The Deep Learning-based Human Activity Recognition Using Smart Wearable Sensors : A Tutorial

Sakorn Mekruksavanich[1] and Anuchit Jitpattanakul[2*]

[1]Department of Computer Engineering, School of Information and Communication Technology
University of Phayao, Phayao, Thailand
sakorn.me@up.ac.th

[2]Intelligent and Nonlinear Dynamic Innovations Research Center
Department of Mathematics, Faculty of Applied Science
King Mongkut's University of Technology North Bangkok, Bangkok, Thailand
anuchit.j@sci.kmutnb.ac.th

Received: December 4, 2021; Accepted: February 15, 2022; Published: February 28, 2022

Abstract

Human activity recognition (HAR) mechanisms that distinguish human behavior utilizing wearable sensors have advanced significantly over several years. Not only have state-of-the-art techniques ignored hand-crafted features in favor of end-to-end deep learning approaches, but best practices for designing experiments, preparing datasets, and assessing activity recognition systems have changed in lockstep. This tutorial will provide an in-depth, hands-on introduction to the topic of sensor-based HAR for those who are new to it. We will concentrate on deep learning-based HAR in this tutorial utilizing data from intelligent wearable sensor devices. This tutorial introduces the SDL-HAR framework, which provides a general-purpose framework for data preprocessing, data generation, model development, and evaluation. We describe each aspect of the provided system in-depth, offer references to relevant research, and explain the community's best practice methodologies for activity identification. Two exemplary deep learning approaches, convolutional neural network (CNN) and long short-term memory neural network (LSTM), are deployed in this lesson using state-of-the-art public HAR datasets. Additionally, this tutorial highlights the problems and future research directions of sensor-based HAR.

Keywords: human activity recognition, deep learning, wearable sensor, hand-on tutorial, convolutional neural network, long short-term memory neural network

1 Introduction

Due to technological advances of wearable sensing innovations, smart wearable devices (e.g., smartwatches, smartphones, and smart glasses) have become one of the most beneficial ubiquitous and pervasive computing devices, owing to their ability to assist us in our everyday lives with healthcare. Wearable devices are electronic gadgets that people wear constantly and ubiquitously to collect or monitor biometric data relating to their health or exercise based on their activity [1]. Wearable gadgets that provide

Research Briefs on Information & Communication Technology Evolution (ReBICTE), Vol. 8, Article No. 1 (February 28, 2022)
DOI:*10.22667/ReBiCTE.2022.02.28.001*

*Corresponding author: 1518 Pracharat 1 Road, Wongsawang, Bangsue, Bangkok 10800 Thailand, Tel: +662-555-2000 (ext. 4510, 4511)

biometric monitoring are a significant source of information collection [2]. They will capture data of various sorts and from a range of situations continuously and uninterruptedly. For example, in the health sector, the detection of physical activity based on wearable sensor data has aided in avoiding adverse outcomes associated with dietary misbehavior. For instance, tracking the amount of time a person spends engaged in eating-related activities could be a valuable criterion for diagnosing and treating obesity, diabetes, cancer, and cardiovascular disease [3, 4, 5]. Another example is smoking detection data, which a person or a health professional could also utilize to control their smoking habit by better understanding their daily smoking habits [6].

At the moment, intelligent wearables also provide a range of sensors, including accelerometers, gyroscopes, magnetometers, and many other environmental sensors. These monitors have been used as a feasible and attractive advantage over traditional wearable devices to investigate human behavior identification. The sensor-based HAR solution could be regarded as a machine learning (ML) framework built on the individual's smartphone and continually identifies the individual's actions. In contrast, the smartphone is linked to a portion of the participant's body. By incorporating state-of-the-art machine learning approaches such as decision trees, support vector machines, naïve Bayes, and ANNs into traditional movement identification systems, substantial improvements have been accomplished [7]. Nevertheless, such conventional machine learning algorithms could depend on heuristic hand-crafted feature extraction, often constrained by domain expertise. Due to this constraint, typical machine learning algorithms fail to perform classification accuracy and other assessment criteria.

Deep learning (DL) methodologies are applied in HAR research to solve the issues mentioned above. Instead of manually extracting features from raw sensor data, they may be learned using many hidden layers. These techniques' deep architecture allows the extraction of deep high-level features that are more appropriate for complicated issues, including HAR. Lately, DL techniques have developed a comprehensive HAR based on smartphones [8, 9].

Currently, convolutional neural networks (CNNs) are one type of DL model that has demonstrated outstanding performance in image classification, speech recognition, and text analysis [10]. When used to classify time-series data, such as data in HAR, the CNN offers two benefits over other standard ML models: local dependence and scale invariance [11]. One-dimensional CNNs have been explored, and it has been demonstrated that these DL models could be employed to address the HAR issue with superior performance indicators than traditional ML techniques [12]. Long-term memory (LSTM) networks have also been proposed to solve the HAR problem since sensor data is time-series data with temporal dependencies [13]. The LSTM network can uncover temporal correlations in data without confounding time steps, as the CNN network does.

There is no comprehensive tutorial on deep learning-based HAR using wearable sensors available at the moment. There are various highly referenced articles on the subject, for example [7, 8]. However, these works do not describe the design, implementation, and assessment of HAR systems invariably. Considering their narrow emphasis on particular activity identification challenges and their tendency to propose a single optimal solution to the challenges at present, these works also lack the range of knowledge that a practical tutorial should provide. Only a few studies compare and contrast various design alternatives, which we consider critical for educating and informing beginners to the aspect of HAR.

This article addresses this knowledge gap by delivering the first tutorial on HAR using wearable sensor data and DL. The tutorial explains the standard techniques and practice guidelines for developing, deploying, and assessing HAR solutions. The architecture given here is broad and is not confined to activity identification through particular sensors. The paper is supplemented by publicly accessible datasets and a feature-rich activity identification system written in Python and accompanying modules for educational reasons.

2 Related Works

Significant advances in the classification of human behavior have enabled continued growth in domains such as medical bioinformatics [14, 15], industrial production [16, 17], environmental sensing [18, 19], and sports assessment [20, 21]. Nevertheless, several critical fundamentals discussed in [22], upon which most studies in this area have been founded, are no longer state of the art. Since deep learning has worked its way into activity identification, previous algorithms' requirements are often not properly relevant.

Feature engineering, commonly used in machine learning techniques, is no longer appropriate because [12] demonstrated that raw sensor information is handled employing deep learning models. However, recent literature such as Chen et al. [23] indicates that it could be advantageous in certain areas and situational factors or when utilizing a particular deep learning algorithm. Conversely, traditional machine learning uses first-order statistics (mean, variance, median values, etc.). In comparison, when the feature selection technique is used in conjunction with deep learning, the determined features are often of a higher order.

Not only technological advances, including such transfer learning [24, 25], data augmentation [26, 27], and active learning [28], but also the constant search for some more effective network frameworks [29, 30, 13, 31], are becoming an increasingly essential part of the HAR society, and are thus leading studies in the areas of deep learning [32]. There are two methods of HAR: (1) video-based, as in [33], and (2) inertial sensor-based. At the very least, an inertial sensor comprises an accelerometer and a gyroscope but is sometimes augmented with a magnetometer. Complementary sensor technologies, such as temperature or light sensors, could increase these records collected using these sensors. By integrating an accelerometer with a gyroscope, machine learning methods' categorization performance can already be significantly improved [34].

As a result, contemporary scientific research is focused on integration algorithms for multimodal datasets, such as [35] and [36], as well as on constructing network designs capable of handling both forms of data input [37, 38]. Nonetheless, methods from computer graphics and computer vision and language processing are frequently used to inspire systems for recognizing human behavior obtained from the sensors data. This causes a divide between the two deep learning fields since state-of-the-art sensor-based HAR lags far behind computer vision.

However, the body of studies on these subjects has grown steadily over the years (for a summary, see [39] and [40]. However, the emphasis on critical principles seems to have shifted. These challenges include that certain studies are difficult to repeat due to the unavailability of open source software, a concentration on very narrow characteristics, and a failure to generalize significantly beyond the data set on which they were generated. Consequently, this paper provides a hands-on tutorial for HAR researchers on employing deep learning to real-world sensor-based HAR issues.

3 Recognizing Human Activity Using Deep Learning Approaches

This part introduces a framework for recognizing human activities based on deep learning techniques and wearable sensor data. The framework is named SDL-HAR and is based on Bulling et al.'s activity recognition chain (ARC) [22]. Figure 1 depicts the various steps of the SDL-HAR framework in this tutorial. The raw data is initially pre-processed and split into sliding windows utilizing a collection of inertial sensors implanted in intelligent wearables. In contrast to the ARC, feature extraction is deprecated in the SDL-HAR. Sensor data is sent into a deep learning network in its raw, pre-processed, and segmented state to generate a collection of predicted labels. The resultant collection of forecasted labels is then evaluated using state-of-the-art HAR criteria (i.e., Accuracy, Recall, Precision, and F1-score).

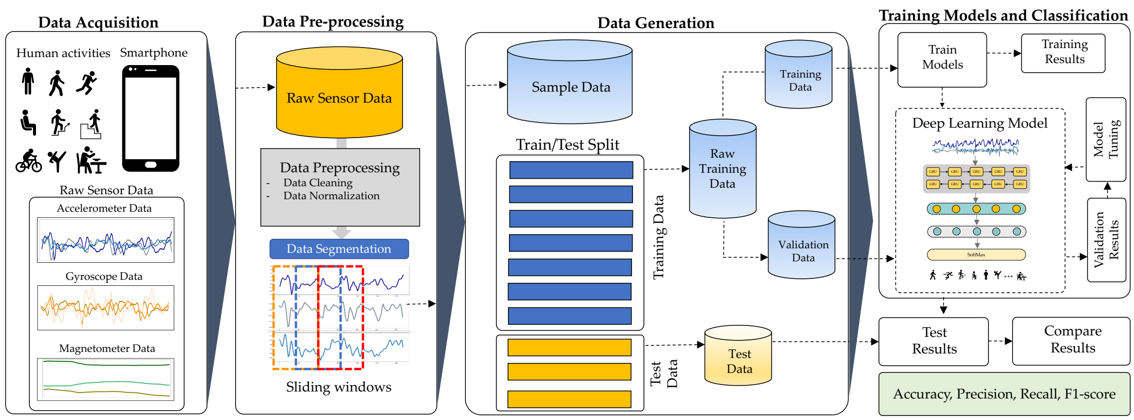


Figure 1: The SDL-HAR framework used in this tutorial

3.1 Data Acquisition

In the initial step of HAR, data acquisition describes the act of gathering and recording original input data collected by inertial sensors for HAR into a dataset. Sensor data are often acquired through a designed program that operates on the wearable device and periodically samples data from built-in wearable sensors.

Straczkiewicz et al. [41] discovered that the accelerometer, gyroscope, and magnetometer are the most often utilized sensors for HAR since they measure acceleration, angular velocity, and smartphone orientation, respectively. These sensors give high-resolution measurements that could differentiate between different types of activity. Table 1 contains information about the sensors.

Table 1: Three sensors often utilized in HAR in detail

Sensor	Details	Output	+/-
Accelerometer	Three orthogonal axes of the smartphone are measured for the rate of change in velocity	Gravitational units (g) or measure per seconds squared (m/s^2)	Depends on the orientation of smartphone
Gyroscope	Determines three orthogonal angles of smartphone angular velocity.	Radians per second (rad/s)	Depends on the direction of rotation
Magnetometer	Indicator of Planet's magnetic flux density relates to the phone's three orthogonal axes	Microtesla (mT)	Depends on the orientation of smartphone

For the purpose of analyzing sensor data, sampling frequency refers to the number of occurrences that a sensor collects during a one-second time frame. Typically, the sampling frequency is chosen to balance measurement precision and battery depletion. Inertial sensors were frequently sampled at a 20–30 Hz frequency in the studied research. The largest differences were shown in cases where power consumption was a concern (e.g., accelerometer samples at 1 Hz [42]) or when researchers utilized sophisticated signal processing techniques including time-frequency segmentation or activity patterns that need a higher sampling frequency (e.g., accelerometer sampled at 100 Hz [43]). According to detailed research, inertial sensors sampling at a rate of 20 Hz supplied appropriate data to differentiate between different modes of transportation [44]. In contrast, a sampling rate of 10 Hz sufficed to distinguish between multiple ways of mobility [45]. Reduce the sample rate from 100 Hz to 12.5 Hz, and the length of data gathering on a single battery charge increases by a factor of threefold [46].

The strategy could serve as a model for establishing confidentiality procedures that allow for future research replication when constructing a sensor-based HAR platform. An alternative is to utilize publically available datasets, as illustrated in Table 2.

Table 2: Details of the three most commonly used sensors in HAR

Dataset	Subject	Environment	Sensors	Body Location	Activities	References
UCI-HAR	30	Controlled	Acc. Gyro.	Lower body part	Posture, mobility	[47]
WISDM	36	Controlled	Acc.	Lower body part	Posture, mobility	[48]
UniMiB SHAR	30	Controlled	Acc.	Lower body part	Posture, mobility, fall	[49]
MobiAct	54	Controlled	Acc. Gyro. Mag.	Lower body part	Mobility, other	[50]
Complex	10	Controlled	Acc. Gyro. Mag.	Lower body part	Posture, mobility	[51]
Real-life HAR	19	Free-living	Acc. Gyro. Mag. GPS	NA/unconstrained	Mobility, locomotion, other	[52]
Transportation Mode Detection	13	Free-living	Acc. Gyro. Mag. Others	NA/unconstrained	Mobility, locomotion	[53]

Additionally, data exploration and visualization are processes used to decipher sensor data characteristics in each HAR dataset. Figure 2 displays some accelerometer data samples, while Figure 3 demonstrates some examples of gyroscope data obtained from smartphone sensors and gathered in the UCI-HAR dataset.

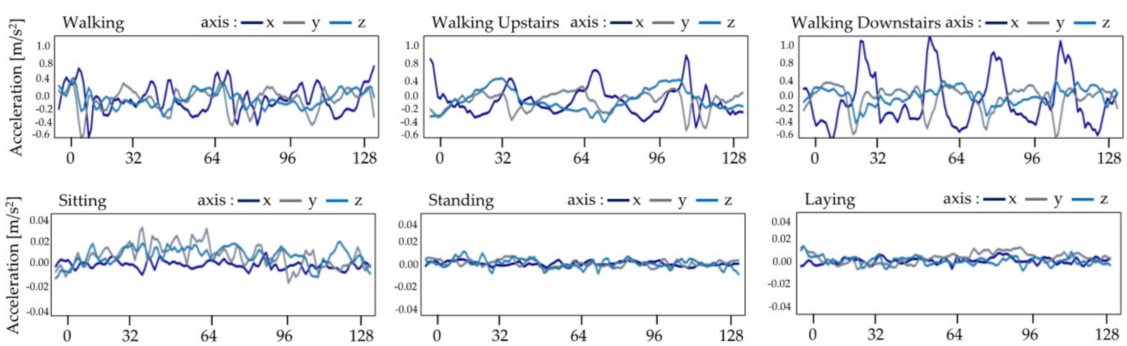


Figure 2: Accelerometer data samples of UCI-HAR dataset

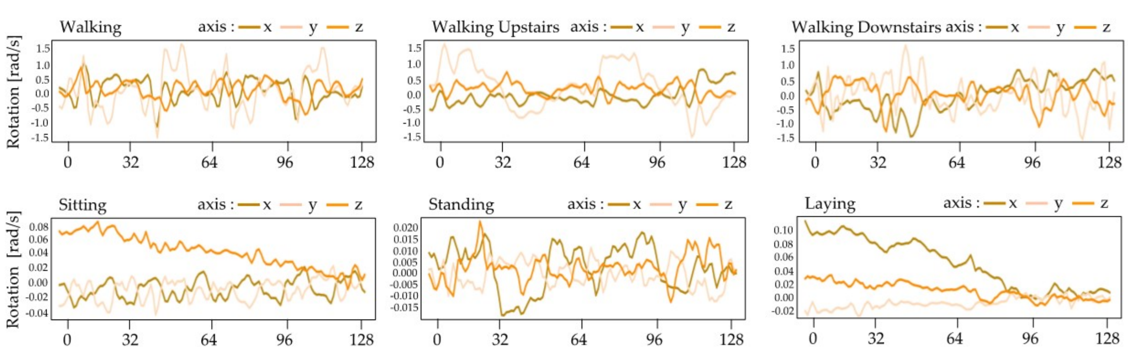


Figure 3: Gyroscope data samples of UCI-HAR dataset

3.2 Data Pre-processing

Data pre-processing is a set of methods used to restore, filter, and manipulate raw sensor data collected for HAR measurements. This phase is essential for three reasons: (1) performance measures integrated into smartphones are frequently less steady than research-grade data acquiring modules, and as a result, data could well be recorded inconsistently or contain incomplete data or outliers unrelated to a participant's behavioral intention; (2) the spatial perspective of the device (for example, how the mobile is placed in a participant's pocket) affects tri-axial readings of inertial sensors, possibly diminishing the HAR overall system performance;

Generally, the first category of impediments was handled utilizing signal processing methods. For example, researchers advocated using linear interpolation [54] or spline interpolation [55] to compensate for the required and operating sampling frequency discrepancy. These processes were applied to various impacted sensors, most notably the accelerometer, gyroscope, magnetometer, and barometer. Additional time-domain pre-processing addressed data reduction, which eliminates redundant data components. To accomplish this, the beginnings and finish of each action bout, a short time of movement of a particular type, were trimmed since they were regarded unrepresentative of the action [56]. The investigators also addressed dataset imbalance, which arises when the training dataset has differing amounts of occurrences for distinct activity groups. This circumstance predisposes the classifier to overfit on behalf of the more significant class; in the examined cases, this problem was overcome by increasing or decreasing the sample size [57, 58, 59]. Additionally, the data was analyzed to eliminate high-frequency noise (i.e., denoising). The literature review revealed some many methodologies are appropriate for this challenge, including through low-pass finite impulse response filters (typically with a cutoff frequency of 10 Hz for inertial sensors and 0.1 Hz for barometers) [60]. These approaches eliminate the fraction of the signal that is highly improbable to be caused by the actions of involvement; exponentially weighted moving [54]; moving median [61]; singular-value decomposition [62].

Some other aspects of data pre-processing take device alignment into account. Wearable sensor readings depend on device orientation, which is influenced by clothing, body form, and motion during dynamic activities [57]. One of the most often described options in the research is to convert the three-dimensional signal into a univariate vector magnitude that is rotational stable and more resilient to interpretation. Traditionally, this distinction was accomplished using a low-order Butterworth filter (e.g., order 3) with a cutoff frequency less than 1 Hz. This approach was often used with data from accelerometers, gyroscopes, and magnetometers. Accelerometer data were additionally filtered digitally to separate it into linear (body-related movements) and gravitational (related to spatial device awareness) acceleration [63]. Traditionally, this distinction was accomplished using a low-order Butterworth filter (e.g., order 3) with a cutoff frequency less than 1 Hz.

3.2.1 Data Segmentation

This step divides sensor data into manageable segments and estimates signal characteristics for each segment. In the analyzed publications, this segmentation was often accomplished by using a windowing approach that allowed for the overlap of subsequent windows. The window size was typically fixed at 1 to 5 seconds, with the overlap of subsequent windows frequently set to 50%, as seen in Figure 4. Numerous investigations on the appropriate window size corroborated this general finding: small windows (1–2 seconds) were adequate for identifying posture and motion. Still, somewhat longer windows (4–5 seconds) performed better in categorization [64, 65]. Even larger windows (10s or more) have been advocated for detecting locomotion patterns or for HAR solutions that use Fourier-domain characteristics (the resolution of the resultant frequency spectrum is inversely related to window length) [44]. In general, this adjustment intends to attain a close match between the size of the window and the time of a single occurrence of the action (e.g., one step). Researchers investigated more adaptable segmentation algorithms for a useful purpose. One possibility was to segment data according to designated time events such as zero-cross points, peak points, or valley points, which correspond to the beginning and finish of a specific motion bout [54, 57]. This enabled segments to be of varying lengths to coincide with a single essential action time. Generally, this method was employed to detect quasiperiodic movements such as walking, running, and stair climbing [62].

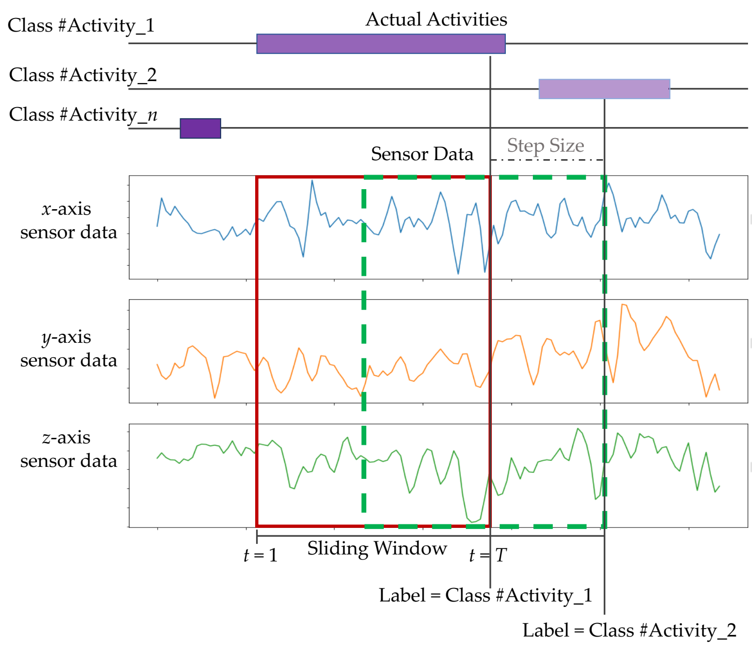


Figure 4: Data segmentation using a windowing technique with 50% overlapping proportion

3.3 Data Generation

Data generation is a phrase that describes the process of splitting a dataset into training and test data according to a predefined procedure. Commonly, the chosen approach is cross-validated, which divides the acquired dataset into two or more pieces and testing – and utilizes just the portion of the data not used in training for testing. Cross-validation processes are mentioned in the literature, with k-fold and leave-one-out cross-validation. This technique is becoming the most frequently used [66] as shown in Figure 5. The most often used train-test ratios were 90/10, 70/30, and 60/40. Validation is particularly beneficial if conducted utilizing surveys involving diverse demographics and smartphone use patterns. This technique enables one to comprehend the HAR system's generalizability to real-world settings and people. In [67], [68], and [69], we discovered a few investigations that used this validation strategy.

3.4 Model Training and Classification

A wide range of inference approaches using diverse strategies has lately been discovered in DL research on HAR. According to HAR researchers, deep neural networks surpass classical machine learning in terms of recognition performance. Classification methods capable of solving the HAR issue with high efficiency include CNNs and LSTMs. Depending on the procedure, the SDL-HAR framework uses either training or classification to analyze the given data further to build HAR classifiers.

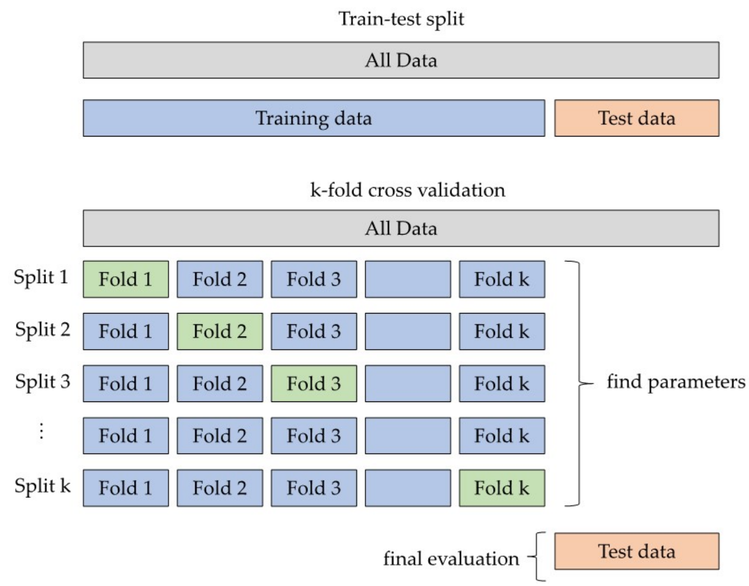


Figure 5: Data generation techniques mostly used in HAR

3.4.1 Training Process

It is necessary to train the models before using supervised inference techniques. Data $T = \{(X_i, y_i)\}_{i=1}^{N}$ is used for training, containing N pairs of feature vectors X_i and their matching ground truth labels y_i. To reduce the classification error on T, model parameters θ could be trained. Using expectation maximization [22], a separate model is trained for each class based on the training data T and an initial estimation of the parameter θ. Gradient descent is minimized through filtering procedures. Sensor-based HAR could benefit from deep learning approaches including CNNs and LSTMs.

1. CNN Model. NNs are multi-layered neural networks, and CNN is one such NN. CNN comprises two major components: a network of interconnected nodes and a series of convolution and sampling layers. Features are extracted from the convolution and sampling layers. A fully-connected network is added at the top to learn the weights of categorization. Convolutional, pooling, and fully connected are the three primary layers of CNNs. An explanation and an understanding of these tiers follow. Filters make up the convolutional layer of the algorithm. These filters' goal is to extract local features (the feature map) from the input data (sensory data). One filter is used for each feature map. The feature maps are created by sliding the matching filter over the input and calculating the dot product (convolution process). Only the receptive field, the same size as the filter, is connected to each neuron in the feature maps. In a single feature map, the weights of all neurons are the same. A benefit of sharing consequences is that the calculation is more effective since there are fewer parameters. It is also possible to recognize a specific pattern, regardless of where it appears in the inputs. The size of the feature map is mostly determined by the number of strides and the length of the filter. The architecture of CNN is illustrated in Figure 6.

2. LSTM Model. LSTM is now one of the fascinating subfields in DL. They have been utilized to exhibit global performance in various challenging issue areas, including language translation, image explanation, and text production. The LSTM network is a subclass of Recurrent Neural Networks (RNNs), a neural network intended to solve sequential issues. A RNN could be considered the addition of loops to an ordinary feed-forward MLP network. The memory cell, memory block, or simply cell is the computational element of the LSTM network. Because the word neuron as a computing unit is so established

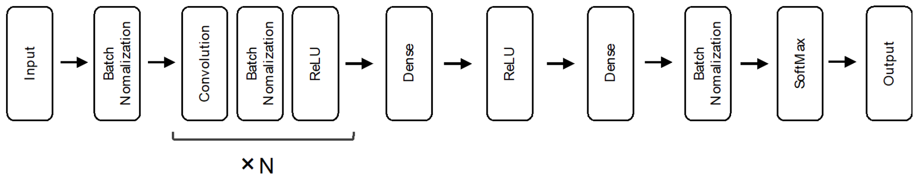


Figure 6: The CNN architecture

in discussions of MLPs, it is frequently referred to as the LSTM memory cell. Weights and gates are used to construct LSTM cells. The gates are critical to the memory cell's operation. These are weighted functions that further regulate the cell's data flow. Three gates are present:

- Forget Gate: Determines the data to reject from the cell.
- Input Gate: Determines the information in the input should be used to modify the memory state.
- Output Gate: Determines what to output depending on the input and the cell's memory.

LSTM cells, like conventional neurons, are grouped in layers, as seen in Figure 7, with the output of each cell being sent to the subsequent cell in the layer and the network's following layer. The production of the final layer could be placed into dense and softmax layers for classification.

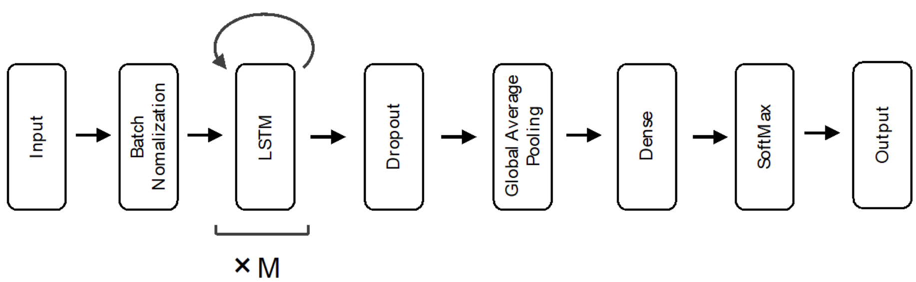


Figure 7: The architecture of LSTM

3. CNN-LSTM Model. A CNN model could be used in conjunction with an LSTM backend to understand subsequences of input that are presented as a sequence to an LSTM model. This hybrid model is referred to as a CNN-LSTM model. The CNN-LSTM design employs CNN layers to extract features from input data and LSTM layers to provide sequencing forecasting. The CNN effectively removes relevant and learning features from univariate time series data. The CNN-LSTM model will receive subsequences of the main sequence as a block, extract features from each block, and then interpret the retrieved features using the LSTM. The CNN-LSTM model's design is depicted in Figure 8.

3.4.2 Classification

The process of categorization consists of two separate phases. In the first stage, each feature vector X_i is mapped to a set of class labels $Y = \{y_1, \ldots, y_c\}$ with accompanying scores using a trained model with attributes $P_i = \{p_i^1, \ldots, p_i^c\}$ (or confidence values) with the inference method I.

$$p_i(y|X_i, \theta) = I(X_i, \theta), \quad for\, y \in Y \tag{1}$$

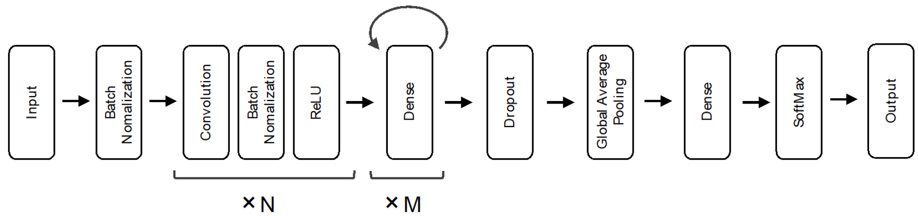


Figure 8: The architecture of CNN-LSTM

The scores correlate to probabilities in Bayesian techniques such as dynamic Bayesian networks or naïve Bayes classifiers. Numerous non-Bayesian classifiers could be calibrated to produce probabilistic outputs comparable to those generated by Bayesian classifiers [70]. In a subsequent stage, the computed P_i scores could be utilized in various ways. One of the most often used methods is to calculate the most significant score and output the matching class label y_i:

$$y_i = \arg\max\ p(y|X_i, \theta) \tag{2}$$

4 Case Study

This tutorial will use the WISDM dataset [48] to demonstrate how to identify human behaviors using accelerometer and gyroscope data from wearable sensors. The Wireless Sensor Data Mining (WISDM) laboratory acquired the dataset in 2010 from 36 individuals under supervised laboratory circumstances. The dataset contains 1,098,207 labeled samples representing six different activities: walking (38.6%), jogging (31.2%), ascending (11.2%), descending (9.1%), sitting (5.5%), and standing (4.4%). The actions were captured at a sampling rate of 20 Hz (one sample every 50 ms). The sensor data was analyzed using the SDL-HAR framework. Numpy, Pandas, and Sci-kit Learn libraries have been used to acquire, research, and generate data. To study the effectiveness of deep learning methodologies, we developed two standard CNN and LSTM models using the TensorFlow framework.

4.1 Software Configuration

The Tesla V100-SXM2-16GB graphics processor module was employed to enhance deep learning model training. This tutorial walked through implementing it on the Google Colab Pro platform. The deep learning models are implemented using Python (v.3.9.1) and CUDA (v.8.0.6). These investigations make use of the following Python libraries:

- When retrieving, processing, and interpreting sensor data, Numpy and Pandas are utilized for data management.
- For charting and viewing the outcomes of data exploration and model assessment, Matplotlib and Seaborn are utilized.
- Scikit-learn (Sklearn) is a package for data collection and generation in experiments.
- Deep learning models are implemented and trained using TensorFlow and Keras.

4.2 Experiments and Results

4.2.1 Experiments

The experiments are diversified to measure the effectiveness of each deep learning network and to solve the HAR concern. The first version employs a standard deep learning network known as the CNN network. The second variant compares the CNN network to a recurrent neural network called LSTM. The third variant is a CNN-LSTM hybrid deep learning model, which integrates an LSTM network with a convolutional layer. Tables 3, 4, and 5 summarize the deep learning networks utilized in this tutorial.

Table 3: The summary of Hyperparameters for a CNN network used in this tutorial

Stage	**Hyperparameters**		**Values**
Architecture	Convolution-1	Kernel Size	3
		Stride	1
		Filters	16
	Dropout-1		0.25
	Maxpooling		2
	Dropout-2		0.25
	Dense		256
Training	Loss Function		Cross-entropy
	Optimizer		Adam
	Batch Size		64
	Learning Rate		0.01
	Number of Epoches		100

Table 4: The summary of Hyperparameters for a LSTM network used in this tutorial

Stage	**Hyperparameters**	**Values**
Architecture	LSTM-neuron	128
	Dropout	0.25
	Dense	256
Training	Loss Function	Cross-entropy
	Optimizer	Adam
	Batch Size	64
	Learning Rate	0.001
	Number of Epoches	100

4.2.2 Results

Table 6 summarizes the experimental findings of deep learning models trained on wearable sensor data from the WISDM dataset. The experimental findings demonstrate that the hybrid deep learning model, which consists of CNN and LSTM, surpasses other standard CNN and LSTM models with a maximum accuracy of 94.76%. To conduct a thorough analysis of these achievements, we could examine the confusion matrices of each deep learning study. Figure 9 illustrates the confusion matrices. The confusion matrices demonstrate that the CNN-LSTM model performs effectively in all activities included in the WISDM dataset. Figure 10 illustrates the comparative outcomes.

Table 5: The summary of Hyperparameters for a LSTM network used in this tutorial

Stage	Hyperparameters		Values
Architecture	Convolution-1	Kernel Size	3
		Stride	1
		Filters	16
	Convolution-2	Kernel Size	5
		Stride	1
		Filters	32
	Dropout-1		0.25
	Maxpooling		2
	LSTM-neuron		128
	Dropout-2		0.25
	Dense		256
Training	Loss Function		Cross-entropy
	Optimizer		Adam
	Batch Size		64
	Learning Rate		0.01
	Number of Epoches		100

Table 6: Average metrics on classifier evaluation of deep learning models using WISDM dataset

Model	Accuracy	Precision	Recall	F1-Score
CNN	93.313%. (+/- 0.375%)	86.957%. (+/- 1.651%)	87.030%. (+/- 0.509%)	86.988%. (+/- 0.981%)
LSTM	92.765%. (+/- 1.265%)	92.888% (+/- 1.289%)	92.765% (+/- 1.265%)	92.807%. (+/- 1.282%)
CNN-LSTM	94.769%. (+/- 0.952%)	94.626% (+/- 1.006%)	94.769%. (+/- 0.952%)	94.592% (+/- 1.030%)

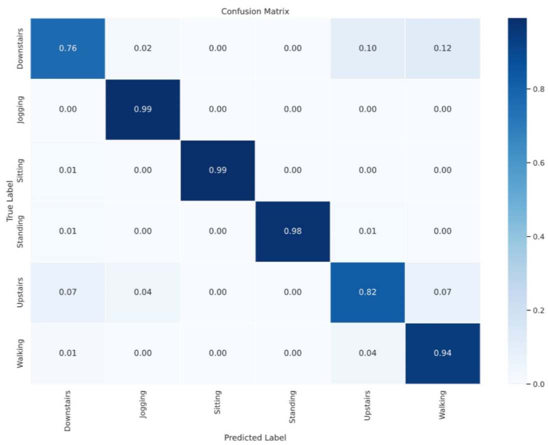
5 Conclusions and Challenging Research

This tutorial is intended for inexperienced with deep learning-based HAR using intelligent wearable sensors. We started by explaining the significant research issues confronting researchers studying human activity recognition. We then discussed activity recognition procedures in-depth as a general-purpose framework for data processing, model creation, and evaluation. We reviewed recommended practice methodologies established by the activity recognition research community. To illustrate the HAR platform in action, including recognizing several everyday activities using inertial sensors integrated with a smartphone. The example's purposeful simplicity enabled us to evaluate several deep learning models in terms of overall action recognition, which we expect will be helpful to beginners when constructing more sophisticated activity detection systems.

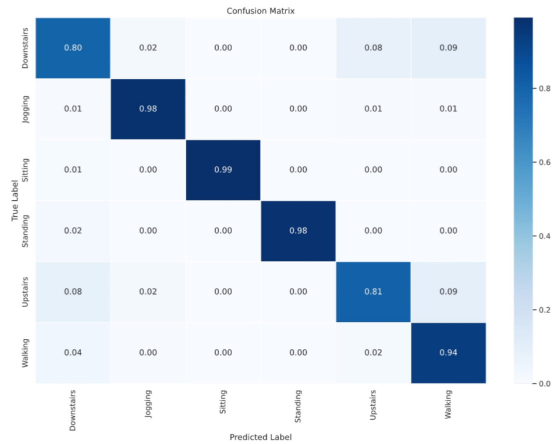
This article provides a tutorial for scholars interested in sensor-based HAR on how to use deep learning to real sensor-based HAR challenges. Leveraging HAR using wearable sensor data as a case study, this tutorial explains data preparation, data production, deep learning model design, training, and testing. The authors believe that this article presents readers with valuable first-hand expertise in developing their deep learning models for their research.

While human activity recognition has many of the same study issues as the broader subject of pattern recognition, it also confronts certain distinct challenges.

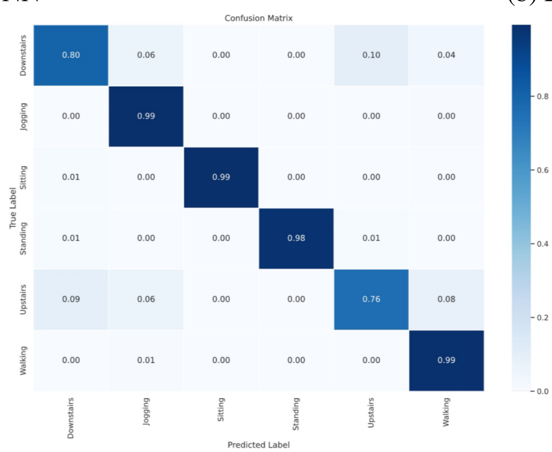
- Deep learning approaches use fixed sliding window techniques to respond to data segmentation. A static time window could be either too broad, collecting more than is required to identify specific



(a) CNN



(b) LSTM



(c) CNN-LSTM

Figure 9: Confusion matrices of (a) CNN, (b) LSTM, and (c) CNN-LSTM computed in the experiment

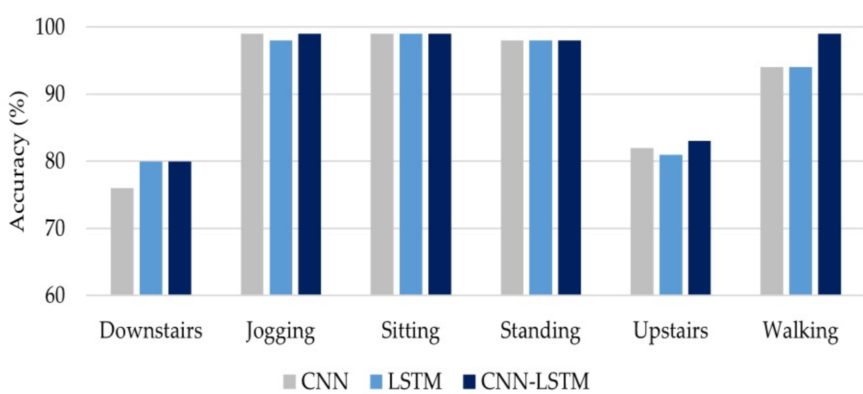


Figure 10: Comparative results of deep learning models for each activity in WISDM dataset

actions, or insufficiently narrow, collecting insufficient of the series to identify lengthy behavior and actions. Currently, academics have focused on optimizing the segmentation of time series data. Additional testing and study on variable activity segments and approaches that include both short-term and long-term properties (e.g., wavelets) are required to develop reliable models across all time frames.

- Modern HAR techniques reach a good performance for basic activities such as jogging. Nevertheless, complicated tasks like eating, which could include a variety of distinct motions, continue to offer obstacles. While hierarchical approaches have been established to address this uncertainty, there is still potential for enhancement.

- When a model works well on data it has never seen before, it has a significant level of generalizability. When a model works well on training data but adversely on new data, it also seems to overfit. K-fold cross-validation or leave-one-participant-out cross-validation is commonly employed to increase the model's generalizability. Numerous variables will impact the model's generalizability in wearable-based HAR. As a result, integrating incremental learning and deep learning into wearable HAR systems remains an ongoing challenge.

References

[1] Sakorn Mekruksavanich, Anuchit Jitpattanakul, Phichai Youplao, and Preecha P. Yupapin. Enhanced hand-oriented activity recognition based on smartwatch sensor data using lstms. *Symmetry*, 12(9):1570, 2020.

[2] Sakorn Mekruksavanich and Anuchit Jitpattanakul. Biometric user identification based on human activity recognition using wearable sensors: An experiment using deep learning models. *Electronics*, 10(3), 2021.

[3] Farman Ullah, Asif Iqbal, Sumbul Iqbal, Daehan Kwak, Hafeez Anwar, Ajmal Khan, Rehmat Ullah, Huma Siddique, and Kyung-Sup Kwak. A framework for maternal physical activities and health monitoring using wearable sensors. *Sensors*, 21(15), 2021.

[4] Deepak Kumar, Chaman Verma, Sanjay Dahiya, Pradeep Kumar Singh, Maria Simona Raboaca, Zoltán Illés, and Brijesh Bakariya. Cardiac diagnostic feature and demographic identification (cdf-di): An iot enabled healthcare framework using machine learning. *Sensors*, 21(19), 2021.

[5] Anitha S Prasad and N Kavanashree. Ecg monitoring system using ad8232 sensor. In *2019 International Conference on Communication and Electronics Systems (ICCES)*, pages 976–980, 2019.

[6] Volkan Senyurek, Masudul Imtiaz, Prajakta Belsare, Stephen Tiffany, and Edward Sazonov. Cigarette smoking detection with an inertial sensor and a smart lighter. *Sensors*, 19(3), 2019.

[7] Sreenivasan Ramasamy Ramamurthy and Nirmalya Roy. Recent trends in machine learning for human activity recognition-a survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8:e1254, 03 2018.

[8] B. Almaslukh, Jalal Muhtadi, and Abdel Artoli. A robust convolutional neural network for online smartphone-based human activity recognition. *Journal of Intelligent & Fuzzy Systems*, 35:1–12, 06 2018.

[9] Andrey Ignatov. Real-time human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing*, 62:915–922, 2018.

[10] Song-Mi Lee, Sang Min Yoon, and Heeryon Cho. Human activity recognition from accelerometer data using convolutional neural network. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 131–134, 2017.

[11] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119, 07 2017.

[12] Alejandro Baldominos, Alejandro Cervantes, Yago Saez, and Pedro Isasi. A comparison of machine learning and deep learning techniques for activity recognition using mobile devices. *Sensors*, 19(3), 2019.

[13] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1), 2016.

[14] Koyu Hori, Yufeng Mao, Yumi Ono, Hiroki Ora, Yuki Hirobe, Hiroyuki Sawada, Akira Inaba, Satoshi Orimo, and Yoshihiro Miyake. Inertial measurement unit-based estimation of foot trajectory for clinical gait analysis. *Frontiers in Physiology*, 10:1530, 01 2020.

[15] Djordje Slijepcevic, Fabian Horst, Sebastian Lapuschkin, Anna-Maria Raberger, Matthias Zeppelzauer, Wojciech Samek, Christian Breiteneder, Wolfgang I. Schöllhorn, and Brian Horsak. On the explanation of machine learning predictions in clinical gait analysis, 2020.

[16] Rene Grzeszick, Jan Marius Lenk, Fernando Moya Rueda, Gernot A. Fink, Sascha Feldhorst, and Michael ten Hompel. Deep neural network based human activity recognition for the order picking process. In *Proceedings of the 4th International Workshop on Sensor-Based Activity Recognition and Interaction*, iWOAR '17, New York, NY, USA, 2017. Association for Computing Machinery.

[17] Heli Koskimaki, Ville Huikari, Pekka Siirtola, Perttu Laurinen, and Juha Roning. Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines. In *2009 17th Mediterranean Conference on Control and Automation*, pages 401–405, 2009.

[18] Ekaterina Gilman, Satu Tamminen, Rumana Yasmin, Eemeli Ristimella, Ella Peltonen, Markus Harju, Lauri Lovén, Jukka Riekki, and Susanna Pirttikangas. Internet of things for smart spaces: A university campus case study. *Sensors*, 20(13), 2020.

[19] Uwe Köckemann, Marjan Alirezaie, Jennifer Renoux, Nicolas Tsiftes, Mobyen Uddin Ahmed, Daniel Morberg, Maria Lindén, and Amy Loutfi. Open-source data collection and data sets for activity recognition in smart homes. *Sensors*, 20(3), 2020.

[20] Alexander Hölzemann and Kristof Van Laerhoven. Using wrist-worn activity recognition for basketball game analysis. In *Proceedings of the 5th International Workshop on Sensor-Based Activity Recognition and Interaction*, iWOAR '18, New York, NY, USA, 2018. Association for Computing Machinery.

[21] Chunyan Ma, Ji Fan, Jinghao Yao, and Tao Zhang. Npu rgbd dataset and a feature-enhanced lstm-dgcn method for action recognition of basketball players+. *Applied Sciences*, 11(10), 2021.

[22] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.*, 46(3), jan 2014.

[23] Jingcheng Chen, Yining Sun, and Shaoming Sun. Improving human activity recognition performance by data fusion and feature engineering. *Sensors*, 21(3), 2021.

[24] Xin Qin, Yiqiang Chen, Jindong Wang, and Chaohui Yu. Cross-dataset activity recognition via adaptive spatial-temporal transfer learning. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(4), dec 2019.

[25] Seyed Ali Rokni and Hassan Ghasemzadeh. Autonomous training of activity recognition algorithms in mobile sensors: A transfer learning approach in context-invariant views. *IEEE Transactions on Mobile Computing*, 17(8):1764–1777, 2018.

[26] Elena-Alexandra Budisteanu and Irina Georgiana Mocanu. Combining supervised and unsupervised learning algorithms for human activity recognition. *Sensors*, 21(18), 2021.

[27] Evaggelos Spyrou, Eirini Mathe, Georgios Pikramenos, Konstantinos Kechagias, and Phivos Mylonas. Data augmentation vs. domain adaptation—a case study in human activity recognition. *Technologies*, 8(4), 2020.

[28] Clayton Frederick Souza Leite and Yu Xiao. Improving cross-subject activity recognition via adversarial learning. *IEEE Access*, 8:90542–90554, 2020.

[29] Sakorn Mekruksavanich and Anuchit Jitpattanakul. LSTM networks using smartphone data for sensor-based human activity recognition in smart homes. *Sensors*, 21(5):1636, 2021.

[30] Sakorn Mekruksavanich and Anuchit Jitpattanakul. Deep convolutional neural network with rnns for complex activity recognition using wrist-worn wearable sensor data. *Electronics*, 10(14), 2021.

[31] Sakorn Mekruksavanich and Anuchit Jitpattanakul. Deep learning approaches for continuous authentication based on activity patterns using mobile sensing. *Sensors*, 21(22), 2021.

[32] Sakorn Mekruksavanich and Anuchit Jitpattanakul. Convolutional neural network and data augmentation for behavioral-based biometric user identification. In Milan Tuba, Shyam Akashe, and Amit Joshi, editors, *ICT Systems and Sustainability*, pages 753–761, Singapore, 2021. Springer Singapore.

[33] Yaser Souri, Mohsen Fayyaz, Luca Minciullo, Gianpiero Francesca, and Juergen Gall. Fast weakly supervised action segmentation using mutual consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[34] Haodong Guo, Ling Chen, Liangying Peng, and Gencai Chen. Wearable sensor based multimodal human activity recognition exploiting the diversity of classifier ensemble. UbiComp '16, page 1112–1123, New York, NY, USA, 2016. Association for Computing Machinery.

[35] James Male and Uriel Martinez-Hernandez. Recognition of human activity and the state of an assembly task using vision and inertial sensor fusion methods. In *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, volume 1, pages 919–924, 2021.

[36] Sebastian Münzner, Philip Schmidt, Attila Reiss, Michael Hanselmann, Rainer Stiefelhagen, and Robert Dürichen. Cnn-based sensor fusion techniques for multimodal human activity recognition. ISWC '17, page 158–165, New York, NY, USA, 2017. Association for Computing Machinery.

[37] Roberto Henschel, Timo Von Marcard, and Bodo Rosenhahn. Accurate long-term multiple people tracking using video and body-worn imus. *IEEE Transactions on Image Processing*, 29:8476–8489, 2020.

[38] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *Proce. of the 2017 British Machine Vision Conference (BMVC'17), London, UK*, pages 14:1–14:13. BMVA Press, September 2017.

[39] Netzahualcoyotl Hernandez, Jens Lundström, Jesus Favela, Ian McChesney, and Bert Arnrich. Literature review on transfer learning for human activity recognition using mobile and wearable devices with environmental technology. *SN Computer Science*, 1, 02 2020.

[40] Alen Rajšp and Iztok Fister. A systematic literature review of intelligent data analysis methods for smart sport training. *Applied Sciences*, 10(9), 2020.

[41] Marcin Straczkiewicz, Peter James, and Jukka-Pekka Onnela. A systematic review of smartphone-based human activity recognition methods for health research. *npj Digital Medicine*, 4(1), oct 2021.

[42] Dongyoun Shin, Daniel Aliaga, Bige Tunçer, Stefan Müller Arisona, Sungah Kim, Dani Zünd, and Gerhard Schmitt. Urban sensing: Using smartphones for transportation mode classification. *Computers, Environment and Urban Systems*, 53:76–86, 2015. Special Issue on Volunteered Geographic Information.

[43] Taeho Hur, Jaehun Bang, Thien Huynh-The, Jongwon Lee, Jee-In Kim, and Sungyoung Lee. Iss2image: A novel signal-encoding technique for cnn-based human activity recognition. *Sensors*, 18(11), 2018.

[44] Martin Gjoreski, Vito Janko, Gašper Slapničar, Miha Mlakar, Nina Reščič, Jani Bizjak, Vid Drobnič, Matej Marinko, Nejc Mlakar, Mitja Lustrek, and Matjaz Gams. Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors. *Information Fusion*, 62, 04 2020.

[45] Johan Wannenburg and Reza Malekian. Physical activity recognition from smartphone accelerometer data for user context awareness sensing. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(12):3142–3149, 2017.

[46] Ozgur Yurur, Miguel Labrador, and Wilfrido Moreno. Adaptive and energy efficient context representation framework in mobile sensing. *IEEE Transactions on Mobile Computing*, 13(8):1681–1693, 2014.

[47] D. Anguita, Alessandro Ghio, L. Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *ESANN*, 2013.

[48] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82, mar 2011.

[49] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10), 2017.

[50] George Vavoulas, Charikleia Chatzaki, Thodoris Malliotakis, Matthew Pediaditis, and Manolis Tsiknakis. The mobiact dataset: Recognition of activities of daily living using smartphones. pages 143–151, 01 2016.

[51] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul J. M. Havinga. Complex human activity recognition using smartphone and wrist-worn motion sensors. *Sensors*, 16(4), 2016.

[52] Daniel Garcia-Gonzalez, Daniel Rivero, Enrique Fernandez-Blanco, and Miguel R. Luaces. A public domain dataset for real-life human activity recognition using smartphone sensors. *Sensors*, 20(8), 2020.

[53] Claudia Carpineti, Vincenzo Lomonaco, Luca Bedogni, Marco Felice, and Luciano Bononi. Custom dual transportation mode detection by smartphone devices exploiting sensor diversity. pages 367–372, 03 2018.

[54] Mohammad Derawi and Patrick Bours. Gait and activity recognition using commercial phones. *Computers & Security*, 39:137–144, 2013.

[55] Fuqiang Gu, Kourosh Khoshelham, Shahrokh Valaee, Jianga Shang, and Rui Zhang. Locomotion activity recognition using stacked denoising autoencoders. *IEEE Internet of Things Journal*, 5(3):2085–2093, 2018.

[56] Muhammad Arshad Awan, Zheng Guangbin, Cheong Ghil Kim, and Shin Dug Kim. Human activity recogni-

tion in wsn: A comparative study. *International Journal of Networked and Distributed Computing*, 2(4):221–230, November 2014.

[57] Yufei Chen and Chao Shen. Performance analysis of smartphone-sensor behavior for human activity recognition. *IEEE Access*, 5:3095–3110, 2017.

[58] Abdul Rehman Javed, Muhammad Usman Sarwar, Suleman Khan, Celestine Iwendi, Mohit Mittal, and Neeraj Kumar. Analyzing the effectiveness and contribution of each axis of tri-axial accelerometer sensor for accurate activity recognition. *Sensors*, 20(8), 2020.

[59] Debadyuti Mukherjee, Riktim Mondal, Pawan Singh, Ram Sarkar, and Debotosh Bhattacharjee. Ensemconvnet: a deep learning approach for human activity recognition using smartphone sensors for healthcare applications. *Multimedia Tools and Applications*, 79:1–28, 11 2020.

[60] Carlos Avilés-Cruz, Andrés Ferreyra-Ramírez, Arturo Zúñiga-López, and Juan Villegas-Cortéz. Coarse-fine convolutional deep-learning strategy for human activity recognition. *Sensors*, 19(7), 2019.

[61] Jayita Saha, Chandreyee Chowdhury, Dip Ghosh, and Sanghamitra Bandyopadhyay. A detailed human activity transition recognition framework for grossly labeled data from smartphone accelerometer. *Multimedia Tools and Applications*, 80:1–22, 03 2021.

[62] Andrey D. Ignatov and Vadim V. Strijov. Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer. *Multimedia Tools Appl.*, 75(12):7257–7270, jun 2016.

[63] Muhammad Arif, Mohsin Bilal, Ahmed Kattan, and Sheikh Ahamed. Better physical activity classification using smartphone acceleration sensor. *Journal of medical systems*, 38:95, 09 2014.

[64] Sulaimon Bashir, Daniel Doolan, and Andrei Petrovski. The effect of window length on accuracy of smartphone-based activity recognition. *IAENG International Journal of Computer Science*, 43:126–136, 02 2016.

[65] Dang-Nhac Lu, Duc-Nhan Nguyen, Thi-Hau Nguyen, and Ha-Nam Nguyen. Vehicle mode and driving activity detection based on analyzing sensor data of smartphones. *Sensors*, 18(4), 2018.

[66] Abayomi Moradeyo Otebolaku and Maria Teresa Andrade. User context recognition using smartphone sensors and classification models. *Journal of Network and Computer Applications*, 66:33–51, 2016.

[67] Michael Del Rosario, Kejia Wang, Jingjing Wang, Ying Liu, Matthew Brodie, Kim Delbaere, Nigel Lovell, and Stephen Lord. A comparison of activity classification in younger and older cohorts using a smartphone. *Physiological Measurement*, 35:2269, 10 2014.

[68] Mark Albert, Santiago Toledo, Mark Shapiro, and Konrad Kording. Using mobile phones for activity recognition in parkinson's patients. *Frontiers in neurology*, 3:158, 11 2012.

[69] Adil Mehmood Khan, Muhammad Hameed Siddiqi, and Seok-Won Lee. Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones. *Sensors*, 13(10):13099–13122, 2013.

[70] Ira Cohen and Moises Goldszmidt. Properties and benefits of calibrated classifiers. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *Knowledge Discovery in Databases: PKDD 2004*, pages 125–136, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

Author Biography

Sakorn Mekruksavanich received the Ph.D. degree in Computer Engineering from the Chulalongkorn University in 2012. He also holds a M.S. in Computer Science from King Mongkut's Institute of Technology Ladkrabang in 2004 and a B.Eng. in Computer Engineering from Chiang Mai University in 1999. He is currently an assistant professor in the Department of Computer Engineering, School of Information and Communication Technology at University of Phayao, Phayao, Thailand. His current research interests are deep and machine learning, applying ML techniques in software engineering field, human activity recognition, and wearable sensors.

Anuchit Jitpattanakul received the B.Sc. degree in applied mathematics from King Mongkut's Institute of Technology North Bangkok, the M.Sc. degree in computational science and Ph.D. degree in computer engineering from Chulalongkorn University. He joined the Intelligent and Nonlinear Dynamic Innovations (INDI) Research Center, KMUTNB. He is currently a Faculty Member of the Department of Mathematics, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand. His current research interests include deep learning approaches applied to human activity recognition, wearable sensors, and healthcare applications.

A Python code for the case study

To assist in disseminating this research, we provide the Python code for the case study. Numpy, Pandas, and Sci-kit Learn packages analyze the information. TensorFlow is used to design, train, and test deep learning models.

```
#Data Reading
import pandas as pd
import numpy as np
file = open(filename)
lines = file.readlines()
processedList = []
for i, line in enumerate(lines):
        try:
        line = line.split(',')
                last = line[5].split(';')[0]
                last = last.strip()
                if last == '':
                        break;
                temp = [line[0], line[1], line[2],
                        line[3], line[4], last]
                processedList.append(temp)
        except:
                print('Error at line number: ', i)

data = pd.DataFrame(processedList, columns = columns)
```

```
#Data Preprocessing
import scipy.stats as stats
def data_segment(df, frame_size, step_size):
        N_FEATURES = 3
        frames = []
        labels = []
        for i in range(0, len(df) - frame_size, step_size):
                x = df['x'].values[i: i + frame_size]
                y = df['y'].values[i: i + frame_size]
                z = df['z'].values[i: i + frame_size]

                # Retrive the most often used Label in this segment
                label = stats.mode(df['label'][i: i+frame_size])[0][0]
                frames.append([x, y, z])
                labels.append(label)
        # Bring the segments into a better shape
        frames = np.asarray(frames).reshape(-1, frame_size,
                N_FEATURES)
        labels = np.asarray(labels)

        return frames, labels

X, y = data_segment(scaled_X, frame_size, step_size)

#Train_test_split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
        test_size = 0.2, random_state = 0, stratify = y)

#Model Building
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Flatten, Dense, Dropout
from tensorflow.keras.layers import Conv1D, MaxPool1D
from tensorflow.keras.optimizers import Adam

model = Sequential()
model.add(Conv1D(16, 3, activation='relu', input_shape = X_train[0].shape))
model.add(Dropout(0.1))
model.add(Conv1D(32, 5, activation = 'relu'))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(64, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(6, activation = 'softmax'))
model.compile(optimizer=Adam(learning_rate = 0.001),
        loss = 'sparse_categorical_crossentropy', metrics = ['accuracy'])
```

```
model.fit(X_train, y_train, epochs = 10,
        validation_data = (X_test, y_test), verbose=1)


#Model Evaluation
from sklearn.metrics import accuracy_score, precision_score,
        recall_score, f1_score, confusion_matrix

print(confusion_matrix(y_test, y_pred))
accuracy = accuracy_score(y_test, y_pred, average="weighted")
precision = precision_score(y_test, y_pred, average="weighted")
recall = recall_score(y_test, y_pred, average="weighted")
f1 = f1_score(y_test, y_pred, average="weighted")
```