Smartphone-based Human Activity Recognition using CNN and Bidirectional LSTM Model

First Authora,[[1]](#footnote-1)\*, Second Authorb and Third Authorb

aJournal Production Department, IOS Press, Nieuwe Hemweg 6b, 1013 BG, Amsterdam, The Netherlands

bDeparment first, then University or Company name, Insert a complete correspondence (mailing) address, Abbreviate US states, Include country

**Abstract.** Global Health Observatory (GHO) data reveals approximately 23% of adults are not sufficiently active. Physical inactivity is one of the ten top risk factors for mortality. Regular recognition and self-monitoring of physical activity potentially encourage users to stay active. Ambient assisted living systems are solutions for the objective. Intelligent human activity recognition (HAR) system is one of the crucial components of such innovations. Hence, a smartphone-based intelligent human physical activity prediction system based on temporal dynamics deep features is proposed in this paper. A stacking spatial-temporal deep model is developed to extract low level to higher-level features of inertial data. In the proposed system, a convolutional architecture is pipelined with bidirectional long-short-term-memory to encapsulate spatial and temporal state dependencies of motion data. In this work, we adopt a support vector machine as a classifier to perform activity classification. Empirical results demonstrate that the proposed HAR system is able to exhibit promising performances on UCI and WISDM databases with 92% and 87% accuracy respectively.

Keywords: smartphone, inertial data, human activity recognition, convolutional neural network, bidirectional long-short-term-memory, machine learning

# Introduction

On the World Health Organization (WHO) website, Global Health Observatory (GHO) data reveals approximately 23% of adults are not sufficiently active [1]. The level of insufficient physical activity is even worse in those high-income countries, in which about 60% of adults are insufficiently physically active. Physical inactivity is one of the ten top risk factors for mortality. WHO applauded that there is a 20% to 30% increased risk of all-cause mortality to people who are insufficiently physically active, compared to those who have at least two and half hours moderate-intensity physical activity per week.

Overwhelming testimony substantiates the belief that the insufficiency of physical activity contributes to a host of chronic diseases such as ischemic heart disease, high blood pressure, diabetes, stroke, hypertension, depression, and cancers [2], [3]. The upsurge of chronic diseases potentially impacts social and economic costs, for example, unemployment in the labour market, financial burden, etc. [4]. Hence, communities must take action to increase individuals’ physical activity.

Regular recognition and self-monitoring of physical activity can potentially encourage habits of adopting a healthy lifestyle such as regular exercise, as they have a more positive outcome expectancy of their body figure [5], [6]. ICT-enabled assisted living or “ambient assisted living” (AAL) systems are solutions for the objective. Intelligent human activity recognition (HAR) system is one of the crucial components of such innovations. There are two kinds of HAR systems: vision-based, wearable sensor-based, and smartphone-based [4], [7]–[10].

# Motivation

Wearable sensor-based approach and vision-based approach are two of the most common approaches in HAR. Both attain promising recognition performance, reaching above 80% accuracy in recognizing human physical activity [7], [11], [12]. However, these approaches are inconvenient and not easy to implement. For instance, there is a privacy issue revolving in the vision-based approach. Placing a surveillance camera in public places may violate the law and require extensive justification to obtain permits. On the other hand, in a wearable sensor-based approach, some people are reluctant to wear the sensor device(s) [13].

|  |
| --- |
|  |
| Fig. 1 Overview of the proposed model |

Henceforth, physical activity prediction using a smartphone is a contemporary research area in the HAR domain. Smartphone is a sensor-based ubiquitous piece of technology that is far more than just a communication device. With great technology development, smartphones are packed with high-end hardware and features. Several sensors are embedded inside smartphones, including motion sensors. The potential of the smartphone-based HAR approach is uplifted due to the mobility and simplicity of smartphone usage, providing people something very accessible and easy to use. Hence, the smartphone-based human physical activity prediction system based on temporal dynamics deep features is proposed in this paper.

An amalgamation of a one-dimensional (1D) convolutional neural network and recurrent neural network (RNN) variant, i.e. bi-directional Long Short Term Memory, is proposed to predict human activity based on the inertial data captured from a smartphone, illustrated in Fig. 1.

# Literature Review on Human Activity Recognition

As aforementioned, HAR can be categorized into three spheres: (1) vision-based HAR, (2) wearable sensor-based HAR and (3) smartphone-based HAR. Vision-based human activity recognition (coined as VHAR) is a process of categorizing a sequence of image recording with action/ activity class labels [7]. VHAR systems are extensively employed in various applications, especially for public area surveillance, healthcare monitoring as well as human-computer interaction. The proposed VHAR approaches include but not limited to Discrete Fourier transform-based HAR to extract the global representation of activity data [14], stacked Fisher vectors to capture more statistical information from frame images [15], extraction of multi-features from body silhouettes and joints information [16], etc.

However, the privacy issue in VHAR is a major concern from the public. Hence, the wearable sensor-based HAR (coined as WHAR) is proposed as an alternative. The main applications of WHAR are in the areas of healthcare, sports training, smart environment, etc. Wearable sensors include accelerometer, gyroscope, and magnetometer. The works include performing HAR using artificial neural network and smartwatch [17], Convolutional Neural Network for k-nearest neighborhood-based wearable sensor HAR [18], adopting J48 classifier in HAR for wearable sensors [19], etc.

Inconveniences of wearing, technological barriers such as limitation of current battery technology and culture barriers such as the association of a stigma with the use of medical sensing devices for monitoring limit the potential of WHAR usage. The smartphone-based approach is a seemly alternative for collecting motion inertial data signals. Most smartphones are equipped with a built-in gyroscope and accelerometer. In recent years, there are extensive research works working on adopting smartphones for HAR done by other researchers [4], [8], [20]–[24]. Kwapisz et al. utilize triaxial acceleration data captured by an Android smartphone to perform human activities [10]. The raw triaxial acceleration data is divided into 10-second segments. Then, forty-three statistical features are computed for each segment. The authors evaluate their self-collected dataset, namely Wireless Sensor Data Mining (WISDM) dataset, with various kinds of classifiers. Empirical results show promising performance.

In [20], a new database, namely UC Irvine (UCI) HAR dataset is collected with six different activity classes from a group of 30 volunteers carrying the smartphone on their waist. The collected data is triaxial acceleration and angular velocity data. The collected inertial signals are sampled in fixed-width sliding windows of 2.56 seconds with a 50% overlap between them. Next, 561 time and frequency domain features are extracted to describe each activity window. Support Vector Machine (SVM) is used to classify the activities.

|  |
| --- |
|  |
| Fig. 2 The block diagram of the proposed model |

Human activities are hierarchical. Hence, deep learning comprising multiple layers of neural networks is explored to reveal features from low to higher levels hierarchically. Deep learning has become a critical research in HAR [25]–[29]. Convolutional Neural Network (CNN) or commonly referred to as Covnet (LeNet) is a popular deep learning model [30], [31]. CNN is proposed to capture the local dependencies and spatial domain of activity signals [32]. The authors utilize multichannel time series data to recognize users’ activity and hand gestures. Since the inertial signal is a one-dimensional (1D) data, the traditional CNN has been improved to perform 1D convolution operation on the accelerometer and gyroscope triaxial sensor data [8], [25].

Long Short-Term Memory (LSTM), a variant of Recurrent Neural Network (RNN), has been proposed on triaxial accelerometers data for smartphone-based human activity prediction [33]. LSTM adopts past information to predict the outcome of a HAR model. It allows the network to learn when to “forget” previous hidden states and when to update hidden states given new information. However, some information may not be captured since human motion is continuous [34].

Since LSTM only takes in past information, Bidirectional Long Short-Term Memory, coined as BLSTM, is proposed to tackle both past information and future information. In other words, BLSTM is stacked into layers both horizontally and vertically. In the model, a single LSTM node can take in information from the horizontal layer for both past and future information, as well as from a vertical layer which is the lower hidden layer. [34] and [35] utilizes BLSTM on HAR using the inertial sensor in the smartphone. Experimental results demonstrate that the models outperform other existing approaches.

# Contributions of the work

Upholding the hypotheses of (1) the spatial and temporal information embedded in the inertial signal is crucial to represent activity, and (2) human activities are hierarchical, we propose a temporal dynamics deep learner that extract features from low to higher levels hierarchically from spatial and temporal domains. The main contributions of this work are summarized into threefold:

1. A stacking spatial-temporal deep model is developed to extract low level to higher-level features of inertial data for human activity recognition. Piling a convolutional architecture to deep BLSTM models enables both spatial and temporal state dependencies encapsulation to predict human activity.
2. An extensive experimental analysis using various performance measures, such as true positive rate, false positive rate, precision, recall, the area under the curve, confusion matrix, etc., is conducted on two publicly available datasets, namely WISDM and UCI datasets.
3. Various machine learning algorithms are explored to evaluate the effectiveness of the deep features extracted by the proposed dynamics deep model. These machine learning algorithms include logistic regression, support vector machine, Naïve Bayes, random forest, multilayer perceptron, k-nearest neighbours, etc. On top of that, performance comparison with other existing approaches is addressed as well.

# Proposed Solution

This work proposes a temporal deep learner stacking a hierarchical convolutional architecture with a model that comprehends the dynamics pattern of the inertial sequential data to predict human activity. Fig. 2 illustrates the block diagram of the proposed solution. Inertial data acquired from the smartphone is transformed into deep features comprising underlying rich features via the CNN-RNN-based feature extraction structure. Then, these deep features are further analyzed to predict human activity by using a machine learning classifier.

Specifically, in the proposed architecture, the feature extraction structure comprises three convolutional layers each with RELU activation function, one max-pooling layer, one flattening layer, and one BLSTM layer, as illustrated in Fig. 3. The neural processors of the lower layers attain local features of the inertial signal to signify the elementary motion in physical activity; whilst, higher layer neural processors extract a better abstraction of the motion with higher-level features and temporal analysis. To better preserve the temporal sequence for BLSTM analysis, each data sequence is sub-segmented into sub-windows and each sub-window is fed into feature extraction structure in the chronological order they are in, see Fig. 3. Hence, each sub-sequence, appearing in the sub-window, has its own feature extraction flow. In each flow, convolutional layers read each sub-sequence using a kernel or neuron that reads in small block at a time and strides across the entire data, see Fig. 4. Each read results in an input, that is sub-sequence in this case, to be projected onto a feature map, representing the internal interpretation of the input. Since each convolutional layer contains multiple neurons/kernels, multiple feature maps will be constructed after every layer and concatenated. It is worth noting that to avoid very long training time for each convolution flow for different weights, the same weights are shared in each layer.

Through the convolution operations, those local dependencies in the inertial data could be apprehended. Hence, the correlation between nearby signal points could be pictured, revealing the structure of the signal pattern. Next, the max-pooling layer is implemented to downsample each feature map independently for a summarized version of the captured features. Besides, pooling helps for the model’s invariance to local translations of the input. With this property, a slight translational variance of the input data will not affect the values of the pooled output.

|  |
| --- |
| C:\Users\User\Desktop\MCO files\chewYS_paper\journal\Process2flows-2.png |
| Fig 3: The proposed architecture |

Since convolutional layers unearth those underlying patterns in the input signal, this CNN model is able to encapsulate the tiny changes in the motion signal. These changes in sequential form are substantial to characterize activity motion. Hence, the RNN variant model is included in the proposed architecture to build time dynamics for the feature map by analyzing the underlying sequential pattern in the spatial-temporal feature map.

In the proposed architecture, bidirectional LSTM (BLSTM) is adopted since it can have better prediction using both past and future information, i.e. utilizing information from the previous and upcoming frames [36]. BLSTM is just like stacking two LSTM on top of each other, illustrated in Fig. 5. One LSTM moves in the forward direction, while the other one moves in the opposite, i.e. backward direction. Then, the outputs of LSTMs are fused and computed as BLSTM output. After then, the deep features of BLSTM are extracted and fed into machine learning classifiers for activity prediction.

|  |
| --- |
|  |
| Fig. 4: One-dimensional convolution on sub-sequence signal |

|  |
| --- |
| Deep Dive into Bidirectional LSTM | i2tutorials |
| Fig. 5: BLSTM architecture [37] |

## Formulation

In this work, both CNN and BLSTM are implemented [30], [34], [38]. The process of the convolutional layer is as followed:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

where denotes convolutional output, called feature map, at th layer, is the activation function, i.e. Rectified Linear Unit in this case, refers to the bias term, *W* is the weight from the previous layer, *V* is the input vector of the inertial signal.

The pooling layer performs downsampling to the generated feature maps by summarizing the presence of features in patches of the feature map. Maximum pooling is used in this work:

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

where *P* is the output of the pooling operation, and is the *i*th patch of a feature map from *n*th convolutional layer. Next, *P* is flattened and fed for temporal dynamics analysis via LSTM nodes. In the LSTM node, there are memory cells and four gates (i.e. forget gate, input gate, input modulation gate, and output gate). Forget gate at any given timestep is formulated as below:

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

where is the forget gate output at *n* layer at timestep and is the sigmoid function, denotes the weight of the connection at forget gate, is the LSTM output from the previous layer, is the input vector, and is the bias term for the forget gate. The next gate is the input gate:

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

where is the input gate output at *n* layer at timestep and is the sigmoid function, denotes the weight of the connection at the input gate, is the LSTM output from the previous layer, is the input vector, and is the bias term for the input gate. Similar to the input gate and forget gate, the output gate exhibits similar equations:

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

where denotes the output gate output at *n* layer at timestep and is the sigmoid function, denotes the weight of the connection at the output gate, is the LSTM output from the previous layer, is the input vector, and is the bias term for the output gate. The input modulation gate is a function of the input vector and the previous state output:

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

where denotes the input modulation gate output at *n* layer at timestep with function, denotes the weight of the connection at the input modulation gate which is based on the previous state, is the LSTM output from the previous layer, is the input vector, and is the bias term for the input modulation gate based on the previous state gate.

The state gate or memory cell consists of two terms: the previous memory cell state which is modulated by forget gate , and input modulation gate which is modulated by input gate at timestep :

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

The LSTM outputs the combination of the output gate and state gate, with the following equation:

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

where is the output gate’s output at *n* layer with function and is the state gate’s output at *n* layer. Each LSTM node generates two values for the next LSTM nodes. The subsequent LSTM node will then use both information accordingly to update their state and eventually the update status of the whole network. This allows the LSTM network takes the ability to consider past information.

As mentioned early, BLSTM is alike stacking two LSTM on top of each other. In other words, BLSTM has both forward sequences and backward sequences in the hidden layer. At time *t*, the hidden layer and the input layer can be defined as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (9) |
|  |  | (10) |
|  | ) | (11) |

where denotes the forward sequence, denotes the backward sequence of LSTM operation, refers to the BLSTM output, A is the activation function used in the network, W is the weight of the connection and B is the bias term. These forward and backward sequences allow the LSTM nodes to take in previous and subsequent information to update its state, this will subsequently update the state of the whole network. This gives the BLSTM’s properties to use past and future information to effectively output a deeper representation of a set of data inputs.

# Experiment

The effectiveness of the proposed architecture is evaluated by using two publicly available databases collected using inertial sensors embedded on smartphones. These databases are: (1) Wireless Sensor Data Mining (WISDM) dataset and (2) UCI HAR dataset. In this work, a support vector machine (SVM) is adopted as the machine learning algorithm to classify the extracted deep features for human activity prediction.

## Databases

WISDM dataset is collected by a research team of Kwapisz et al. which popularizes the application of smartphone-based human activity recognition [10]. This dataset contains six different activity classes performed by volunteers with the Android smartphone in the pocket. On the other hand, the UCI dataset also contains six different activities and the location of the smartphone is at the waist. Table 1 summarizes the details of these datasets.

|  |  |  |
| --- | --- | --- |
| Table 1  Details of Databases | | |
| **Remark** | **UCI** | **WISDM** |
| **Activities** | Walking,  Walking upstairs,  Walking downstairs,  Sitting,  Standing Laying down | Walking,  Jogging,  Stairs-Up,  Stairs-Down,  Sitting, and Standing |
| **Location of smartphone** | Waist | Front leg pocket |
| **Data used in this work** | Triaxial gravity acceleration data  Triaxial body acceleration  Triaxial angular velocity | Triaxial gravity acceleration data |
| **Time steps** | 2.56 seconds (128 timesteps) | 10 seconds (200 timesteps) |

## Experimental Setup

In this work, experiments are conducted in the environment of an Intel Core i7-8550u 1.85GHz processor laptop and 16GB RAM with a Windows 10 operating system. We implement the training-testing split protocol: 70% of each dataset is used for training and the remaining 30% is used for testing the system performance. Among the training data, 20% is selected for system validation and the remaining is for model training.

Various hyperparameters have been tested. For example, different numbers of kernels in the convolutional layers are examined from 32 kernels to 256 kernels; different hidden nodes in the BLSTM layer are studied from 32 units to 400 units. Based on the validation performance, the optimal parameters are determined and adopted for system performance testing. In UCI, there are 64 kernels in each convolutional layer and 256 nodes in the BLSTM layer; whereas in WISDM, 100 kernels and 240 nodes are employed. The model is trained with a learning rate of 0.001, weight decay of 0.0001 to minimize the overfitting issue and 30 epochs with a batch size of 32 in UCI and 80 in WISDM.

The confusion matrix can be used for system performance analysis. The conventional confusion matrix is portrayed in Fig. 6. Confusion matrix reports the counts of true positives, false positives, true negatives, and false negatives:

* True Positive (TP): the label belongs to the class, and it is CORRECTLY predicted
* False Positive (FP): the label does not belong to the class, but it is classified/ predicted as positive
* True Negative (TN): the label does not belong to the class, and it is CORRECTLY predicted
* False Negative (FN): the label belongs to the class, but it is classified/ predicted as negative

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | PREDICTED CLASS | | |  |
| 1 | 2 | 3 | FN |
| ACTUAL CLASS | 1 | A | B | C | B+C |
| 2 | D | E | F | D+F |
| 3 | G | H | I | G+H |
|  | FP | D+G | B+H | C+F |  |

Fig. 6. Performance Measurement Confusion Matrix**.**

*Classification Accuracy* is usually used to measure the overall accuracy of a classification model. However, it is not substantial to decide the model is well enough to make robust predictions, i.e. it is not able to discriminate kinds of misclassifications. So, performance metrics such as precision, recall, F1-score, and AUC are topped up for performance measurement. *Precision* is the number of positive predictions divided by the total number of positive class values predicted. Precision can be thought of as a measure of a model’s efficacy. A low precision indicates a large number of False Positive.

On the other hand, r*ecall* is a measure of a model’s completeness/ accuracy. It is the number of positive predictions divided by the number of positive class values in the test data. A low recall shows many False Negatives. *F1 score*, also known as *F measure*, is a harmonic mean of precision and recall, conveying the balance between the two measures. The value of the *Area under the Receiver Operating Characteristic Curv*e (AUC) represents the robustness of a classification model. A higher value of AUC indicates the probability of a classifier to rank a randomly chosen positive instance higher than a randomly chosen negative instance. Fig. 7 illustrates various performance measures obtained from the confusion matrix. In this work, the empirical results are testified based on performance evaluation metrics [39], [40] of (i) true positive (TP) rate, (ii) false positive (FP) rate, (iii) precision, (iv) recall (v) area under the Receiver Operating Characteristic Curve (AUC), (vi) F1 score and (vii) classification accuracy.

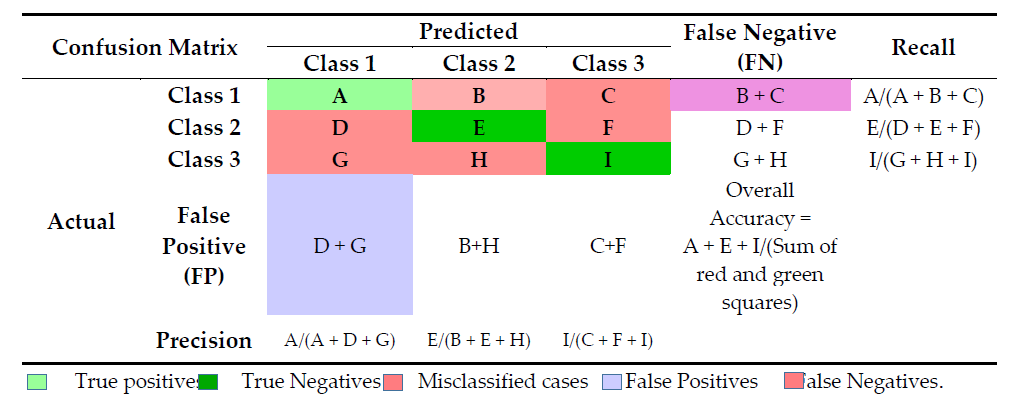


Fig. 7. Performance measures computed from the confusion matrix. [40]

# Evaluation and Discussion

The extracted deep features from the proposed architecture are classified using Support Vector Machine (SVM) with different kernels. Table 2 and Table 3 record the system performance measures of different types of SVMs for UCI and WISDM databases, respectively. Fig. 8 and 9 illustrate the confusion matrices of SVM-based classifiers for UCI and WISDM databases, respectively

From the tables, we can observe that the proposed classification model, comprising the proposed feature extraction architecture and SVM classifier, shows promising performance in predicting human activity based on inertial data captured by smartphones. The model exhibits relatively good classification accuracy in UCI and WISDM databases, i.e. about 92% and 87% accuracy respectively. The adoption of SVM as a machine learning algorithm enables a decision hyperplane creation with the maximum margin. The maximization of margin distance of inter-class data provides a certain degree of reinforcement so that the future unknown data can be classified with more confidence.

Further, we also could observe that the extracted deep features showing comparable efficacy in representing the human activity with different kernels in SVM classifier in both databases. However, the proposed classification model is slightly inferior in classifying sitting and standing actions in the UCI database, as recorded in Table 2. Lower precision and recall are observed in sitting and standing activities. This denotes that there are more false positives and false negatives in predicting sitting and standing inertial data. The key reason for such higher false positive and negative rates is due to the proposed model is not well enough to distinguish inertial data between sitting and standing classes in the UCI database, as illustrated in the confusion matrix in Fig. 8.

On the other hand, the proposed classification model suffers inferiority when dealing with inertial data of downstairs and upstairs activities in the WISDM database. The model has a high possibility to falsely predict a downstairs activity as upstairs and walking activities, leading to low recall in the downstairs class. The high false negatives can also be observed in the confusion matrix in Fig. 9. Besides, it is observed that low precision is obtained in the upstairs activity. The model falsely predicts other activity as the upstairs class, causing more false positives. From the confusion matrix in Fig. 9, the proposed model confuses downstairs and walking data as the upstairs activity. The position of the smartphone could be one of the reasons for the performance discrepancy between these two databases. In the UCI database, the smartphone is placed at the body waist, whereas the smartphone is placed in the front leg pocket. Undeniably, the position of the smartphone is one of the factors that could greatly influence the quality of data, affecting the accuracy of the classification model [41]. Overall, the F1 score is pretty good for the proposed model, 0.922, and 0.875 for UCI and WISDM databases, respectively.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 2  Performance Measures of UCI database. | | | | | | | | |
| **SVM’s Kernel** | **Class** | **TP rate** | **FP rate** | **Precision** | **Recall** | **F1 Score** | **AUC** | **Accuracy** |
| Linear | Walking | 0.942 | 0.004 | 0.977 | 0.942 | 0.959 | 0.969 | 92.1276 |
| Walking upstairs | 0.947 | 0.002 | 0.989 | 0.947 | 0.967 | 0.972 |
| Walking downstairs | 0.998 | 0.017 | 0.907 | 0.998 | 0.950 | 0.990 |
| Sitting | 0.819 | 0.028 | 0.855 | 0.819 | 0.837 | 0.896 |
| Standing | 0.872 | 0.041 | 0.824 | 0.872 | 0.847 | 0.916 |
| Laying | 0.963 | 0.002 | 0.989 | 0.963 | 0.975 | 0.980 |
| **Weighted average** | **0.921** | **0.016** | **0.923** | **0.921** | **0.922** | **0.953** |
| Radial | Walking | 0.942 | 0.003 | 0.983 | 0.942 | 0.962 | 0.969 | 92.0258 |
| Walking upstairs | 0.943 | 0.003 | 0.984 | 0.943 | 0.963 | 0.970 |
| Walking downstairs | 0.998 | 0.020 | 0.893 | 0.998 | 0.943 | 0.989 |
| Sitting | 0.845 | 0.031 | 0.845 | 0.845 | 0.845 | 0.907 |
| Standing | 0.852 | 0.037 | 0.836 | 0.852 | 0.844 | 0.907 |
| Laying | 0.957 | 0.002 | 0.990 | 0.957 | 0.973 | 0.978 |
| **Weighted average** | **0.920** | **0.016** | **0.922** | **0.920** | **0.921** | **0.952** |
| Sigmoid | Walking | 0.942 | 0.003 | 0.983 | 0.942 | 0.962 | 0.969 | 92.0258 |
| Walking upstairs | 0.941 | 0.003 | 0.984 | 0.941 | 0.962 | 0.969 |
| Walking downstairs | 0.998 | 0.020 | 0.891 | 0.998 | 0.942 | 0.989 |
| Sitting | 0.837 | 0.029 | 0.853 | 0.837 | 0.845 | 0.904 |
| Standing | 0.861 | 0.039 | 0.831 | 0.861 | 0.846 | 0.911 |
| Laying | 0.957 | 0.002 | 0.990 | 0.957 | 0.973 | 0.978 |
| **Weighted average** | **0.920** | **0.016** | **0.922** | **0.920** | **0.921** | **0.952** |

|  |  |
| --- | --- |
|  |  |
| (a) | (b) |
|  | |
| (c) | |
| Fig. 8. Confusion matrices of SVM with kernel (a) linear, (b) radial and (c) sigmoid for UCI database | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 3  Performance Measures of WISDM database. | | | | | | | | |
| **SVM’s Kernel** | **Class** | **TP rate** | **FP rate** | **Precision** | **Recall** | **F1 Score** | **AUC** | **Accuracy** |
| Linear | Downstairs | 0.540 | 0.028 | 0.678 | 0.540 | 0.601 | 0.756 | 86.938 |
| Jogging | 0.955 | 0.001 | 0.997 | 0.955 | 0.976 | 0.977 |
| Sitting | 0.996 | 0.011 | 0.870 | 0.996 | 0.929 | 0.992 |
| Standing | 0.808 | 0.000 | 0.990 | 0.808 | 0.890 | 0.904 |
| Upstairs | 0.687 | 0.046 | 0.649 | 0.687 | 0.668 | 0.820 |
| Walking | 0.928 | 0.083 | 0.864 | 0.928 | 0.895 | 0.922 |
| **Weighted average** | **0.869** | **0.039** | **0.870** | **0.869** | **0.867** | **0.915** |
| Radial | Downstairs | 0.542 | 0.019 | 0.754 | 0.542 | 0.630 | 0.761 | 87.7278 |
| Jogging | 0.966 | 0.003 | 0.993 | 0.966 | 0.979 | 0.981 |
| Sitting | 0.966 | 0.010 | 0.882 | 0.966 | 0.936 | 0.993 |
| Standing | 0.819 | 0.001 | 0.984 | 0.819 | 0.894 | 0.909 |
| Upstairs | 0.690 | 0.048 | 0.642 | 0.690 | 0.665 | 0.821 |
| Walking | 0.938 | 0.080 | 0.870 | 0.938 | 0.903 | 0.929 |
| **Weighted average** | **0.877** | **0.038** | **0.878** | **0.877** | **0.875** | **0.920** |
| Sigmoid | Downstairs | 0.535 | 0.020 | 0.750 | 0.535 | 0.625 | 0.758 | 87.6671 |
| Jogging | 0.966 | 0.004 | 0.990 | 0.966 | 0.978 | 0.981 |
| Sitting | 0.966 | 0.010 | 0.882 | 0.996 | 0.936 | 0.993 |
| Standing | 0.819 | 0.001 | 0.981 | 0.819 | 0.892 | 0.909 |
| Upstairs | 0.686 | 0.047 | 0.642 | 0.686 | 0.663 | 0.819 |
| Walking | 0.939 | 0.080 | 0.871 | 0.939 | 0.904 | 0.930 |
| **Weighted average** | **0.877** | **0.038** | **0.877** | **0.877** | **0.874** | **0.919** |

|  |  |
| --- | --- |
|  |  |
| (a) | (b) |
|  | |
| (c) | |
| Fig. 8. Confusion matrices of SVM with kernel (a) linear, (b) radial and (c) sigmoid for WISDM database | |

## Additional Experiments: With Machine Learning Algorithms

In this section, different kinds of machine learning algorithms will be examined in the extracted deep features. We implement these classifiers (with default parameters) in the Waikato Environment for Knowledge Analysis (Weka) Experimenter Workbench. Table 4 and 5 record the classification results of different machine learning algorithms for UCI and WISDM databases, respectively.

From the empirical results, it is observed that Random Forest and Bayesian Network give remarkable results in both databases with good precision and recall, leading to a high F1 score. Their classification performances are comparable with the SVM classifier. The scores signify that the classification models with Random Forest and Bayesian Network are able to perform a promising generate low false positives and negatives.

Random Forest is basically a collection of Decision Trees. Random Forest randomly picks observations and specific features to build multiple decision trees. Those results are then aggregated into one final result. Undeniably, Random Forest is more robust than a decision tree (i.e. J48 in this work), as revealed in the empirical results too. Random Forest aggregates multiple decision trees to regulate overfitting as well as error due to bias, yielding encouraging results. On the other hand, the success of Bayesian Network is because of the integration using Bayes law with respect to the observed information to figure a posterior by encompassing prior information. Based on the posterior, Bayesian Network could have a good prediction.

Table 4

Weighted Average of True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall and F1 Score of different machine learning algorithms for UCI database.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Classifier** | **Weighted Average** | | | | |
| **TPR** | **FPR** | **Precision** | **Recall** | **F1 Score** |
| Logistic Regression | 0.906 | 0.020 | 0.911 | 0.906 | 0.907 |
| Random Forest (RF) | **0.923** | **0.016** | **0.925** | **0.923** | **0.923** |
| Naive Bayes | 0.902 | 0.020 | 0.911 | 0.902 | 0.904 |
| Multilayer Perceptron | 0.916 | 0.017 | 0.919 | 0.916 | 0.916 |
| kNN: Scaled Manhattan | 0.914 | 0.017 | 0.917 | 0.914 | 0.915 |
| kNN: Euclidean | 0.911 | 0.018 | 0.914 | 0.911 | 0.912 |
| Bayesian Network | **0.917** | **0.017** | **0.920** | **0.917** | **0.917** |
| J48 | **0.887** | **0.023** | **0.891** | **0.887** | **0.888** |
| SVM-  radial | **0.920** | **0.016** | **0.922** | **0.920** | **0.921** |

Table 5

Weighted Average of True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall and F1 Score of different machine learning algorithms for WISDM database.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Classifier** | **Weighted Average** | | | | |
| **TPR** | **FPR** | **Precision** | **Recall** | **F1 Score** |
| Logistic Regression | 0.859 | 0.042 | 0.859 | 0.859 | 0.856 |
| Random Forest (RF) | **0.872** | **0.041** | **0.873** | **0.872** | **0.869** |
| Naive Bayes | 0.859 | 0.030 | 0.877 | 0.859 | 0.864 |
| Multilayer Perceptron | 0.862 | 0.038 | 0.867 | 0.862 | 0.862 |
| kNN: Scaled Manhattan | 0.861 | 0.043 | 0.862 | 0.861 | 0.858 |
| kNN: Euclidean | 0.863 | 0.043 | 0.864 | 0.863 | 0.860 |
| Bayesian Network | **0.873** | **0.029** | **0.884** | **0.873** | **0.876** |
| J48 | **0.851** | **0.045** | **0.852** | **0.851** | **0.849** |
| SVM-  radial | **0.877** | **0.038** | **0.878** | **0.877** | **0.875** |

## Additional Experiments: Performance Comparison with Other Approaches

In this experiment, several existing approaches are adopted for performance comparison. The classification accuracies of the approaches are summarized in Table 6 and 7. Generally, the proposed classification models are able to attain superior or comparable performance with the other state-of-the-art approaches. However, the models show slightly inferior to CNN+statistical\_feature approach. The main difference between the proposed models and this CNN-based approach is the later model encompasses statistical features in the extracted deep features before classification. In other words, the approach not only utilizes the deep features extracted from CNN but also embraces the global properties of the time series data for human activity recognition. The augmentation of CNN with statistical features (with data centering preprocess) could improve the system performance by about 2%.

Further, we also can observe from the empirical results that Hierarchical Multi-View Aggregation Network (HMVAN) exhibits higher classification accuracy in the UCI database. HMVAN integrates black-box features with white-box features in a hierarchical multi-view structure. Such hierarchical composition constructs multi-view feature spaces for each sensor from the point of white-box features and black-box features. Those features are then aggregated into a unified representation. Comparing with our proposed models which simply concatenating features in layers, HMVAN is able to better capture the correlation between features.

Table 6

Classification accuracy of various approaches on the UCI database

|  |  |  |
| --- | --- | --- |
| **Model** | **Algorithm/ Architecture** | **Accuracy (%)** |
| Neural Network-based model\*\* [42] | Autoencoder + Random Forest | 76.26 |
| HMVAN\*\* [43] | Hierarchical Multi-View Aggregation Network (two-layer multi-view model without views of position) | 95.5 |
| Simplified HMVAN\*\* [43] | Hierarchical Multi-View Aggregation Network (one-layer multi-view model) | 94.7 |
| CHMM\*\* [44] | Hierarchical Continuous HMM | 93.18 |
| CNN based model\*\* [23] | CNN + Statistical features + data centering | 97.63% |
| CNN\* [25] | C(96)-MAX-C(96)-MAX-C(96)-MAX-DL(1000)-DL(6) | 89.41 |
| CNN\* [8] | C(128)-MAX-C(128)-MAX-DL(384)-DL(6) | 89.24 |
| LSTM\* [33] | L(100)-L(100)-DL(128)-DL(6) | 89.79 |
| Bidir-LSTM\* [34] | B(L(28))-B(L(28)-B(L(28))-DL(128)-DL(6) | 89.07 |
| Bidir-LSTM\* [35] | B(L(175))-B(L(175)-B(L(175))-DL(128) | 87.41 |
| Proposed model + SVM\* | C(64)-C(64)-C(64)-MAX-B(L(256)) | 92.03 |
| Proposed model + RF\* | C(64)-C(64)-C(64)-MAX-B(L(256)) | 92.26 |

\* denotes models are trained using 30 epochs

\*\* denotes the empirical results reported in the respective papers

Table 7

Classification accuracy of various approaches on the WISDM database

|  |  |  |
| --- | --- | --- |
| **Model** | **Algorithm/ Architecture** | **Accuracy (%)** |
| Neural Network-based model \*\* [42] | Autoencoder + Dropout | 85.36 |
| ISAR\*\* [45] | Impersonal Smartphone-based Activity Recognition | 75.22 |
| CNN based model\*\* [23] | CNN + Statistical features | 93.32 |
| Simplified HMVAN\*\* [43] | Hierarchical Multi-View Aggregation Network (one-layer multi-view model) | 93.1 |
| CNN\* [25] | C(96)-MAX-C(96)-MAX-C(96)-MAX-DL(1000)-DL(6) | 74.79 |
| CNN\* [8] | C(128)-MAX-C(128)-MAX-DL(384)-DL(6) | 85.98 |
| LSTM\* [33] | L(100)-L(100)-DL(80)-DL(6) | 79.62 |
| Bidir-LSTM\* [34] | B(L(28))-B(L(28)-B(L(28))-DL(80)-DL(6) | 72.98 |
| Bidir-LSTM\* [35] | B(L(175))-B(L(175)-B(L(175))-DL(80)-DL(6) | 82.09 |
| Proposed model + SVM\* | C(100)-C(100)-C(100)-MAX-B(L(240)) | 87.73 |
| Proposed model + RF\* | C(100)-C(100)-C(100)-MAX-B(L(240)) | 87.21 |

\* denotes models are trained using 30 epochs

\*\* denotes the empirical results reported in the respective papers

# Concluding Remark

In this paper, we propose a spatial-temporal deep feature extraction architecture that integrates the one-dimensional convolutional model with a bi-directional long-short-term-memory (LSTM) model. This stacking deep model is capable of extracting low level to higher-level features of inertial data for human activity recognition while encapsulating both spatial and temporal state dependencies of inertial data to predict human activity. The employment of support vector machine in the proposed feature extraction architecture enables good classification accuracy in both UCI and WISDM databases. The support vector machine creates a discriminant decision hyperplane with the maximum margin which supports a certain extent of reinforcement to facilitate higher confidence to classify future unknown data. Besides, we also examine various machine learning algorithms to perform classification. Bayesian Network and Random Forest exhibit promising performance. In future work, we plan to explore the deep hierarchical multi-view aggregative analysis for smartphone-based human activity recognition.

# References

[1] “WHO | Prevalence of insufficient physical activity,” *WHO*, 2018.

[2] F. W. Booth, C. K. Roberts, and M. J. Laye, “Lack of exercise is a major cause of chronic diseases,” *Compr. Physiol.*, vol. 2, no. 2, pp. 1143–1211, Apr. 2012.

[3] G. Cattadori, C. Segurini, A. Picozzi, L. Padeletti, and C. Anzà, “Exercise and heart failure: an update,” *ESC Heart Failure*, vol. 5, no. 2. Wiley-Blackwell, pp. 222–232, Apr-2018.

[4] R. A. Voicu, C. Dobre, L. Bajenaru, and R. I. Ciobanu, “Human physical activity recognition using smartphone sensors,” *Sensors (Switzerland)*, vol. 19, no. 3, Feb. 2019.

[5] R. A. Carels, L. A. Darby, S. Rydin, O. M. Douglass, H. M. Cacciapaglia, and W. H. O’Brien, “The relationship between self-monitoring, outcome expectancies, difficulties with eating and exercise, and physical activity and weight loss treatment outcomes,” *Ann. Behav. Med.*, vol. 30, no. 3, pp. 182–190, Dec. 2005.

[6] M. M. MacPherson, K. J. Merry, S. R. Locke, and M. E. Jung, “Effects of Mobile Health Prompts on Self-Monitoring and Exercise Behaviors Following a Diabetes Prevention Program: Secondary Analysis From a Randomized Controlled Trial.,” *JMIR mHealth uHealth*, vol. 7, no. 9, p. e12956, Sep. 2019.

[7] R. Poppe, “A survey on vision-based human action recognition,” *Image Vis. Comput.*, vol. 28, no. 6, pp. 976–990, 2010.

[8] S. M. Lee, S. M., Cho, H., & Yoon, “Human Activity Recognition From Accelerometer Data Using Convolutional Neural Network,” *IEEE Int. Conf. Big Data Smart Comput. (BigComp).*, vol. 62, pp. 131–134, 2017.

[9] K. Lee and M. P. Kwan, “Physical activity classification in free-living conditions using smartphone accelerometer data and exploration of predicted results,” *Comput. Environ. Urban Syst.*, vol. 67, no. September 2017, pp. 124–131, 2018.

[10] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” *ACM SIGKDD Explor. Newsl.*, 2011.

[11] L. Gao, A. K. Bourke, and J. Nelson, “Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems,” *Med. Eng. Phys.*, vol. 36, no. 6, pp. 779–785, 2014.

[12] C. Li, “DigitalCommons@USU Wearable Computing: Accelerometer-Based Human Activity Classification Using Decision Tree,” 2017.

[13] D. Ledger and D. McCaffrey, “Inside wearables: how the science of human behavior change offers the secret to long-term engagement,” 2014.

[14] S. Kumari and S. K. Mitra, “Human action recognition using DFT,” in *Proceedings - 2011 3rd National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, NCVPRIPG 2011*, 2011.

[15] X. Peng, C. Zou, Y. Qiao, and Q. Peng, “Action recognition with stacked fisher vectors,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014.

[16] A. Jalal, S. Kamal, and D. Kim, “A Depth Video-based Human Detection and Activity Recognition using Multi-features and Embedded Hidden Markov Models for Health Care Monitoring Systems,” *Int. J. Interact. Multimed. Artif. Intell.*, vol. 4, no. 4, p. 54, 2017.

[17] M. C. Kwon and S. Choi, “Recognition of Daily Human Activity Using an Artificial Neural Network and Smartwatch,” *Wirel. Commun. Mob. Comput.*, 2018.

[18] S. Sani, N. Wiratunga, and S. Massie, “Learning Deep Features for kNN-Based Human Activity Recognition,” 2017.

[19] K. W. Ching, M. M. Singh, and Z. F. Zaaba, “Human activity recognition (HAR) for wearable sensors with classification techniques,” *Adv. Sci. Lett.*, vol. 23, no. 5, pp. 4206–4210, May 2017.

[20] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones,” in *ESANN 2013 proceedings, 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013.

[21] A. Bayat, M. Pomplun, and D. A. Tran, “A study on human activity recognition using accelerometer data from smartphones,” *Procedia Comput. Sci.*, vol. 34, pp. 450–457, 2014.

[22] T. Brezmes, J. L. Gorricho, and J. Cotrina, “Activity recognition from accelerometer data on a mobile phone,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5518 LNCS, no. PART 2, pp. 796–799, 2009.

[23] A. Ignatov, “Real-time human activity recognition from accelerometer data using Convolutional Neural Networks,” *Appl. Soft Comput. J.*, vol. 62, pp. 915–922, 2018.

[24] J. W. Lockhart, “The Benefits of Personalized Data Mining Approaches to Human Activity Recognition with Smartphone Sensor Data,” p. 46, 2014.

[25] 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, 2016.

[26] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensor-based activity recognition: A survey,” *Pattern Recognit. Lett.*, 2019.

[27] X. Shi, Y. Li, F. Zhou, and L. Liu, “Human Activity Recognition Based on Deep Learning Method,” *2018 Int. Conf. Radar, RADAR 2018*, 2018.

[28] H. Friday Nweke, T. Ying Wah, and U. Alo, “Deep Learning Algorithms for Human Activity Recognition using Mobile and Wearable Sensor Networks: State of the Art and Research Challenges Mobile Cloud Computing View project Novel Deep Learning Architecture for Physical Activities assessment, mental Res,” vol. 105, pp. 233–261, 2018.

[29] A. Murad and J. Y. Pyun, “Deep recurrent neural networks for human activity recognition,” *Sensors (Switzerland)*, 2017.

[30] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, “Object recognition with gradient-based learning,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1999.

[31] Y. LeCun *et al.*, “Backpropagation Applied to Handwritten Zip Code Recognition,” *Neural Comput.*, 1989.

[32] M. Zeng *et al.*, “Convolutional Neural Networks for human activity recognition using mobile sensors Article,” pp. 381–388, 2014.

[33] Y. Chen, K. Zhong, J. Zhang, Q. Sun, and X. Zhao, “LSTM Networks for Mobile Human Activity Recognition,” no. Icaita, pp. 50–53, 2016.

[34] S. Yu and L. Qin, “Human activity recognition with smartphone inertial sensors using bidir-LSTM networks,” *Proc. - 2018 3rd Int. Conf. Mech. Control Comput. Eng. ICMCCE 2018*, pp. 219–224, 2018.

[35] F. Hernández, L. F. Suárez, J. Villamizar, and M. Altuve, “Human Activity Recognition on Smartphones Using a Bidirectional LSTM Network,” *2019 22nd Symp. Image, Signal Process. Artif. Vision, STSIVA 2019 - Conf. Proc.*, 2019.

[36] A. Ogawa and T. Hori, “Error detection and accuracy estimation in automatic speech recognition using deep bidirectional recurrent neural networks,” *Speech Commun.*, 2017.

[37] “Deep Dive into Bidirectional LSTM | i2tutorials.” [Online]. Available: https://www.i2tutorials.com/deep-dive-into-bidirectional-lstm/. [Accessed: 06-May-2020].

[38] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities (associative memory/parallel processing/categorization/content-addressable memory/fail-soft devices),” *Biophysics (Oxf).*, 1982.

[39] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognit. Lett.*, 2006.

[40] M. Ali, D. H. Son, S. H. Kang, and S. R. Nam, “An accurate CT saturation classification using a deep learning approach based on unsupervised feature extraction and supervised fine-tuning strategy,” *Energies*, 2017.

[41] W. S. Lima, E. Souto, K. El-Khatib, R. Jalali, and J. Gama, “Human activity recognition using inertial sensors in a smartphone: An overview,” *Sensors (Switzerland)*, 2019.

[42] B. Kolosnjaji and C. Eckert, “Neural network-based user-independent physical activity recognition for mobile devices,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015.

[43] X. Zhang, Y. Wong, M. S. Kankanhalli, and W. Geng, “Hierarchical multi-view aggregation network for sensor-based human activity recognition,” *PLoS One*, 2019.

[44] C. A. Ronao and S. B. Cho, “Recognizing human activities from smartphone sensors using hierarchical continuous hidden Markov models,” *Int. J. Distrib. Sens. Networks*, 2017.

[45] T. Dungkaew, J. Suksawatchon, and U. Suksawatchon, “Impersonal smartphone-based activity recognition using the accelerometer sensory data,” in *Proceeding of 2017 2nd International Conference on Information Technology, INCIT 2017*, 2017.

1. \*Corresponding author. E-mail: editorial@iospress.nl. Check if the checkbox in menu *Tools/Options/Compatibility/Lay out footnotes like Word 6.x/95/97* is selected if you make a footnote for the corresponding author. [↑](#footnote-ref-1)