = \sigma\left(W_f[h_{t-1},x_t] + b_f\right) \tag{1}$$

$$i\ (t) = \sigma\ (W_i[h_{t-1}, x_t] + b_i) \tag{2}$$

$$o\ (t) = \sigma\ (W_o[h_{t-1},x_t] + b_o) \tag{3}$$

$$\hat{c}\ (t) = tanh(W_g[h_{t-1}, x_t] + b_g) \tag{4}$$

$$C_t = f\ (t) \times C_{t-1} + i\ (t) \times g_t \tag{5}$$

$$h_t = o\ (t) \times tanh(C_t) \tag{6}$$

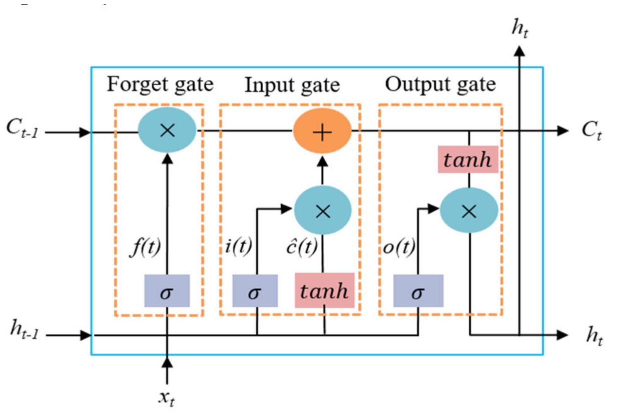


FIGURE 1. A Long short-term memory (LSTM) unit architecture.

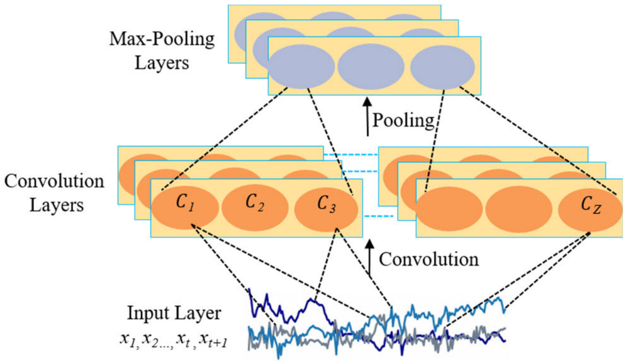


FIGURE 2. An example of a one-dimensional convolutional neural network architecture.

B. CONVOLUTIONAL NEURAL NETWORK (CNN)

CNN is a feed forward neural network (FNN) that includes an input layer, an output layer, and hidden layers. These hidden layers are represented by convolutional layers combined with max-pooling layers. The convolution layers are the most significant of the layers in a CNN. In these layers, a configuration of filters is used to smooth input signals and generate feature maps of a dataset. These maps are activated as a result of a convolution operation with a kernel over the dataset. Figure 2 shows an example of a one-dimensional convolutional neural network architecture. An input layer with feature signals $x_1 \ldots x_t, x_{t+1}$ are represented by X_t. These signals are connected to convolutional layers with a kernel size K. Multiple convolutional layers help feature extraction from an input data to greater levels of abstraction. Max-pooling layers coming after the convolutional layers enhance the extracted feature signals by reducing its dimension using a pooling function. A feature maps extraction is calculated using Equation (7), where C_z is the feature map of a convolutional layer, X_t is input feature map, σ is the sigmoid activation function, $*$ is the convolutional operator, and $W_{t,z}$ represents the weight vector, which connects the t^{th} input signal to the z^{th} feature signal.

$$C_z = \sigma\left[\sum_{t=1}^{T} X_t * W_{t,z}\right] \tag{7}$$

III. PROPOSED MODELS

The LSTM, CNN, and combination CNN-LSTM models are employed here for classifying human activities. The proposed models are selected because of their approved robust performance. Especially, the LSTM is found to be suitable for processing time series data [54], the CNN is suitable for processing spatial data [55], and the CNN-LSTM is suitable for processing both temporal and spatial data. Specifically, LSTM architectures have been successfully used with time series [56]. These architectures are memory extensions for the RNN with the advantage that use of the LSTM avoids the vanishing gradient problem [57]. CNN architectures have been applied to time series to extract significant patterns by reducing noise [58] with the advantage that use of CNN structures allows learning of complex input features [59]. Finally, LSTM architectures were combined with CNN architectures [60] to exploit the advantages of both architectures and extract both temporal and spatial data.

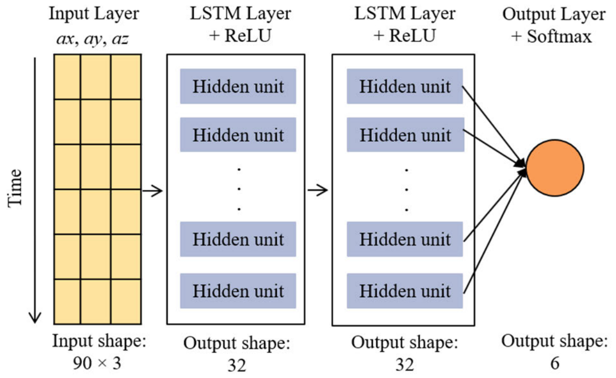


FIGURE 3. The proposed LSTM model architecture.

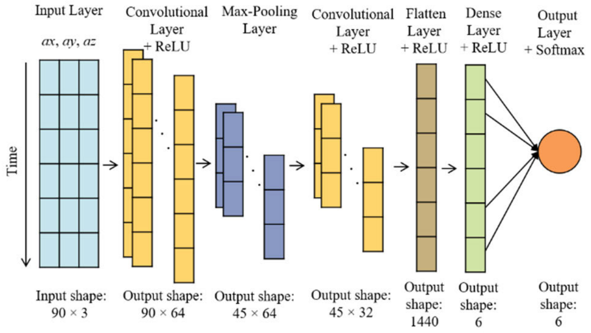


FIGURE 4. The proposed CNN model architecture.

A. LONG SHORT-TERM MEMORY (LSTM) MODEL

The proposed LSTM model architecture is shown in Figure 3. This architecture is implemented using TensorFlow framework. It consists of an input layer, two LSTM layers, and an output layer. The input layer has a shape: number of samples, 90 time steps, and 3 features. These features are a_x, a_y, and a_z, where a_x is the acceleration in the x-axis, a_y is the acceleration in the y-axis, and a_z is the acceleration in the z-axis. The two LSTM layers are utilized to extract the time features in the data sequence. The two LSTM layers are stacked to add depth to the model and increase its stability and accuracy. Each LSTM layer has 32 hidden units [61], and uses a rectified linear unit (ReLU) activation function to increase the robustness of the model. The output layer consists of 6 neurons with a softmax function used as activation to obtain the classes. In this model, the batch size is optimized to 64 with training epochs of 50 and the learning rate is set to 0.0025. This rate determines the speed of the model. In addition, the optimizer is the Adam optimization algorithm, which finds the optimum weights for this model, minimizes the errors, and maximizes the training accuracy [62]. Furthermore, a regularization technique, based on a cross-entropy loss function, is implemented to prevent the model from over-fitting [63].

B. CONVOLUTIONAL NEURAL NETWORK (CNN) MODEL

Figure 4 illustrates the proposed CNN model architecture. The model consists of an input layer, a one dimension (1-D) convolutional layer, a 1-D max-pooling layer, another 1-D convolutional layer, a flatten layer, a dense layer, and an output layer. Firstly, the input layer receives three channels of acceleration data, a_x, a_y, and a_z. The input shape of the channels has a width of 90, a height of 1, and 3 features. Secondly, the 1-D convolutional layer is applied to each input channel separately with a ReLU activation function. It is utilized to extract/capture spatial features. This layer has 64 filters, a kernel size of 5, and a stride of 1. Thirdly, a max-pooling layer is utilized to reduce the complexity of the convolutional output, performing a down sampling operation. The max-pooling layer has a pool size of 5 and a stride of 2. Fourthly, the second 1-D convolutional layer has a configuration of 32 filters, a kernel size of 5, a stride of 1, and uses a ReLU function. This layer is added to enable the model to detect higher level features that are missed in the first convolutional layer. Fifthly, a flatten layer is used to flatten the output of the convolutional layer. Sixthly, the dense fully connected layer is configured with 6 neurons and its activation function is a ReLU. Finally, the output layer is connected to a softmax activation function, which is used to reduce the output to six activity classes. In this model, a cross-entropy loss function is used to measure the error between the prediction and the true values and the optimizer is Adam. This model is trained using a batch size of 64 for 50 training epochs, with a learning rate of 0.0025.

C. CNN-LSTM MODEL

The proposed CNN-LSTM architecture is presented in Figure 5. It consists of an input layer, two staked 1-D convolutional layers, two subsequent LSTM layers, a dense layer, and an output layer. The input layer has three channels of acceleration data a_x, a_y, a_z. Its input shape has 90 time steps, height of 1, and 3 features. The two convolutional layers are

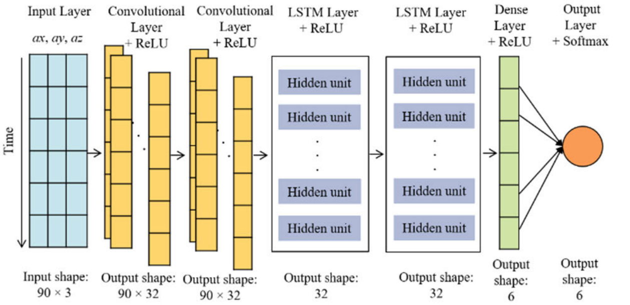


FIGURE 5. The proposed CNN-LSTM model architecture.

configured with 32 filters, a kernel size of 5, and a stride of 1. They are followed by the two stacked LSTM layers; 32 hidden units are used for each LSTM. In addition, a dense layer is applied with 6 neurons to pass the classes. All layers contain a ReLU nonlinear activation function. Finally, the output layer has 6 neurons to classify six classes of human activities and uses a softmax activation function to distinguish these classes. In this model, hyper-parameters are configured to 50 training epochs, 64 batch size, and 0.0025 learning rate. For optimizing this model, the Adam optimizer is utilized. A loss function is used depending on the cross-entropy.

D. DATASET DESCRIPTION

The dataset used for the classification is from the Wireless Sensor Data Mining (WISDM) Lab [49], with 1,098,207 samples of various activities including downstairs, jogging, sitting, standing, upstairs, and walking, and the sample percentages for each activity are 9.1%, 31.2%, 5.5%, 4.4%, 11.2%, and 38.6%, respectively. The dataset was collected from 36 persons using an Android mobile phone, which contains a built-in accelerometer placed in a front trouser pocket. The dataset readings are taken with a sampling rate of 20 Hz. The dataset has 6 attributes with information related to the activities of the users' activities: time, and x-, y-, and z-accelerations. The dataset is divided into a training set (80%) and a test set (20%), with the test set used to evaluate the proposed models.

The WISDM dataset is chosen as it contains a considerable variety of daily life activities as well as it has sufficient data with a sample size of 1,098,207 samples for training the proposed models. The dataset is split into 80% for training and 20% for testing using Scikit-learn framework. A probability sampling method is utilized for a random distribution of the dataset, which offers a robust training process for the models and thus enhances their performance [64].

Figure 6 demonstrates a sample of the acceleration data collected from a single person over a period of time. In this figure, three acceleration signals "raws" "a_x, a_y, and a_z" as functions in terms of the gravity acceleration (g) are illustrated. The blue raw represents the acceleration in the x-direction a_x, while the green and orange raws represent the a_y, and a_z which are the accelerations in the y- and z-direction, respectively. These signals are observed for 22,000 seconds. The signals have an amplitude range roughly from -g to g.

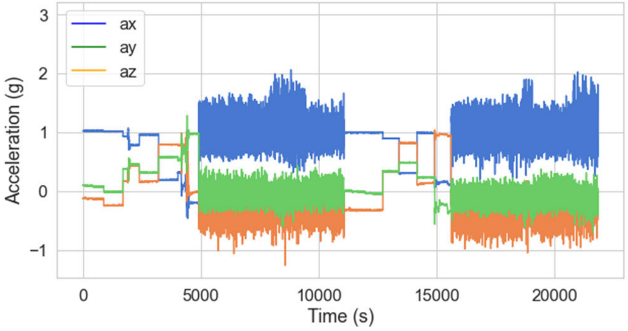


FIGURE 6. A sample of the acceleration detected for one person.

IV. EVALUATION METRICS

Different evaluation metrics are used after the training process to evaluate the performance of the proposed models [65]. These metrics include accuracy, precision, recall, and $F1$-Measure. These metrics are based on statistical values which are the results of a confusion matrix: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP represents an outcome in which a model correctly predicts the positive class, and TN represents an outcome where a model correctly predicts the negative class. FP means a model wrongly predicts the positive class, and FN presents a model wrongly predicts the negative class. The confusion matrix is also used to determine the ability of a model to classify multi-classes accurately with the models using the testing dataset to compare an output predicted class with a true class.

The first metric, Accuracy, represents the ratio of correct predictions to the total predictions of the testing dataset. It is calculated using Equation (8). The second metric, Precision, is the ratio of correct predictions of a specific class activity to the total predictions of the same-class in the testing dataset. Precision is mathematically represented in Equation (9). The third metric is the Recall, the ratio of accurately classified positives of a specific class to the total number of true class activities in the test dataset. Recall is also known as Sensitivity and can be calculated using Equation (10). The fourth metric is $F1$-Measure, which is the average of the precision and recall with a weight of 2, see Equation (11). It is also known as the balanced $F1$-Score. It takes both false positives (FP) and false negatives (FN) into consideration. Thus, it is a more useful metric for an evaluation than the accuracy metric. The worst possible value of all four metrics is 0%, and the best is 100%.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

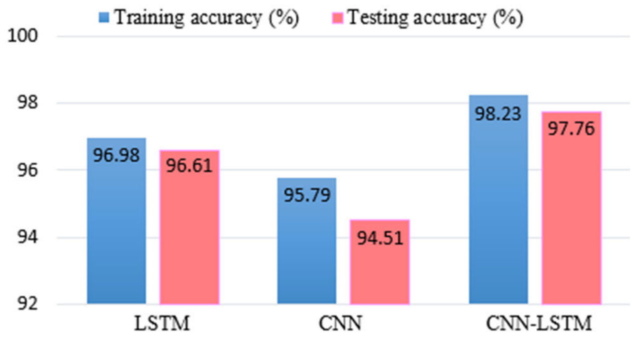


FIGURE 7. Training and testing accuracy of the proposed models.

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F1Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{11}$$

Area under the ROC curve (AUC) is also used as an evaluation metric for the models. It is defined as the integral of the true positive rate (TPR) multiplied by the false positive rate (FPR). TPR, FPR, and AUC are described in Equations (12), (13), and (14). The AUC value is always between 0 and 1. A high value of AUC implies a model is capable of distinguishing between classes of human activity. So, the model with the largest area under the ROC curve, is the best model for a classification. The TPR is the ratio of the number of true positives (TP) to the sum of true positives and false negatives ($TP + FN$). The FPR is the ratio of the number of false positives (FP) to the sum of false positives and true negatives ($FP + TN$). FPR is also known as 1-Specificity. The utilized evaluation metrics "accuracy, precision, recall, $F1$-Measure, and area under the curve" are chosen due to their effectiveness in analyzing and evaluating the performance of the proposed models [66].

$$TPR = \frac{TP}{TP + FN} \tag{12}$$

$$FPR = \frac{FP}{FP + TN} \tag{13}$$

$$AUC = \int_0^1 TPR \, d(FPR) \tag{14}$$

V. EXPERIMENTAL RESULTS

Experimentally, the proposed models are implemented using Python programming language in a Jupyter Notebook environment. The models are executed on a personal computer with Microsoft Windows 10 operating system, Intel Core i3, 4 CPU$_S$, 2.2 GHz processor, and 4 GB RAM memory. Figure 7 shows the training and testing accuracy of the models. The achieved training and testing accuracy of the LSTM are 96.98% and 96.61%, respectively, for the CNN, 95.79% and 94.51%, respectively, and for the CNN-LSTM 98.23% and 97.76%, respectively, showing that the CNN-LSTM has the highest accuracy.

TABLE 1. Training and testing loss rates of the proposed models.

Model	Training Loss	Testing Loss
LSTM	0.219	0.236
CNN	0.181	0.232
CNN-LSTM	**0.154**	**0.167**

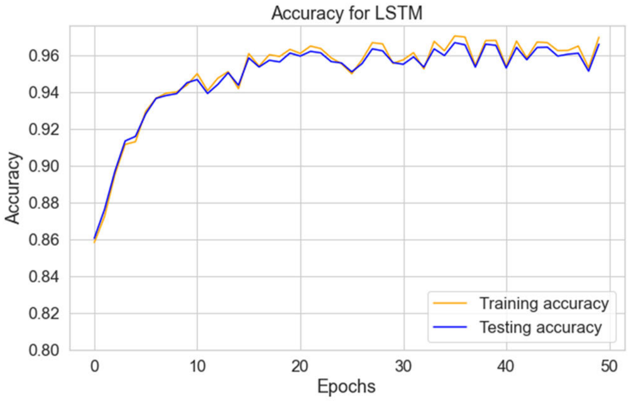


FIGURE 8. Training and testing accuracy curves for the LSTM model.

Table 1 illustrates the training and testing loss rates of the models. For the LSTM, the training and testing loss rates are 0.219 and 0.236, respectively, whereas for the CNN 0.181 and 0.232, respectively, and for the CNN-LSTM 0.154 and 0.167, respectively, showing that the CNN-LSTM achieves the lowest loss rates for training and testing.

Figure 8 shows the two curves for training and test accuracy of the LSTM model. The training curves are also called learning curves. The orange curve is training accuracy, which continuously changes and reaches a maximum value of 96.98% after 50 epochs. The blue curve represents the test accuracy, which starts from 86.06% and reaches 96.61% after 50 epochs.

Fig. 9 shows the loss rate curves of the LSTM model based on the training and test sets. The loss rate continuously decreases with increasing number of epochs. In the case of the training set, the loss rate reaches 0.219 after 50 epochs, while the rate loss for the test set starts at 0.612 and declines to 0.236.

Figure 10 shows the training and testing accuracy for the CNN over 50 epochs. The training, orange, curve starts at 79.93% and reaches 94.51%. Similarly, the test, blue, curve starts at 80.56% and increases to 95.79%. Figure 11 represents the training and testing loss for the CNN. The training loss starts from 0.585 and decreases to 0.181 after 50 epochs, while the test loss curve starts at 0.577 and decreases to 0.232.

Figure 12 illustrates the training and testing accuracy for the CNN-LSTM. The training, orange, curve starts at 86.37%, and reaches 98.23% after 50 epochs. The test, blue, curve starts from 86.70% and rises to 97.76%.

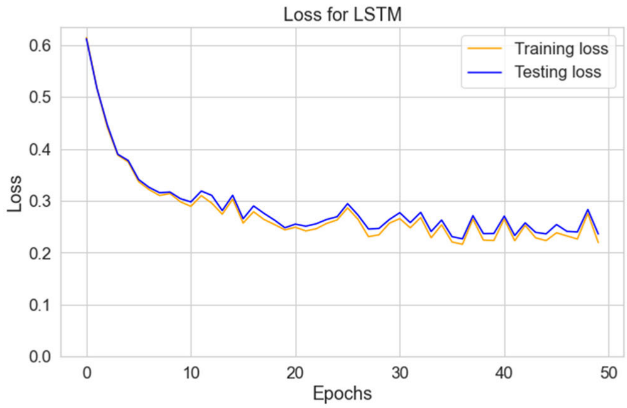


FIGURE 9. Training and testing loss curves for the LSTM model.

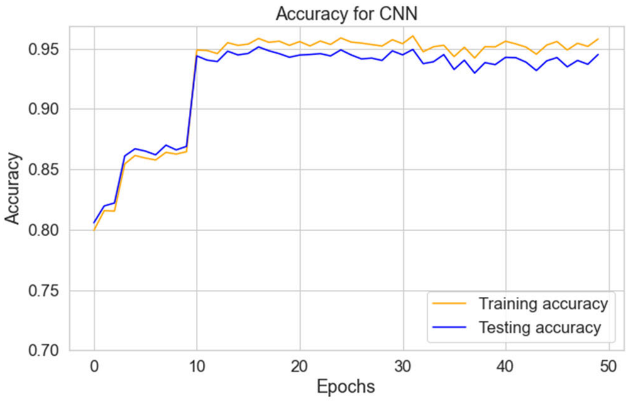


FIGURE 10. Training and testing accuracy curves for the CNN model.

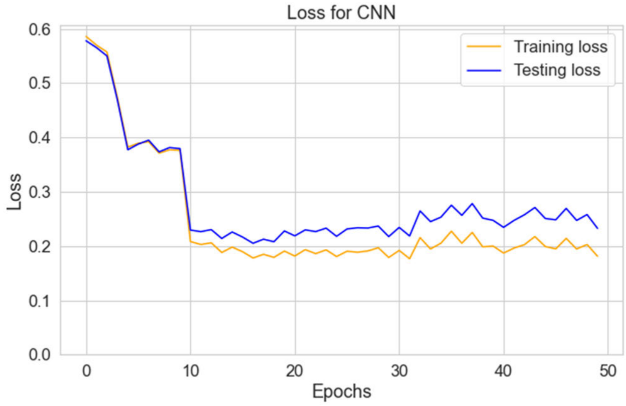


FIGURE 11. Training and testing loss curves for the CNN model.

Fig. 13 shows the training and test loss for the CNN-LSTM model. The training loss rate starts from 0.485 and reduces to 0.154, while the test loss rate starts from 0.476 and falls to 0.167.

Figure 14 depicts the confusion matrix obtained from the LSTM model. It predicts a class label, the ''predicted label'', which is compared to the true label in the test dataset. The diagonal values of the matrix indicate the accuracy of the classification, while the values above and below the diagonal illustrate errors that occurred. Here, the confusion matrix detects data of 884, 3465, 587, 480, 1013, and 4181 as true positives for the six activities: downstairs, jogging, sitting, standing, upstairs, and walking, respectively.

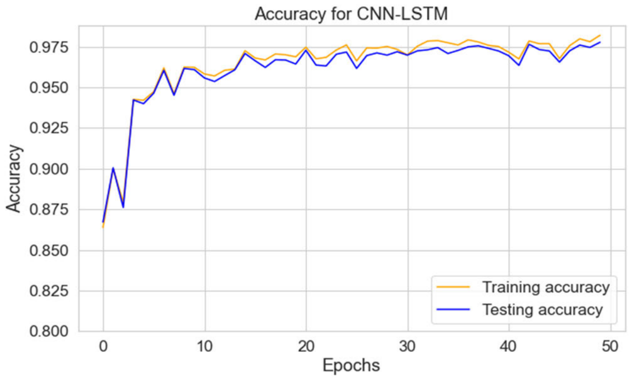


FIGURE 12. Training and testing accuracy curves for the CNN-LSTM model.

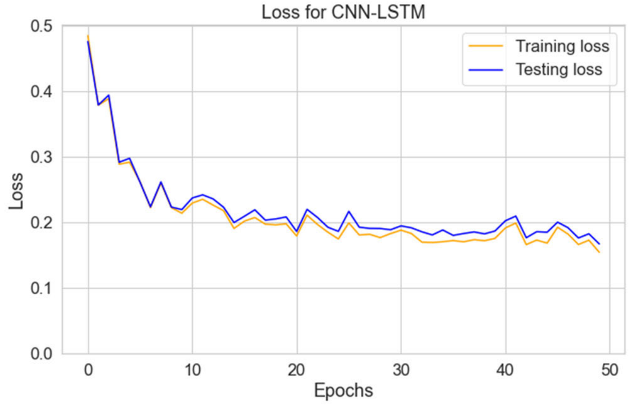


FIGURE 13. Training and testing loss curves for the CNN-LSTM model.

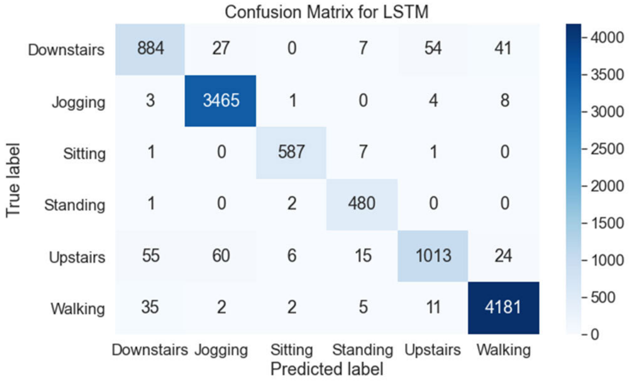


FIGURE 14. The confusion matrix for the LSTM model.

Figure 15 shows the test dataset based on the confusion matrix of the CNN model, showing 10379 instances in the test dataset which are correctly classified. Predicted activity matched the true class for downstairs, jogging, sitting, standing, upstairs, and walking in 817, 3422, 585, 456, 1034, 4065 cases, respectively.

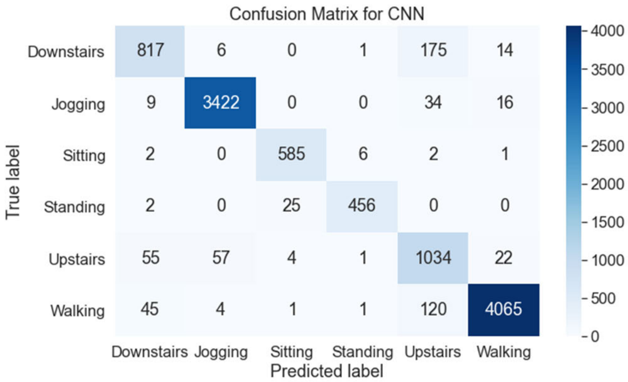


FIGURE 15. The confusion matrix for the CNN model.

TABLE 2. The evaluation metrics for the three models.

Metrics	LSTM	CNN	CNN-LSTM
Precision (%)	96.57	94.83	**97.75**
Recall (%)	96.61	94.51	**97.77**
$F1$-Measure (%)	96.57	94.61	**97.76**

The confusion matrix for the CNN-LSTM model is presented in Figure 16. It is clear from this figure that the CNN-LSTM model performs very well, with few error values occurring below and above the diagonal of the matrix. Table 2 summarizes the evaluation metrics in terms of the precision, recall, and $F1$-Measure of the three models, to evaluate their relative performance on the WISDM dataset. The precision, recall, and $F1$-Measure for the LSTM are 96.57%, 96.61%, and 96.57%, respectively. For the CNN model, 94.83%, 94.51%, and 94.61%, respectively. While the CNN-LSTM achieved 97.75%, 97.77%, and 97.76% for precision, recall, and $F1$-Measure, respectively. Figure 17 shows a comparison of these metrics of the proposed models.

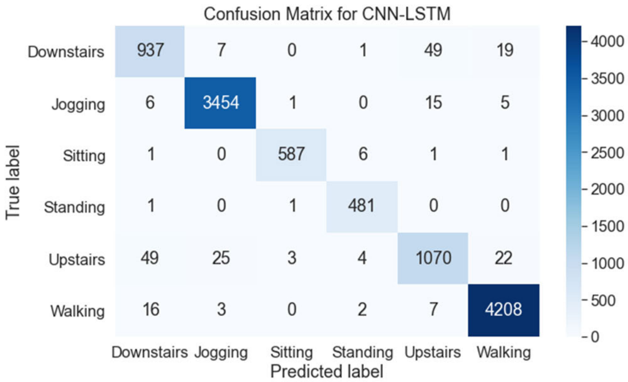


FIGURE 16. The confusion matrix for the CNN-LSTM model.

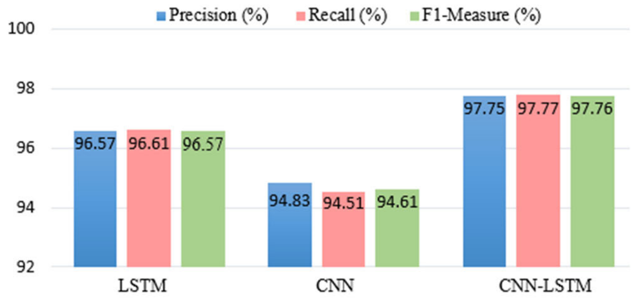


FIGURE 17. Comparison of the precision, recall, and F1-Measure for the three proposed models: LSTM, CNN, and CNN-LSTM.

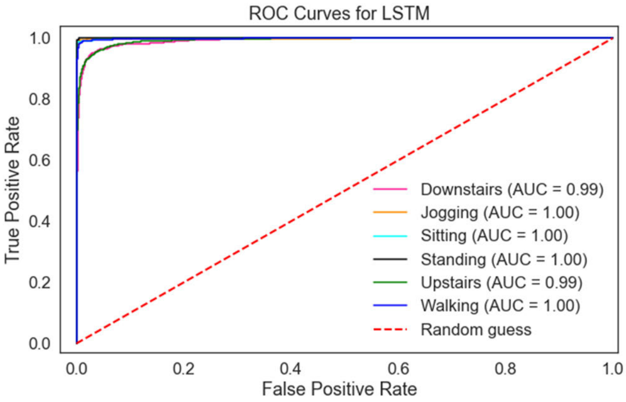


FIGURE 18. ROC curves for the LSTM model.

ROC curves are shown in Figures 18, 19, and 20. These curves are probability curves used to evaluate the models and show the true positive rate (TPR) on the ordinate axis and the false positive rate (FPR) on the abscissa for a range of threshold values varying from 0.0 to 1.0.

Figure 18 illustrates the ROC curves for the LSTM model for each activity. Downstairs and upstairs activities have AUC values of 0.99, and jogging, sitting, standing, and walking activities achieve 1.0 AUC.

Figure 19 demonstrates the ROC curves per class for the CNN model. In this figure, the highest value is the standing class, which achieves 1.0 AUC, and the lowest AUC value is the upstairs class, which achieved 0.86 AUC. The CNN model achieved 0.91 AUC for downstairs, and 0.99 for each of jogging, sitting, and walking.

Figure 20 illustrates the ROC curves for the CNN-LSTM, and we see the AUC values for all activities are 1.00. This result indicates that the CNN-LSTM model achieves better performance results than the LSTM and CNN models.

Figure 21 illustrates the precision-recall curves for the LSTM model for each activity. Downstairs "class 0", jogging "class 1", sitting "class 2", standing "class 3", upstairs "class 4", and walking "class 5" achieve area values of 0.938, 0.999, 0.997, 0.995, 0.956, and 0.998, respectively. Therefore, the area under the micro-average precision-recall curve is 0.994. This value is computed via the area summation of each class "from class 0 to class 5" divided by the number of the classes (six classes in this case).

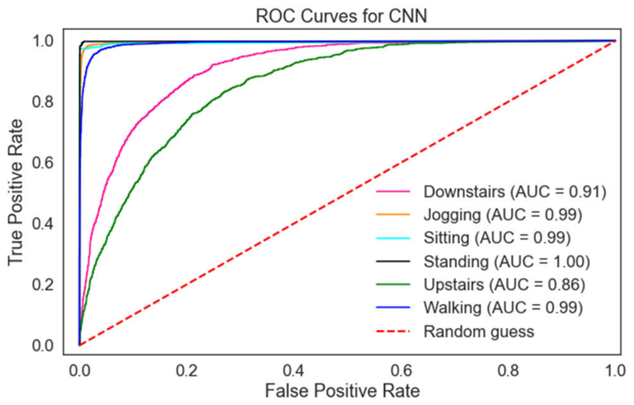


FIGURE 19. **ROC curves for the CNN model.**

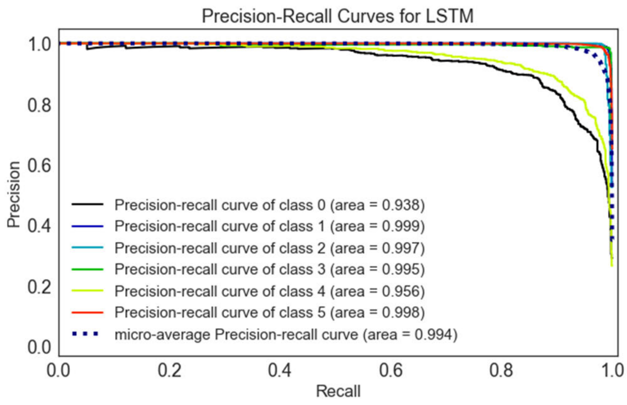


FIGURE 21. **Precision-recall curves for the LSTM model.**

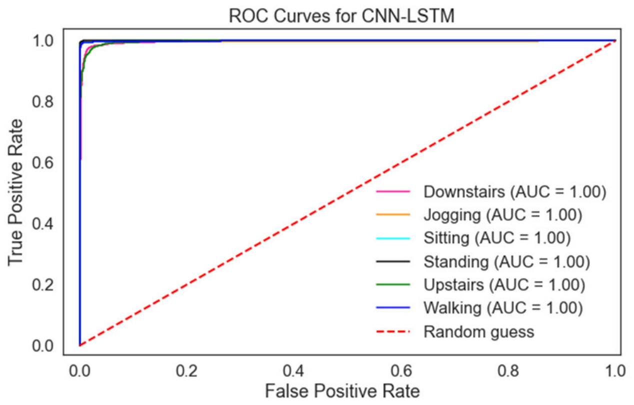


FIGURE 20. **ROC curves for the CNN-LSTM model.**

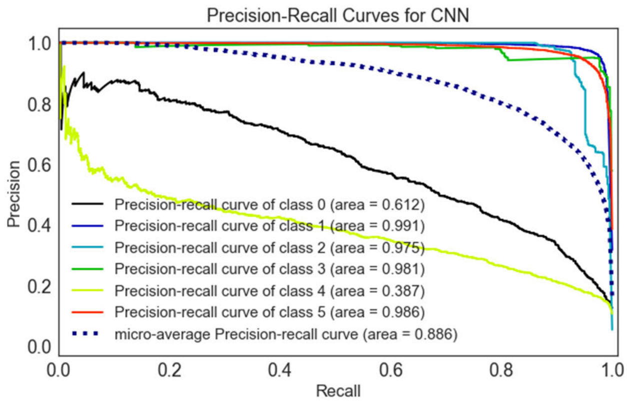


FIGURE 22. **Precision-recall curves for the CNN model.**

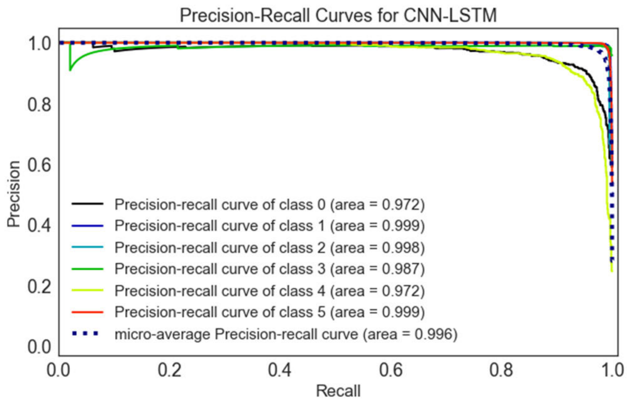


FIGURE 23. **Precision-recall curves for the CNN-LSTM model.**

Figure 22 demonstrates the precision-recall curves per class for the CNN model. In this figure, the highest value is the jogging activity "class 1", which achieved 0.991 area, and the lowest area value is the upstairs activity "class 4", which achieved 0.387 area. The CNN model achieved 0.612 area for downstairs activity "class 0" and 0.975 for sitting activity "class 2". Also, this model has 0.981 and 0.986 areas for standing activity "class 3" and walking activity "class 5", respectively, and the average area for the precision-recall curve is 0.886.

Figure 23 shows the precision-recall curves for the CNN-LSTM, in which the area values for class 0, class 1, class 2, class 3, class 4, and class 5 are 0.972, 0.999, 0.998, 0.987, 972, and 0.999, respectively. Therefore, the area under the micro-average precision-recall curve is 0.996. This value means that the CNN-LSTM model achieves better performance results than both the LSTM and CNN models.

Due to the fact that ROC and precision-recall curves are comprehensive metrics, which count all statistical values "*TP*, *TN*, *FP*, and *FN*", they are used in the evaluation of the proposed models. In addition, both curves enable easy and instant visual diagnosis of the models' behavior. Also, these curves depend on the area parameter, where a greater area means a more feasible test, and the areas under ROC curves are used to compare the usefulness of tests.

Table 3 shows the performance of the proposed models using the k-fold cross-validation technique. In this technique, the dataset is divided into 5 equal partitions (4 partitions are utilized for training and one partition for validation), where k denotes the partition number and equals 5 in this case. The proposed models are repeatedly trained 5 times using different partitioning of the dataset.

TABLE 3. The performance of the proposed models using 5-fold cross-validation.

Metrics	LSTM	CNN	CNN-LSTM
Average Accuracy	96.97% (±0.47%)	94.83% (±0.54%)	**97.98% (±0.39%)**
Average Precision	96.92% (±0.58%)	94.70% (±0.62%)	**97.81% (±0.50%)**
Average Recall	96.96% (±0.52%)	94.75% (±0.60%)	**97.89% (±0.41%)**
Average $F1$-Measure	96.91% (±0.42%)	94.89% (±0.51%)	**97.99% (±0.32%)**

According to this division, the average accuracy for the LSTM is 96.97% with ±0.47% standard deviation, while the average precision is 96.92% with ±0.58% standard deviation. Similarly, the average recall is 96.96% with ±0.52% standard deviation, and the average $F1$-Measure is 96.91% with ±0.42% standard deviation. For the CNN, the average accuracy, precision, recall, and $F1$-Measure are 94.83%, 94.70%, 94.75%, and 94.89%, respectively, while the standard deviations are ±0.54%, ±0.62%, ±0.60%, and ±0.51%, respectively. Finally, the CNN-LSTM achieved the highest average accuracy, precision, recall, $F1$-Measure, which are 97.98%, 97.81%, 97.89%, and 97.99%, respectively, with standard deviations of ±0.39%, ±0.50%, ±0.41%, and ±0.32%, respectively. Therefore, the proposed CNN-LSTM model has the best performance, with the minimum standard deviation. Moreover, the k-fold cross-validation technique improved the models' performance and avoided the biasing of the performance results via a convenient division of the training and testing dataset.

VI. DISCUSSION

In Figure 7, the proposed models are compared in terms of their accuracy. The results demonstrate that the proposed CNN-LSTM has the greatest accuracy and the least loss rate for training and testing. Results from the ROC curves and confusion matrices demonstrate that the accurate classification of human activities can be achieved with the proposed model. Figures 8, 10, and 12 present the performance of the proposed models in terms of the accuracy of training and testing using the WISDM dataset. Figures 14, 15, and 16 show distributions of error rates across the classes.

Figure 17 presents a comparison of the three models in terms of precision, recall, and $F1$-Measure. The micro-average precision-recall curve has an area of 99.6% for the CNN-LSTM model. Also, the AUC is 100% for the CNN-LSTM model for all six activities, a better performance than either the LSTM or CNN models. Further, from Table 3, the k-fold cross validation technique is utilized to verify the performance results for the proposed models in terms of the mean accuracy, precision, recall, and $F1$-Measure at $k = 5$.

Finally, Table 4 presents a comparison of the accuracy between this work and previous published works. We see that the accuracy achieved by the CNN-LSTM model, 97.76%, is better than the models referred in [24], [26], [28], [30], [31].

TABLE 4. Comparison of the accuracy between this work and previous works.

Reference	Model	Dataset Used	Accuracy (%)
[24]	LSTM-RNN	WISDM	94
[26]	Bi-directional LSTM	Opportunity	92.7
[28]	CNN	UCI-HAR	91.98
[30]	LSTM-CNN	WISDM	95.85
[31]	ConvLSTM	Skoda	95.8
This Work	LSTM	WISDM	**96.61**
	CNN	WISDM	**94.51**
	CNN-LSTM	WISDM	**97.76**

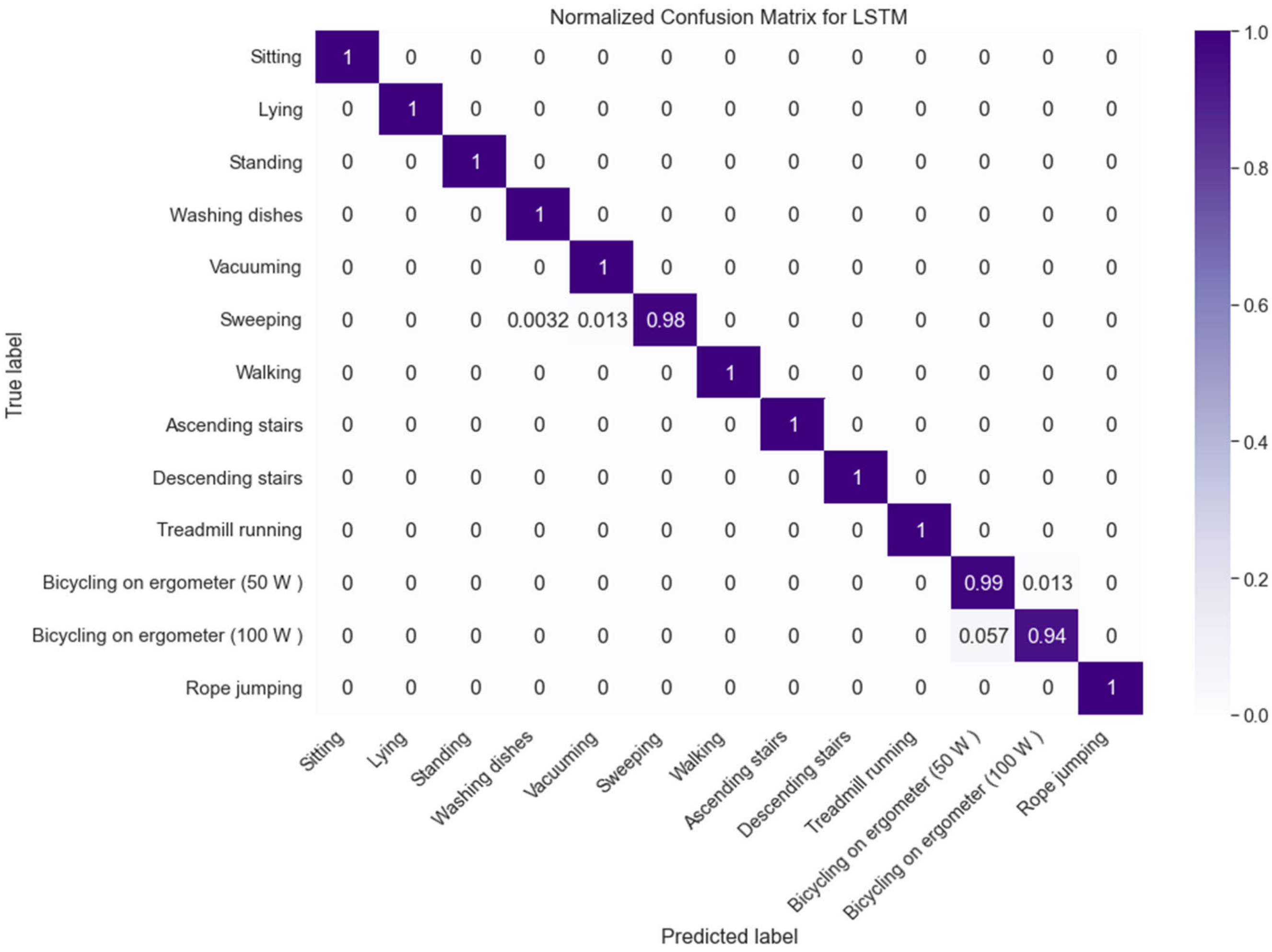
The better performance of the proposed models can be attributed to the fine-tuning of the hyper-parameters of the models, which includes the number of training epochs, loss and activation functions, learning rate, batch size, dropout rate, optimizer type, and number of neurons for the utilized layers in the proposed models.

In particular, for the CNN, when the batch size, learning rate, and training epochs number were set to 32, 0.0025, and 10, respectively, and using a softmax activation function, the CNN model achieved a testing accuracy of 93.87%, but in case the batch size was changed to **12**, the learning rate was adjusted to 0.00, the training epochs were reconfigured to 5, and the same activation function was used, the CNN model' testing accuracy reached **92.3**%.

Also, for the LSTM and the CNN-LSTM, when the batch size, learning rate, and training epochs number were set to be 128, 0.0050, and 10, respectively, and using a sigmoid activation function, the LSTM and CNN-LSTM models achieved a testing accuracy of 87.47% and 86.3%, respectively, but when the batch size is changed to be 256, the learning rate is tuned to be 0.001, the training epochs are reconfigured to be 5, and the same activation function, the two models testing accuracy reached 88.44% and 89%, respectively.

So, the optimization and proper settings of these parameters significantly enhances the results. Moreover, the best possible performance of the three models was achieved when the governing parameters were set to a batch size of 64 with training epochs of 50 and the learning rate of 0.0025, the Adam optimization algorithm is utilized, and a regularization technique, based on a cross-entropy loss function, is implemented.

The hyper-parameters of the proposed models are tuned using GridSearchCV method, which automatically computes the optimum values of the hyper-parameters to achieve better performance of the proposed models.



	Sitting	Lying	Standing	Washing dishes	Vacuuming	Sweeping	Walking	Ascending stairs	Descending stairs	Treadmill running	Bicycling on ergometer (50 W)	Bicycling on ergometer (100 W)	Rope jumping
Sitting	1	0	0	0	0	0	0	0	0	0	0	0	0
Lying	0	1	0	0	0	0	0	0	0	0	0	0	0
Standing	0	0	1	0	0	0	0	0	0	0	0	0	0
Washing dishes	0	0	0	1	0	0	0	0	0	0	0	0	0
Vacuuming	0	0	0	0	1	0	0	0	0	0	0	0	0
Sweeping	0	0	0	0.0032	0.013	0.98	0	0	0	0	0	0	0
Walking	0	0	0	0	0	0	1	0	0	0	0	0	0
Ascending stairs	0	0	0	0	0	0	0	1	0	0	0	0	0
Descending stairs	0	0	0	0	0	0	0	0	1	0	0	0	0
Treadmill running	0	0	0	0	0	0	0	0	0	1	0	0	0
Bicycling on ergometer (50 W)	0	0	0	0	0	0	0	0	0	0	0.99	0.013	0
Bicycling on ergometer (100 W)	0	0	0	0	0	0	0	0	0	0	0.057	0.94	0
Rope jumping	0	0	0	0	0	0	0	0	0	0	0	0	1

FIGURE 24. The normalized confusion matrix for the LSTM model using the DaLiAc dataset.

TABLE 5. Classification report of the LSTM using daliac dataset.

Activity	Precision	Recall	$F1$-Measure
Sitting	1.0000	1.0000	1.0000
Lying	1.0000	1.0000	1.0000
Standing	1.0000	1.0000	1.0000
Washing dishes	0.9974	1.0000	0.9987
Vacuuming	0.9796	1.0000	0.9897
Sweeping	1.0000	0.9840	0.9919
Waking	1.0000	1.0000	1.0000
Ascending stairs	1.0000	1.0000	1.0000
Descending stairs	1.0000	1.0000	1.0000
Treadmill running	1.0000	1.0000	1.0000
Bicycle ergometer (50 W)	0.9453	0.9870	0.9657
Bicycle ergometer (100 W)	0.9864	0.9430	0.9642
Rope jumping	1.0000	1.0000	1.0000
Accuracy	-	-	0.9917
Macro average	0.9930	0.9934	0.9931
Weighted average	0.9919	0.9917	0.9917

Additionally, the proposed LSTM model is applied to the Daily Life Activities (DaLiAc) dataset [67] to widely evaluate the performance of the model with other previously reported models in the literature. Figure 24 shows the normalized confusion matrix for the LSTM model using the DaLiAc dataset. Table 5 presents a classification report of the LSTM in terms of precision, recall, and $F1$-Measure. For the DaLiAc dataset, the achieved testing accuracy of the LSTM is 99.17%, while the accuracy was 98.9% in the case where the DaLiAc dataset was trained using the CNN model [68]. So, one can argue that the LSTM model proposed in this paper has the highest accuracy compared to the state-of-the-art methods reported in [68]–[70].

VII. CONCLUSION

This paper has presented an implementation of three deep learning models for the recognition of daily human activities. Experiments conducted to test the accuracy of the proposed models achieved the testing accuracy for LSTM, CNN, and CNN-LSTM of 96.61%, 94.51%, and 97.76%, respectively, the test loss rate of 0.236, 0.232, and 0.167, respectively, and the training loss rate of 0.219, 0.181, and 0.154, respectively. The optimum model is the combination of

CNN-LSTM with an accuracy of 98.23% for training and 97.76% for testing.

The confusion matrix, precision, recall, $F1$-Measure, ROC curves, and AUC are presented to allow evaluation of the model's performance. It is clear that the CNN model scored least for precision (94.83%), LSTM has median precision (96.57%), and CNN-LSTM has the greatest precision (97.75%).

The achieved recalls are 96.61%, 94.51%, and 97.77% for LSTM, CNN, and CNN-LSTM, respectively with the corresponding $F1$-Measures of 96.57%, 94.61%, and 97.76%. The average area under the precision-recall curve is 99.6% for the CNN-LSTM model. Finally, the AUC is 100% with all classes for the CNN-LSTM model.

When the LSTM was applied to the DaLiAc dataset, a high testing accuracy of 99.17% was obtained, which is higher than what is reported in the literature for CNN.

The hyper-parameters of the three models i.e., the loss and activation functions, the optimizer type, learning rate, number of training epochs, dropout rate, batch size, and number of neurons for the utilized layers in the proposed models, are found to significantly affect the accuracy of the proposed models, and high performance achieved is highly correlated to the optimal settings applied.

Additionally, the performance of the three models is evaluated in terms of accuracy, precision, recall, and $F1$-Measure using the k-fold cross-validation technique.

Therefore, referring to Industry 4.0, these models can accurately recognize human activities in a smart factory environment. In the future, the proposed models can be applied with more datasets and the training performance can be compared with a graphics processing unit (GPU) environment. One could also build new models of deep learning, such as gated recurrent unit (GRU).

The used dataset and developed code are available online at: https://www.kaggle.com/drsaeedmohsen/wisdmdataset2021

REFERENCES

[1] M. A. Moktadir, S. M. Ali, S. Kusi-Sarpong, and M. A. Shaikh, "Assessing challenges for implementing industry 4.0: Implications for process safety and environmental protection," *Process Saf. Environ. Protection*, vol. 117, pp. 730–741, Jul. 2018.

[2] A. Elkaseer, M. Salem, H. Ali, and S. G. Scholz, "Approaches to practical implementation of industry 4.0," in *Proc. 8th Int. Conf. Adv. Comput.-Hum. Interact. (ACHI)*, 2018, pp. 141–146.

[3] C. Zhang, Y. Chen, H. Chen, and D. Chong, "Industry 4.0 and its implementation: A review," *Inf. Syst. Frontiers*, pp. 1–12, Jun. 2021.

[4] S. Mohsen, A. Elkaseer, and S. G. Scholz, "Human activity recognition using k-nearest neighbor machine learning algorithm," in *Proc. 8th Int. Conf. Sustain. Design Manuf. (KES-SDM)*, Split, Croatia, 2021, pp. 304–313.

[5] I. A. Lawal and S. Bano, "Deep human activity recognition with localisation of wearable sensors," *IEEE Access*, vol. 8, pp. 155060–155070, 2020.

[6] S. Tanberk, Z. H. Kilimci, D. B. Tükel, M. Uysal, and S. Akyokus, "A hybrid deep model using deep learning and dense optical flow approaches for human activity recognition," *IEEE Access*, vol. 8, pp. 19799–19809, 2020.

[7] Y.-L. Hsu, H.-C. Chang, and Y.-J. Chiu, "Wearable sport activity classification based on deep convolutional neural network," *IEEE Access*, vol. 7, pp. 170199–170212, 2019.

[8] F. Demrozi, R. Bacchin, S. Tamburin, M. Cristani, and G. Pravadelli, "Toward a wearable system for predicting freezing of gait in people affected by Parkinson's disease," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 9, pp. 2444–2451, Sep. 2020.

[9] A. Malaisé, P. Maurice, F. Colas, F. Charpillet, and S. Ivaldi, "Activity recognition with multiple wearable sensors for industrial applications," in *Proc. 1st Int. Conf. Adv. Comput. Hum. Interact.*, 2018, pp. 1–7.

[10] Y. Wang, S. Cang, and H. Yu, "A survey on wearable sensor modality centred human activity recognition in health care," *Expert Syst. Appl.*, vol. 137, pp. 167–190, Dec. 2019.

[11] R. S. Antunes, L. A. Seewald, V. F. Rodrigues, C. A. D. Costa, L. Gonzaga, Jr., R. R. Righi, A. Maier, B. Eskofier, M. Ollenschläger, F. Naderi, R. Fahrig, S. Bauer, S. Klein, and G. Campanatti, "A survey of sensors in healthcare workflow monitoring," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 1–37, Jun. 2018.

[12] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition," presented at the 13th ACM Conf. Embedded Netw. Sensor Syst., Seoul, South Korea, 2015, doi: 10.1145/2809695.2809718.

[13] C. A. Ronao and S. B. Cho, "deep convolutional neural networks for human activity recognition with smartphone sensors," in *Neural Information Processing* (Lecture Notes in Computer Science), vol. 9492, S. Arik, T. Huang, W. Lai, and Q. Liu, Eds. Cham, Switzerland: Springer, 2015, doi: 10.1007/978-3-319-26561-2_6.

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

[15] M. Shoaib, S. Bosch, O. Incel, H. Scholten, and P. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, Jan. 2015.

[16] E. Lattanzi and V. Freschi, "Evaluation of human standing balance using wearable inertial sensors: A machine learning approach," *Eng. Appl. Artif. Intell.*, vol. 94, Sep. 2020, Art. no. 103812.

[17] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, Mar. 2019.

[18] 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.

[19] J.-H. Hong, J. Ramos, and A. K. Dey, "Toward personalized activity recognition systems with a semipopulation approach," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 1, pp. 101–112, Feb. 2016.

[20] R. Igual, C. Medrano, and I. Plaza, "A comparison of public datasets for acceleration-based fall detection," *Med. Eng. Phys.*, vol. 37, no. 9, pp. 870–878, 2015.

[21] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. London, U.K.: Springer, 2014.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[23] P. Klosowski, "Deep learning for natural language processing and language modelling," in *Proc. Signal Process., Algorithms, Architectures, Arrangements, Appl. (SPA)*, Sep. 2018, pp. 223–228.

[24] S. W. Pienaar and R. Malekian, "Human activity recognition using LSTM-RNN deep neural network architecture," in *Proc. IEEE 2nd Wireless Afr. Conf. (WAC)*, Aug. 2019, pp. 1–5.

[25] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *J. SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, 2011.

[26] 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.*, New York, NY, USA, 2016, pp. 1533–1540.

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

[28] 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.

[29] 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. Esann*, vol. 3, 2020, p. 3.

[30] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56855–56866, 2020.

[31] 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.

[32] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tröster, "Wearable activity tracking in car manufacturing," *IEEE Pervasive Comput.*, vol. 7, no. 2, pp. 42–50, Apr./Jun. 2008.

[33] A. A. Alani, G. Cosma, and A. Taherkhani, "Classifying imbalanced multi-modal sensor data for human activity recognition in a smart home using deep learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Glasgow, U.K., Jul. 2020, pp. 1–8.

[34] N. Twomey, T. Diethe, M. Kull, H. Song, M. Camplani, S. Hannuna, X. Fafoutis, N. Zhu, P. Woznowski, P. Flach, and I. Craddock. (2016). *The SPHERE Challenge: Activity Recognition With Multimodal Sensor Data*. [Online]. Available: http://arxiv.org/abs/1603.00797v3

[35] M. Alzantot, S. Chakraborty, and M. Srivastava, "SenseGen: A deep learning architecture for synthetic sensor data generation," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Kona, HI, USA, Mar. 2017, pp. 188–193.

[36] M. A. Alsheikh, D. Niyato, S. Lin, H.-P. Tan, and Z. Han, "Mobile big data analytics using deep learning and apache spark," *IEEE Netw.*, vol. 30, no. 3, pp. 22–29, Jun. 2016.

[37] S. R. Shakya, C. Zhang, and Z. Zhou, "Comparative study of machine learning and deep learning architecture for human activity recognition using accelerometer data," *Int. J. Mach. Learn. Comput*, vol. 8, no. 6, pp. 577–582, 2018.

[38] Y. Chen, K. Zhong, J. Zhang, Q. Sun, and X. Zhao, "LSTM networks for mobile human activity recognition," in *Proc. Int. Conf. Artif. Intell., Technol. Appl.*, 2016, pp. 50–53.

[39] S. Mekruksavanich, A. Jitpattanakul, P. Youplao, and P. Yupapin, "Enhanced hand-oriented activity recognition based on smartwatch sensor data using LSTMs," *Symmetry*, vol. 12, no. 9, p. 1570, Sep. 2020.

[40] P. Agarwal and M. Alam, "A lightweight deep learning model for human activity recognition on edge devices," *Proc. Comput. Sci.*, vol. 167, pp. 2364–2373, Jan. 2020.

[41] E. Cipolla, I. Infantino, U. Maniscalco, G. Pilato, and F. Vella, "Indoor actions classification through long short term memory neural networks," in *Proc. Int. Conf. Image Anal. Process.*, 2017, pp. 435–444.

[42] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional bi-directional LSTM networks," *Sensors*, vol. 17, no. 2, p. 273, 2017.

[43] W. Tao, M. C. Leu, and Z. Yin, "Multi-modal recognition of worker activity for human-centered intelligent manufacturing," *Eng. Appl. Artif. Intell.*, vol. 95, Oct. 2020, Art. no. 103868.

[44] C. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proc. 25th Int. Conf. Comput. Linguistics (COLING)*, Dublin, Ireland, 2014, pp. 69–78.

[45] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," 2015, *arXiv:1510.03820*.

[46] W. Xu, Y. Pang, Y. Yang, and Y. Liu, "Human activity recognition based on convolutional neural network," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Beijing, China, Aug. 2018, pp. 165–170.

[47] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Appl. Soft Comput.*, vol. 62, pp. 915–922, Jan. 2018.

[48] J. Huang, S. Lin, N. Wang, G. Dai, Y. Xie, and J. Zhou, "TSE-CNN: A two-stage end-to-end CNN for human activity recognition," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 1, pp. 292–299, Jan. 2020.

[49] (2012). *In Wireless Sensor Data Mining Dataset*. [Online]. Available: https://www.cis.fordham.edu/wisdm/dataset.php

[50] D. Karmiani, R. Kazi, A. Nambisan, A. Shah, and V. Kamble, "Comparison of predictive algorithms: Backpropagation, SVM, LSTM and Kalman filter for stock market," in *Proc. Amity Int. Conf. Artif. Intell. (AICAI)*, Feb. 2019, pp. 228–234.

[51] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[52] M. Shakeel, K. Itoyama, K. Nishida, and K. Nakadai, "Detecting earthquakes: A novel deep learning-based approach for effective disaster response," *Appl. Intell.*, vol. 51, pp. 8305–8315, Apr. 2021.

[53] A. Chu, Y. Lai, and J. Liu, "Industrial control intrusion detection approach based on multiclassification GoogLeNet-LSTM model," *Secur. Commun. Netw.*, vol. 2019, Dec. 2019, Art. no. 6757685.

[54] A. Khorram, M. Khalooei, and M. Rezghi, "End-to-end CNN + LSTM deep learning approach for bearing fault diagnosis," *Int. J. Speech Technol.*, vol. 51, no. 2, pp. 736–751, Feb. 2021.

[55] J. A. Gamble and J. Huang, "Convolutional neural network for human activity recognition and identification," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Montreal, QC, Canada, Aug. 2020, pp. 1–7.

[56] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[57] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proc. 23rd Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn. (ESANN)*, 2015, pp. 89–94.

[58] M. Gil-Martín, R. San-Segundo, F. Fernández-Martínez, and J. Ferreiros-López, "Improving physical activity recognition using a new deep learning architecture and post-processing techniques," *Eng. Appl. Artif. Intell.*, vol. 92, Jun. 2020, Art. no. 103679.

[59] M. Ibrahim, M. Torki, and M. ElNainay, "CNN based indoor localization using RSS time-series," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 1044–1049.

[60] A. Yenter and A. Verma, "Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis," in *Proc. IEEE 8th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2017, pp. 540–546.

[61] R. Chandra, S. Goyal, and R. Gupta, "Evaluation of deep learning models for multi-step ahead time series prediction," *IEEE Access*, vol. 9, pp. 83105–83123, 2021.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015, *arXiv:1412.6980*.

[63] J. Xiong, K. Zhang, and H. Zhang, "A vibrating mechanism to prevent neural networks from overfitting," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 1737–1742.

[64] J. Pengfei, Z. Chunkai, and H. Zhenyu, "A new sampling approach for classification of imbalanced data sets with high density," in *Proc. Int. Conf. Big Data Smart Comput. (BIGCOMP)*, Jan. 2014, pp. 217–222.

[65] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[66] M. Haggag, M. M. Tantawy, and M. M. S. El-Soudani, "Implementing a deep learning model for intrusion detection on apache spark platform," *IEEE Access*, vol. 8, pp. 163660–163672, 2020.

[67] H. Leutheuser, D. Schuldhaus, and B. M. Eskofier, "Hierarchical, multi-sensor based classification of daily life activities: Comparison with state-of-the-art algorithms using a benchmark dataset," *PLoS ONE*, vol. 8, no. 10, p. 75196, 2013.

[68] T. Hur, J. Bang, T. Huynh-The, J. Lee, J.-I. Kim, and S. Lee, "Iss2Image: A novel signal-encoding technique for CNN-based human activity recognition," *Sensors*, vol. 18, no. 11, p. 3910, Nov. 2018.

[69] T. Huynh-The, C.-H. Hua, N. A. Tu, and D.-S. Kim, "Physical activity recognition with statistical-deep fusion model using multiple sensory data for smart health," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1533–1543, Feb. 2021.

[70] T. Huynh-The, C.-H. Hua, and D.-S. Kim, "Visualizing inertial data for wearable sensor based daily life activity recognition using convolutional neural network," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2019, pp. 2478–2481.

SAEED MOHSEN received the B.Sc. degree (Hons.) in electronics engineering and electrical communications from Thebes Higher Institute, Cairo, Egypt, in 2013, and the M.Sc. and Ph.D. degrees in electrical engineering from Ain Shams University, Cairo, Egypt, in 2016 and 2020, respectively. He is currently an Assistant Professor with the Al-Madina Higher Institute for Engineering and Technology, Giza, Egypt. He made intensive research on applications of artificial intelligence (AI), such as deep learning and machine learning. He published a number of papers in specialized international conferences and peer-reviewed periodicals. His research interests include analog electronics, smart grids, biomedical electronics, energy harvesting, and the Internet of Things (IoT).

AHMED ELKASEER is currently a Senior Research Associate at the Institute for Automation and Applied Informatics (IAI), Karlsruhe Institute of Technology (KIT), Germany. He received the Ph.D. degree from the Cardiff School of Engineering, Cardiff University, U.K., in 2011. He has more than 15 years' experience of research in advanced manufacturing technologies. He has been working on different EC and EPSRC funded research projects. His work entails performing experimental and laboratory work, modeling, simulation and optimization-based studies of mechanical, and EDM and laser processing of advanced materials on conventional and micro scales, with a recent emphasis on additive and smart manufacturing and Industry 4.0 applications. His studies have led to several publications in the area of conventional and advanced micro and nano-manufacturing technologies. He is being invited to serves as an editorial board member and a reviewer for a number of journals, and to act as a scientific committee chair and a program committee member for a number of international conferences.

STEFFEN G. SCHOLZ is currently the Head of the Research Team 'Process optimization, Information management and Applications,' part of the Institute for Automation and Applied Informatics (IAI), Karlsruhe Institute of Technology. He is also an Honorary Professor at Swansea University, U.K., and an Adjunct Professor at the Vellore Institute of Technology, India. He is also the Principal Investigator in the Helmholtz funded long-term programs 'Digital System Integration' and 'Printed Materials and Systems.' He has more than 20 years of experience in the field of system integration and automation, sustainable flexible production, polymer micro- and nano-replication, process optimization and control, with a recent emphasis on additive manufacturing and Industry 4.0 applications. In addition to pursuing and leading research, he has been very active with knowledge transfer to industry. He has been involved in over 30 national and international projects, and he won in excess of 20M EUR research grants, in which he has acted as a co-ordinator and/or a principal investigator. His academic output includes more than 150 technical articles and five books. He is a chair of different international conferences within the scope of advanced and sustainable manufacturing technologies.