

DL-HAR: Deep Learning-Based Human Activity Recognition Framework for Edge Computing

Abdu Gumaiei^{1,2,*}, Mabrook Al-Rakhami^{1,2}, Hussain AlSalman², Sk. Md. Mizanur Rahman³ and Atif Alamri^{1,2}

Abstract: Human activity recognition is commonly used in several Internet of Things applications to recognize different contexts and respond to them. Deep learning has gained momentum for identifying activities through sensors, smartphones or even surveillance cameras. However, it is often difficult to train deep learning models on constrained IoT devices. The focus of this paper is to propose an alternative model by constructing a Deep Learning-based Human Activity Recognition framework for edge computing, which we call DL-HAR. The goal of this framework is to exploit the capabilities of cloud computing to train a deep learning model and deploy it on less-powerful edge devices for recognition. The idea is to conduct the training of the model in the Cloud and distribute it to the edge nodes. We demonstrate how the DL-HAR can perform human activity recognition at the edge while improving efficiency and accuracy. In order to evaluate the proposed framework, we conducted a comprehensive set of experiments to validate the applicability of DL-HAR. Experimental results on the benchmark dataset show a significant increase in performance compared with the state-of-the-art models.

Keywords: Human activity recognition, edge computing, deep neural network, recurrent neural network, Docker.

1 Introduction

The innovative Internet of Things (IoT) paradigm has already started to alter the environment, making it much smarter in recent years [Liu, Qiu, Zhang et al. (2020)]. Actuation, storage, computation, and sensing resources have seen a large growth in availability as IoT device numbers have increased [H De La Iglesia, Villarrubia González, Sales Mendes et al. (2019)]. We are also aided in this by the widespread availability of devices for general purpose that are more affordable, such as Raspberry Pi; this will help

¹ Research Chair of Pervasive and Mobile Computing, King Saud University, Riyadh, 11543, Saudi Arabia.

² College of Computer and Information Sciences, King Saud University, Riyadh, 11543, Saudi Arabia.

³ Information and Communication Engineering Technology, School of Engineering Technology and Applied Science, Centennial College, Toronto, Canada.

* Corresponding Author: Abdu Gumaiei. Email: abduGumaiei@gmail.com.

Received: 27 May 2020; Accepted: 03 July 2020.

to improve the ease of IoT's adoption potential in a number of situations [Trihinas, Pallis and Dikaiakos (2018)]. In this regard, developers have also given lightweight virtualization technology more attention in recent years, which has led to developments such as Linux containers (LXC) and Docker containers [Merelli, Fornari, Tordini et al. (2019)]. As with the case of hypervisor-based virtualization technology, virtualized services have benefited from these new solutions, which bring increased efficiency and reduced costs. There have also been many examples of the investigation and use of containers in IoT devices that are resource-restricted [Morabito (2017)].

Recently, the use of the edge-computing framework in executing human activity recognition models at the edge of the network is still at its infancy stage [Greco, Ritrovato and Xhafa (2019)]. Research has shown that there is need to process sensor data generated from wearable sensors in a quick and efficient way [Sahinovic, Dzebo, Ustundag et al. (2018)]. Consequently, this affects how decisions can be reached based on the data analysis for application in various areas, such areas of data application include rehabilitation [Sahinovic, Dzebo, Ustundag et al. (2018)], healthcare [Al-Rakhami, Gumaei, Alsahli et al. (2020); Li, Huang, Li et al. (2019)], human surveillance [Park, Min and Lee (2017)], education [Fernández-Caramés and Fraga-Lamas (2019)] and industrial sectors [Uzunovic, Golubovic, Tucakovic et al. (2018)], etc. Systems for edge computing have seen major improvements with deep learning and machine learning algorithms, offering intelligent solutions for IoT applications used in quasi-real-time and real-time ways [Li, Ota and Dong (2018)]. New applications for deep learning techniques have become increasingly possible due to a number of reasons, such as a better computing ability and higher volume of data [Jan, Farman, Khan et al. (2019)]. In a typical scenario, a deep learning model is made of an ensemble of layers, each with an ability to scale down data with the output of the model yielding the desired results. With that in mind, a deep learning model is more suited to edge computing environments, as some layers in the edge can be offloaded then the reduced intermediary data sent to a cloud server [Esteva, Robicquet, Ramsundar et al. (2019)]. The other advantage is that this type of computing approach is more secure when transferring intermediate data. Deep learning is also quite useful when time-series data is large enough. However, if the length of this data is very long, it cannot reflect past information well. Beyond a certain limit, the information will not be analyzed with great accuracy levels. To overcome this, an artificial recurrent neural network (RNN) known as long short-term memory (LSTM) is used, which has a cell, an input gate, an output gate and a forget gate. This network can be used to control the use of cell state information [Mishra, Tripathi, Gupta et al. (2019)].

The growing number of powerful and smart devices along with and the centralized nature of cloud computing approach present numerous challenges and problems such as timely services, bandwidth constraints, connectivity, and security issues. In fact, the need for different IoT applications to operate at several locations such as smart homes, hospitals, and schools is also emerged [Samarah, Zamil, Rawashdeh et al. (2018)]. This is the reason behind the extensive research interest in the area, as scientists work to find solutions for problems such as jitter effects and latency, the location and distance of servers, applications' location awareness, the security of personal data and privacy, mobility support, etc. The main challenge is that this leans towards creating an inconsistency in the solution architecture, which has a repercussion in new markets such as in robotics, virtual reality,

modern e-health care, and automation [Hammerla, Halloran and Ploetz (2016); Ordóñez and Roggen (2016); Ravi, Wong, Lo et al. (2016); Ronao and Cho (2016)].

The models of application will create huge demand when it comes to dealing with high data transfer rates within a significantly short time [Al-Rakhami, Alsahli, Hassan et al. (2018)]. The emerging network architectures are thus faced with the challenge of adapting to these needs, with most of them unable to handle terabytes of information transfer churned from data sources in a few seconds. Obviously, data technologies are evolving quickly; thus, the data rates are bound to grow faster with time. Moreover, proper network designs can also help in improving the performance of emerging architectures [Yin, Zeng, Chen et al. (2016)].

This work looks into the possibilities of using edge computing coupled with artificial intelligence in vertical specific cases of basic human activity recognition. This work is particularly focused on developing a solution that uses an economical, light and efficient intelligent edge application to facilitate swift decision making and communication.

The main contributions of this paper are as follows:

- We propose a new deep learning-based human activity recognition framework for edge computing, called DL-HAR. The proposed framework aims to increase the speed of decision making. It uses a deep learning algorithm to reduce communication to the Cloud servers, thus reducing delays and round trips that may be unnecessary;
- We also conduct a comprehensive exploration and examination of the potential of edge computing distributed framework utilization as a lightweight and effective virtualization for the recognition of human activity;
- To evaluate the potential of the proposed framework, we examine the accuracy, F1-score, and recognition time performance measures during the experiment and compare them with shallow learning and the state-of-the-art models.

Our paper is organized as follows: Section 2 covers the related work from three perspectives, which are the pervasive sensing systems for HAR, algorithms for the recognition of activity from machine learning, and the edge computing approach. Section 3 introduces the DL-HAR framework and describe its components. Section 4 provides more testbed implementation details and a report of the experimental results. Section 5 covers conclusions and future work.

2 Related work

In this section, we cover three varied perspectives of work completed on edge computing and the recognition of human activity over recent years. The perspectives include the recognition of human activity via pervasive sensor systems, algorithms for the recognition of activity from machine learning, and approaches used in edge computing.

2.1 Pervasive sensing systems for human activity recognition

The annotation or labeling of activity, the recognition of activity, and the collection of data are three of the main functions of systems for smart sensing. In Tapia et al. [Tapia, Intille and Larson (2004)], a ubiquitous system for sensing in order to achieve human

activity recognition was suggested by the researchers. In a residential setting, the researchers placed devices with the purpose of monitoring an object's state. When considering the functions of the sensing system, there were a number of components included within the test, including the recognition of activities through supervised classification algorithms, the manual labeling of activities in the home via the context-aware experience sampling (EMS) tool, and the collection of data in its raw binary form.

Uddin [Uddin (2019)] proposed a wearable sensor-based system that relies on RNN on an edge device to predict activity. In this case, the input fed into the system is obtained from a variety of wearable healthcare sensors, such as a magnetometer, accelerometer, gyroscope, electrocardiograph (ECG), etc. The authors in this work used LSTM to model human activities which were represented by time-series based changes from the sensor.

Park et al. [Park, Kim, An et al. (2018)] proposed a system known as LiRed. The work reviewed a number of industrial robot manipulation algorithms, and it was found that the LSTM-based fault detection model was far more efficient in finding faults than the other six models analyzed.

Ferdowsi et al. [Ferdowsi, Challita and Saad (2019)], a convolutional neural network (CNN) together with LSTM is proposed in self-driving vehicles. The work proposes a deep learning approach in mobile edge analytics in the realm of intelligent transport systems (ITSs). The CNN algorithm is leveraged to find objects given an input image. The extracted features are then fed to an LSTM network which is employed in the next phase of processing.

Van Kasteren et al. [Van Kasteren, Noulas, Englebienne et al. (2008)] conducted in a residential setting, the researchers used a combination of wearable and ambient sensors. RFM DM 1810 kits allowed for the creation of a new wireless network; this enabled the simple introduction and installation of additional sensors in the existing system. The use of particular speech commands then assisted with performing the annotations. Conditional random fields (CRF) and the hidden Markov model (HMM) were two of the probabilistic models for activity recognition that the research adopted.

The activity recognition with ambient sensing (ARAS) dataset for human activity was the focus of another study [Alemdar, Ertan, Incel et al. (2013)]. The dataset included data on two residents of a property; the connection of seven types of sensor, with 20 sensors in total, to a house's ZigBee protocol enabled the generation of this dataset. Overall, the sensors detected a total of 27 activities during the study, which involved simultaneous collection. There were some good results from the study, after the application of a simple HMM algorithm to the sequential event data from the sensors. A further study conducted by Cook [Cook (2010)] involved the construction of a number of smart house testbeds; there was variation in the environment and type of property within the testbeds. The test gathered and passed on information by linking middleware and a network of sensors to a protocol similar to Jabber. The number of sensors used in the study was in the double figures; this allowed for the detection of movement from many residents and pets. Once data collection was complete, the researchers were able to use three different models, CRF, HMM, and the naive Bayes classifier (NBC), with the datasets.

Our focus, therefore, is to determine how a particular kind of deep learning method can be leveraged in increasing the accuracy of diverse human activity recognition. The study

explores how DRNN recognition can be developed, besides attempting to understand how it can be implemented on the edge computing devices.

2.2 Machine learning for activity recognition

Deep learning is an approach that has shown a great deal of promise regarding mining IoT data in noisy and complex scenarios [Hassan, Gumaei, Aloï et al. (2019); Gumaei, Hassan, Alelaiwi et al. (2019)]. It has, in fact, found many applications in the world of IoT [Mohammadi, Al-Fuqaha, Sorour et al. (2018)]. For instance, it can be used to predict future energy consumption trends on the basis of data collected from devices such as smart sensors. Based on such trends, proper planning can be done on the smart grid and even for an entire power supply system. This kind of efficiency means that deep learning is bound to play a massive role in the future of IoT-based systems. To address issues with classification and recognition, both deep learning and machine learning are very useful. Using ML algorithms in activity recognition could also be of potential benefit, as recent research has shown. Activity recognition has also seen the use of a number of exceptional algorithms, including the convolutional neural network (CNN) and support vector machine (SVM).

SVM is designed for binary classification [Noble (2006)]; it implements a nonlinear mapping (performed by an internal kernel product) of input data into a characteristic high-dimensional space, in which an optimal hyperplane is constructed to separate the data linearly into binary classification. However, for multiclass classification, SVM takes much more time as it applies a multiple binary classification tasks via one-versus-one.

This means that, if we have N classes, it requires $\frac{N(N-1)}{2}$ models.

The use of machine learning techniques in many research projects has also provided fairly good results. Alemdar et al. [Alemdar, Ertan, Incel et al. (2013)] analyzed event data from sequential sensors, the researchers adopted an HMM model. The study took place in two residential properties, each with varying environments, with the accuracy of activity recognition being 76.2% and 61.5%. Further researchers [Cook (2010)] have performed studies, including one study where the test used three models, CRF, NBC, and HMM, and took place in 11 environments. In the study, the researchers attempted to use a decision-tree-style classifier and three further classifiers to take the form of a hierarchical classifier. The test results demonstrated that the recognition accuracy of many datasets was higher with an ensemble classifier.

2.3 Edge computing approach

One of the most recently adopted distributed computing models is edge computing [Liu, Qiu, Zhang et al. (2020)]. The goal of edge computing is to take computing power away from the Cloud, a remote data center, and instead move it to the network edge; this meets the requirements of conserving energy, data privacy, and the location-aware processing of data needed for IoT applications that are time-restricted [Li, Ota and Dong (2018)]. There is a significant difference between edge device types, from powerful desktop computers to computers with comparatively low power. In recent years, systems for edge computing have seen major improvement, with deep learning and machine learning algorithms offering intelligent solutions for IoT applications used in quasi-real-time and

real-time ways.

Fog computing uses a centralized system that interacts with industrial gate way and embedded computer systems on a local area network [Bonomi, Milito, Zhu et al. (2012)]. On the other hand, edge computing performs much of the processing on embedded computing platforms directly interfacing with sensors and controllers. In essence, fog and edge technology link data to the world of the IoT. Progressive environments that involve rapid creation, placement, and use of diverse resources, require more developed techniques [Kratzke and Quint (2017); Pahl, El Ioini, Helmer et al. (2019)]. In a research conducted by Kratzke [Kratzke (2018)], two fundamental developments in cloud-native technology have been recognized and analyzed. The first trend points to the dynamic evolution of cloud computing technology architecture, which has optimized resource utilization within the cloud computing technology infrastructure. The second trend which has been evident is that the optimization of resource utilization has contributed to the emergence of diverse ways of how cloud application is used.

In some instances, the use of deep learning algorithms has enabled edge IoT applications to perform video recognition [Li, Ota and Dong (2018)]. First, it is essential to divide the neural model used into multiple layers; upper layers deploy from the Cloud, and lower layers deploy from the edge. This method is also able to provide a solution to the issue of scheduling that is the result of distributing the layers, and attempts to provide a guarantee of minimal communication and processing time cost amongst the various deep learning duties by splitting the layers up between the Cloud and the edge. This is not the only work that has considered the possible strategies for where the deep learning model's layers should be placed. Teerapittayanon et al. [Teerapittayanon, McDanel and Kung (2017)] looked at the automation of layers of the DNN, sending them to Cloud servers, sensors, and edge devices. As a result of this, there is a proposal of a training and collection scheme for different DNN applications used in combination to improve data privacy, enhance the fault tolerance of the system, and boost sensor fusion.

Even though deep learning techniques have been considered too heavy for application on the edge devices, DL-HAR framework proposed here-as far as we know-is the first of its kind that applied DRNN on edge devices for human activity recognition in a lightweight and non-expensive way. The closest academic work in this respect is Agarwal et al. [Agarwal and Alam (2020)], which propounded a lightweight model based on Shallow Recurrent Neural Network, in combination with the LSTM algorithm of deep learning. Our model has outperformed the Agarwal and Alam's to score a higher accuracy.

3 Proposed DL-HAR framework

In this section, we describe the DL-HAR framework in detail. This framework consists of a network of accelerometer-based wearable body sensors and an edge layer that receives their data and classifies them, as can be seen in Fig. 1. The scenario is to make the patient wear the sensors, which will load the data towards the edge's Raspberry Pi devices to lower the server's demand on resources. The edge device is located in-house near to the patient. In each edge layer, Docker containers are utilized as a lightweight virtualization tool; more details about Docker technology and its role in our framework can be found in Section 3.1. Section 3.2 also describes how the deep recurrent neural network is used to

classify human activity data in the proposed framework. As a communication protocol, we used The Message Queuing Telemetry Transport (MQTT) which achieved varied use in the IoT [Katsikeas, Fysarakis, Miaoudakis et al. (2017)]. MQTT comes with its advantages, it is characterized by smaller sized, reduced data packets, and also a relatively reduced power consumption. This is besides real-time data transfer and the fact that is relatively easy to implement. Hence, it meets the needs of our framework. It takes advantage of the subscribing operations for data exchange between clients and the servers. Outputs will be dispatched to the remote side (e.g., the hospital) where healthcare givers, relatives or emergency are notified in case of hazard activity. Software for patient remote monitoring can be developed as a mobile application to ensure fast reaction by caregivers and relative persons. Each cluster of sensors is combined with its dedicated Raspberry Pi, which serves as a bridge between the sensors and central server. The server receives the final analyzed data and can push the latest Docker image of the trained classifier. In the following subsections, we describe the used technologies in more detail.

3.1 Docker containers

The edge computing applications focus on lightweight virtualized resources as opposed to the bigger data center clouds, hence bringing services to the user [Pahl and Lee (2015)]. To carry out the human activity recognition in our framework, we used Docker containers that are application-oriented throughout the methods described in this paper. The base of Docker functionalities is a Docker engine, or container engine. This container technology is lightweight, and the components of the software are container-management focused [Morabito, Petrolo, Loscri et al. (2018)]. The technology also offers functional application programming interface (API), constructing, managing, and removing virtualized applications is simple. The use of containers that are application-oriented improves the ability to handle the demands of micro-service design; micro-service use is a great development towards the future deployment of IoT services. Easier management and deployment when using multiple hosts (virtual or physical) and more than one containerized application is in part achieved by container provisioning and orchestration; this is particularly true when using a data center setting.

3.2 Deep recurrent neural network (DRNN) architecture

RNN is a type of artificial neural network that is suitable for processing time-series data, generated from body sensors or smartphone devices. The algorithm can encode dependencies between inputs, but it is only limited to shorter data sequences. In the case of long data sequences, the algorithm will create a vanishing state in relation to its gradient. Therefore, LSTM units are used in DRNN architecture to overcome the challenges of traditional RNN units. The former is made of gates with the input layer having an input gate and the output layer a forget gate and an output gate. The schematic diagram in Fig. 1 shows a proposal of a DL-HAR system. The system is capable of end-to-end direct mapping using sensor inputs in unrefined multi-modal form to classify activity labels. In a set window of time, the system can classify labels for performed activities. Samples spaced evenly, to form a discrete structured sequence, such as (x_1, x_2, x_T) , are the input; separate samples indicate time t observations from the sensors, where the vector is a data point x_t .

To become maximum time (T) index windows, the samples become separated; then, they are moved into a deep recurrent neural network (DRNN) model based on stacked LSTM cells on top of each other.

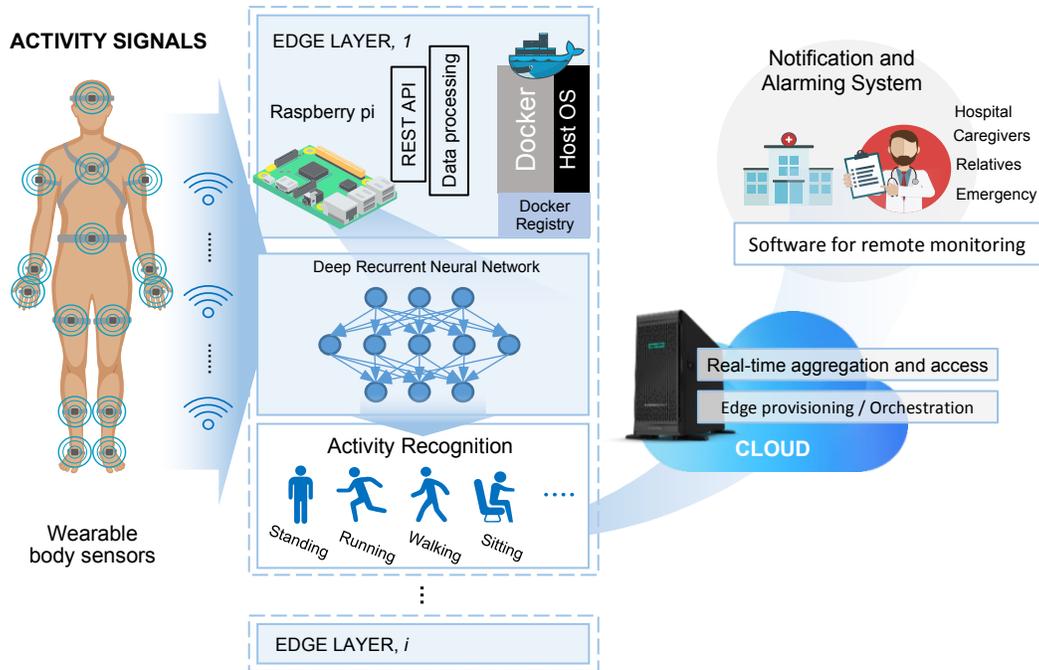


Figure 1: DL-HAR conceptual framework

Then, the model begins outputting predictions for activity labels, in the form of a score sequence. Every time step, i.e., $(y_1^l, y_2^l, \dots, y_T^l)$, receives a prediction; the symbol l represents the number of activity classes, and the vector of scores is $y_1^l \in R^l$, which represents individual predictions for input samples, x_t . The provision of a prediction of activity type that will occur at time t is in the form of a score given at every time step. Creating one prediction, by combining all the separate scores, will provide a prediction that will apply to the whole T window. In order to create probabilities from the score, over \mathcal{Y} we added a softmax function in the output layer. The probabilities we created, as in Eq. (1):

$$y = \frac{1}{T} \sum_{t=1}^T y_t^l \quad (1)$$

LSTM is selected to avoid the vanishing/exploding problem of gradient in a recurrent neural network [Pascanu, Mikolov and Bengio (2013)]. In a multi-layer network, gradients for layers in the deep architecture are computed as outputs of many activation functions. If these values are small, then they will vanish, contrary to values bigger than 1, which will probably explode. Thus, it is difficult to compute and update these values.

Accordingly, LSTM is used to learn long-range dependencies among a multi-layer network no matter how deep the network architecture is, or how long the input sequence

is [Bayer (2015)]. In Fig. 2, the inputs take the form of raw signals that come from multi-modal sensors, split into length T windows before entering into N hidden layers of the DRNN model that is based on LSTM. Scores for class predictions from every time step are the output of the model which combines the scores and then, to conclude the probability of class membership, feeds them to the softmax function in the output layer.

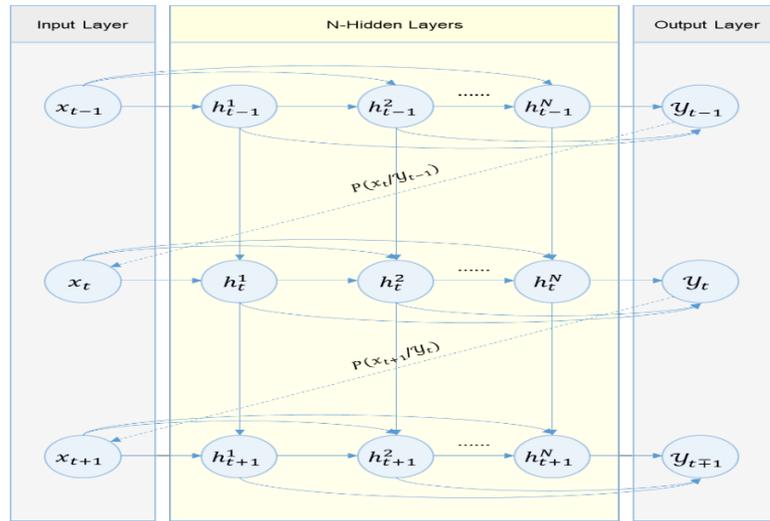


Figure 2: Proposed DRNN architecture

4 Experiment and discussion

In our experiment, we used the WISDM dataset [Kwapisz, Weiss and Moore (2011)], which is collected by the Wireless Sensor Data Mining Laboratory. In detail, this dataset is used as a case study to evaluate the proposed framework. During the experiment, a set of evaluation terms such as accuracy, precision, recall, and F1-score, as well as recognition time were used to show the effectiveness of the proposed DRNN model. The following subsections describe the dataset used and the experimental results with more detail.

4.1 Datasets

4.1.1 WISDM dataset

WISDM is a challenging dataset which contains 1,098,207 instances of 36 subjects collected by smart phone mobiles placed in the front leg pocket at 20 Hz of sampling frequency (20 data points per second). Each instance in the dataset contains a set of information including the *user id*, *activity class*, *timestamp*, *x-axis acceleration value*, *y-axis acceleration value*, and *z-axis acceleration value*. The activity class has six types of categorical names which are *standing*, *sitting*, *downstairs*, *upstairs*, *jogging*, and *walking*. Fig. 3 shows the number of instances in each activity class.

There is another version of the WISDM dataset consisting of a feature space extracted from the original input space of the acceleration time-series. In this feature space, the input acceleration time-series for every 10 s of each activity, containing 200 points of

continuous samples, is converted into 43 eigenvalues to form a 1×43 feature vector. Accordingly, a matrix of 5418×43 features is produced from $1,098,209 \times 3$ features of the dataset. To reduce the time of feature extraction, our framework deals with the original acceleration time-series directly without degrading the accuracy of the activity recognition task. In Kolosnjaji et al. [Kolosnjaji and Eckert (2015); Almaslukh, Al Muhtadi, Artoli (2018)], the authors divided this dataset into a training set and testing set using the leave-one-subject-out method to enable the researchers able to evaluate the performance of their proposed models.

Our experiment follows the same procedure, in which the dataset is divided into two sets: the training set, that contains the activities of subjects from 1 to 26, and the testing set, which has the activities of the other 10 subjects. A window of 6.4 s (a segment size of 128) is used for each activity to generate 7367 instances in the training set and 3026 instances in the testing set. The distribution of activities in the training set and testing set are shown in Tab. 1.

Table 1: Number of instances in training and testing sets of WISDM dataset

Activity Name	Training set.	Testing set.
Downstairs	557	277
Jogging	2450	890
Sitting	376	190
Standing	286	165
Upstairs	742	306
Walking	2956	1198
Total	7367	3026

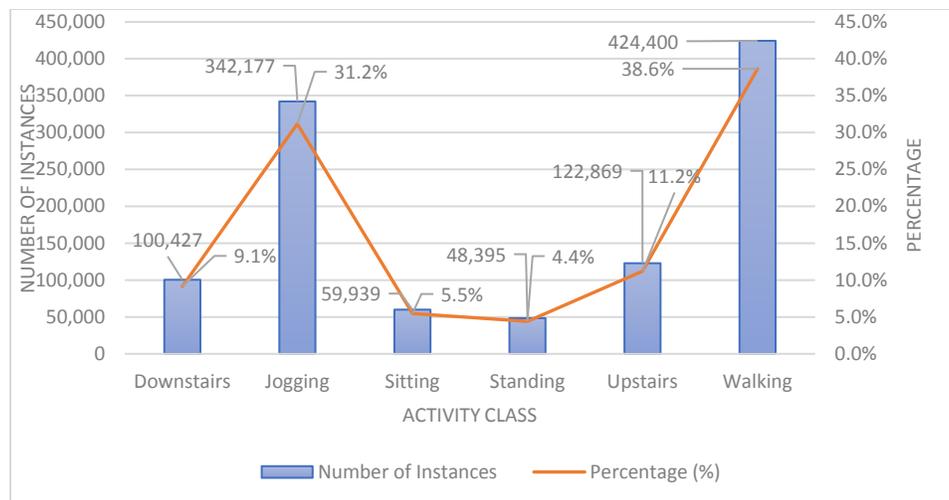


Figure 3: Distribution of instances in each activity of the Wireless Sensor Data Mining (WISDM) dataset

4.1.2 PUC-Rio dataset

This dataset has published by Pontifical Catholic University of Rio de Janeiro (PUC-Rio). It contains 165,633 continuous data points of four accelerometers in three axes: x , y , and z . It is collected for eight hours of five activities (sitting, sitting-down, standing, standing-up, and walking) from four subjects wearing four wearable accelerometer sensors placed on their left thigh, right ankle, right arm, and waist to generate a dataset in three axes: x , y , and z . The continuous data of each activity was recorded in one-second windows with 150 ms overlap. The number of instances in each activity is shown in Fig. 4.

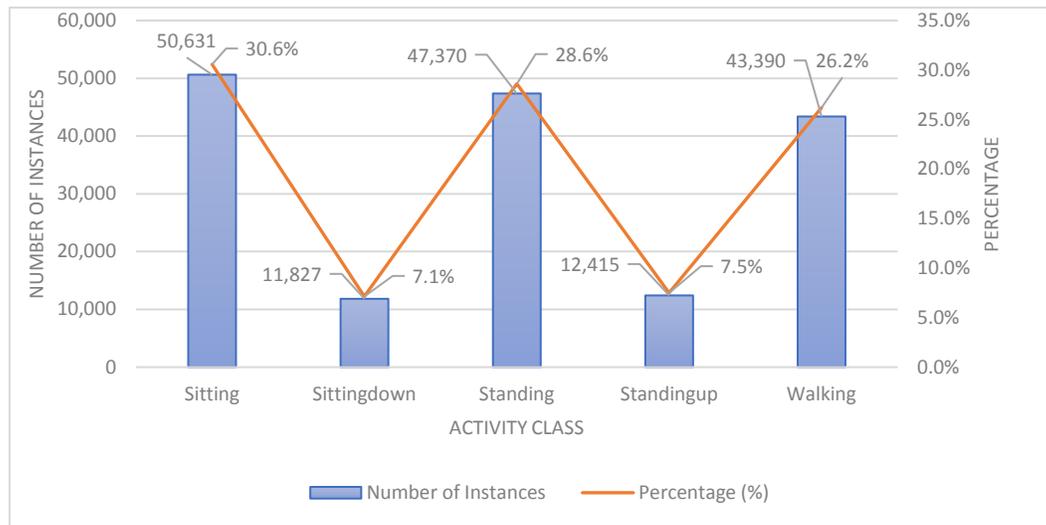


Figure 4: Distribution of instances in each activity of the Pontifical Catholic University of Rio de Janeiro (PUC-Rio) dataset

We divided the dataset into training set (70%) and testing set (30%) using the hold-out method, which is common in the machine learning field. A window of 5 s (a segment size of 5) is used for each activity to generate 23,188 instances in the training set and 9938 instances in the testing set. The distribution of activities in the training and testing sets is described in Tab. 2.

Table 2: Number of instances in training and testing sets of PUC-Rio dataset

Activity Name	Training set	Testing set
Sitting	7114	3012
Sitting down	1680	686
Standing	6611	2863
Standing up	1734	749
Walking	6049	2628
Total	23188	9938

4.2 Evaluation measures

To evaluate the efficiency of the DRNN model for the proposed framework, we calculated the recognition time spent on the Raspberry Pi edge device for recognizing the activities through a number of requests. In addition to that, we also used four of the most common validation measures to evaluate the performance of the model. These validation measures are accuracy, precision, recall, and F1-score, computed as follows:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (2)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (3)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (4)$$

$$F1\text{-score} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (5)$$

where TP and FP are the true and false positive rates; and TN and FN are the true and false negative rates.

4.3 Experimental setup

The configuration of our experiment consists of one Raspberry Pi3 as an edge device for recognizing the requested activity, one laptop as a sensor to send the activity time-series data, and one server as a Cloud for training the DRNN model and distribute it to the Raspberry Pi3. The Raspberry Pi3 used in the experiment is a third-generation model with a 1.2 GHz 64-bit quad-core CPU and 1 GB RAM. Also, the laptop is a Core i7 with a 2.60 GHz CPU and 16.0 GB RAM, operating Windows 10. The server is a workstation with a 2.2 GHz Intel Xeon processor, 13.75 MB cache, 10 cores, 32 GB DDR4-2666, and ECC SDRAM (2×16 GB), operating Windows 10. Keras with TensorFlow libraries in the Python programming language are used to implement the experiment of this research work.

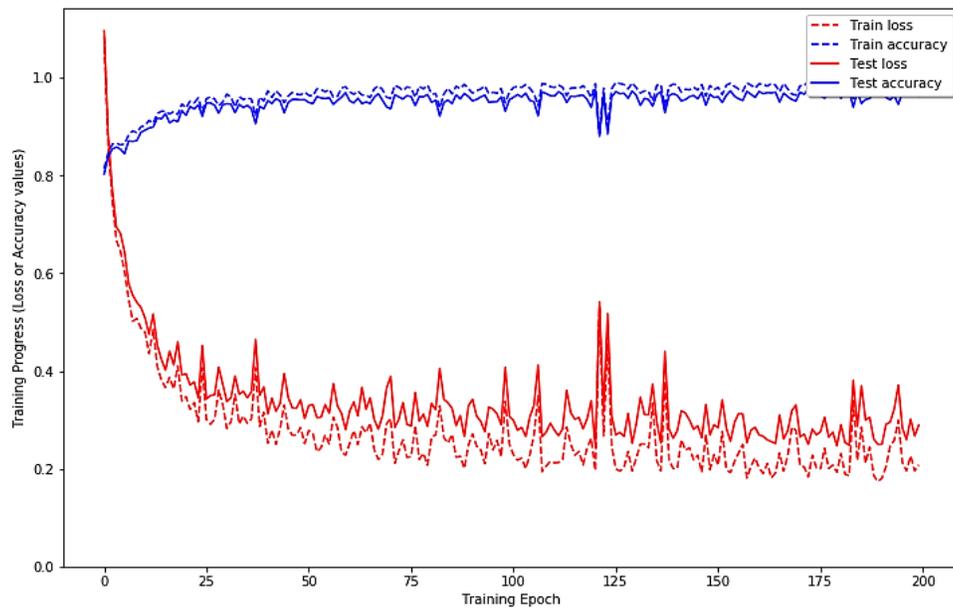
After setting up the configuration, the DRNN model is trained on the training set at the server side and pushed as a trained Docker image to the Raspberry Pi3 via the TCP protocol. During the training phase, the model is trained with different parameter values using a grid search technique to select the best possible values. Grid search is an optimization technique which can use a specific range of different values to find the best combination of hyper-parameters [Goliatt, Capriles and Duarte (2018)]. We chose this technique as it allowed us to enter some good probable values for the model parameters based on our experience in the deep learning field. For example, the number of hidden layers parameter is initialized with a small range of integer values (2, 3, and 4). From this range, we found that the model performed best when the number of hidden layers was three. Tab. 3 gives the hyper-parameters of the proposed DRNN model with their corresponding best values. Moreover, Figs. 5 and 6 display the accuracy and training loss of the DRNN model on the datasets used over 200 iterations (epochs).

Table 3: Hyper parameter values of the DRNN model

Hyper Parameter	Value
Number of hidden layers	3
Number of epochs	200
L2_LOSS	0.0015
Learning rate	0.0025
Number of hidden neurons	50
Batch size	32
Optimizer	AdamOptimizer

4.4 Experimental results and comparisons

In this section, the results of the proposed DRNN model are demonstrated and compared. The comparisons are performed with one of the most popular shallow learning models, namely the support vector machine (SVM) model with radial base function (RBF), as well as some comparisons with the state-of-the-art models applied on the same datasets.

**Figure 5:** Training progress of the DRNN model on WISDM dataset over 200 epochs

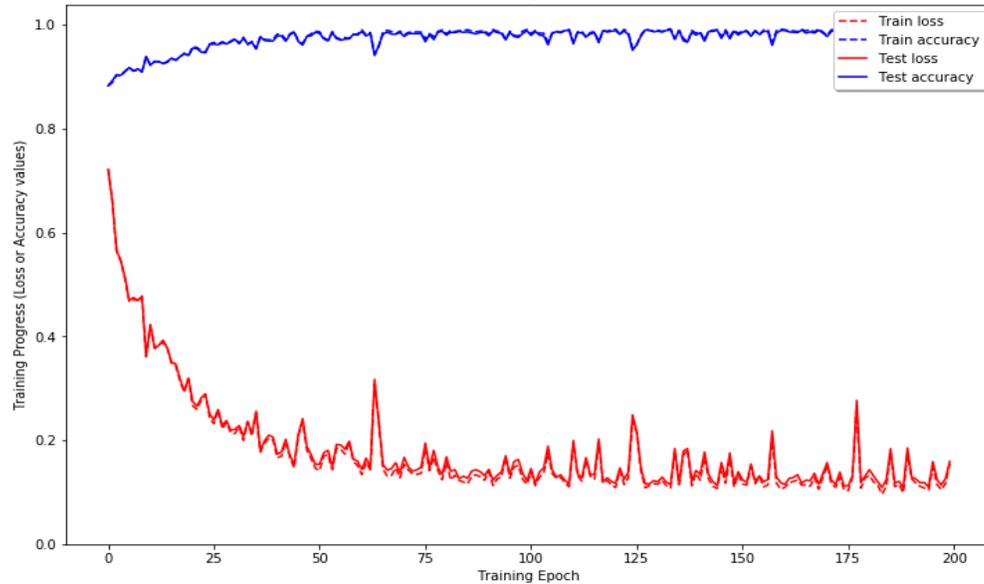


Figure 6: Training progress of the DRNN model on PUC-Rio dataset over 200 epochs

An RBF kernel is used with SVM as a mapping function to map the data from low-dimensional to high-dimensional spaces. Even though RBF allows SVM to work better for higher-dimensional feature vectors, it becomes computationally very expensive. Figs. 7-10 demonstrate the normalized confusion matrices of the DRNN and SVM models. From these matrices, we found that the DRNN model achieves 97.03% accuracy compared to the SVM which achieves 82.98% accuracy on the WISDM dataset. Additionally, we found that the proposed DRNN model attains 99.024% accuracy compared to the SVM model that attains 95.775% accuracy on the PUC-Rio dataset.

Downstairs	0.88	0.01	0.00	0.00	0.10	0.01
Jogging	0.01	0.99	0.00	0.00	0.00	0.00
Sitting	0.01	0.00	0.95	0.01	0.00	0.00
Standing	0.02	0.00	0.01	0.97	0.00	0.00
Upstairs	0.05	0.01	0.00	0.00	0.92	0.01
Walking	0.00	0.00	0.00	0.00	0.00	0.99
	Downstairs	Jogging	Sitting	Standing	Upstairs	Walking

Figure 7: Normalized confusion matrix of the DRNN model on the WISDM dataset

Downstairs	0.11	0.02	0.00	0.02	0.26	0.58
Jogging	0.00	0.97	0.00	0.00	0.00	0.02
Sitting	0.00	0.00	0.96	0.04	0.00	0.00
Standing	0.00	0.00	0.00	0.97	0.03	0.01
Upstairs	0.05	0.10	0.00	0.01	0.38	0.46
Walking	0.00	0.01	0.00	0.00	0.00	0.99
	Downstairs	Jogging	Sitting	Standing	Upstairs	Walking

Figure 8: Normalized confusion matrix of the SVM model on the WISDM dataset

Sitting	1.00	0.00	0.00	0.00	0.00
Sitting down	0.01	0.96	0.00	0.03	0.01
Standing	0.00	0.00	1.00	0.00	0.00
Standing up	0.01	0.02	0.02	0.95	0.00
Walking	0.00	0.00	0.00	0.00	0.99
	Sitting	Sitting down	Standing	Standing up	Walking

Figure 9: Normalized confusion matrix of the DRNN model on the PUC-Rio dataset

Sitting	0.99	0.00	0.00	0.01	0.00
Sitting down	0.00	0.89	0.03	0.03	0.05
Standing	0.00	0.00	0.97	0.00	0.03
Standing up	0.00	0.05	0.11	0.68	0.16
Walking	0.00	0.00	0.00	0.00	1.00
	Sitting	Sitting down	Standing	Standing up	Walking

Figure 10: Normalized confusion matrix of the SVM model on the PUC-Rio dataset

Moreover, Tabs. 4-7 show the validation measures of the DRNN and SVM models, while Tabs. 8 and 9 list the F1-scores of the proposed DRNN model against the-state-of-the-art models on both datasets. Because the results of previous works on the datasets were conducted on different test modes and it is hard to compare them, the result of these works are reproduced with our test mode for a fair comparison. In this comparison, F1-score is used because it provides more realistic estimates of real-world classification problems when the class distribution is imbalanced as in the activity recognition problem.

Table 4: Validation measures of the DRNN model on the WISDM dataset

Activity Name	Precision (%)	Recall (%)	F1-Score (%)
Downstairs	89	88	88
Jogging	99	99	99
Sitting	99	98	98
Standing	99	97	98
Upstairs	89	92	90
Walking	99	99	99
Weighted avg.	97	97	97

Table 5: Validation measures of the SVM model on the WISDM dataset

Activity Name	Precision (%)	Recall (%)	F1-Score (%)
Downstairs	63	11	19
Jogging	95	97	96
Sitting	98	96	97
Standing	90	97	93
Upstairs	58	38	46
Walking	77	99	87
Weighted avg.	81	83	80

Table 6: Validation measures of the DRNN model on the PUC-Rio dataset

Activity Name	Precision (%)	Recall (%)	F1-Score (%)
Sitting	100	100	100
Sitting down	97	96	96
Standing	99	100	99
Standing up	95	95	95
Walking	100	99	100
Weighted avg.	99	99	99

Table 7: Validation measures of the SVM model on the PUC-Rio dataset

Activity Name	Precision (%)	Recall (%)	F1-Score (%)
Sitting	100	99	100
Sitting down	94	89	92
Standing	96	97	97
Standing up	93	68	79
Walking	92	100	96
Weighted avg.	96	96	96

Table 8: Comparison results of F1-score for the proposed DRNN model against the state-of-the-art models on the WISDM dataset. CNN: convolutional neural network

Model	F1-Score (%)
Basic features and random forest [Ignatov (2018)]	82.66
PCA and random forest [Ignatov (2018)]	75.28
Handcrafted features and dropout [Kolosnjaji and Eckert (2015)]	85.36
Statistical features and shallow CNN [Ignatov (2018)]	93.32
Statistical features and deep CNN [Almaslukh, Al Muhtadi, Artoli (2018)]	95.13
Proposed DRNN model	97.00

Table 9: Comparison results of F1-score for the proposed DRNN model against the state-of-the-art models on the PUC-Rio dataset. SVM-RBF: support vector machine with radial base function

Model	F1-Score (%)
Adaboost [Ugulino, Cardador, Tucakovic et al. (2012)]	84
SVM-RBF [Cheng, Guan, Zhu et al. (2017)]	96
Proposed DRNN model	99

As we see from the previous tables and figures, the DRNN model has the highest results and outperforms the SVM model and the state-of-the-art models with regards to all validation measures, which are highlighted in bold font. The reason that our model achieves high accuracy is the deep architecture of the model and the best values of its hyper parameters, enabling it to learn the feature representation of the time-series data of activities in the best way and recognize them accurately.

To evaluate the efficiency of the DRNN model, the recognition time taken to recognize 100 instances from the testing set of the WISDM dataset is computed and compared with the SVM model, as shown in Fig. 11. Furthermore, the total recognition time taken to recognize all activities in the testing set is also computed and shown in Fig. 12.

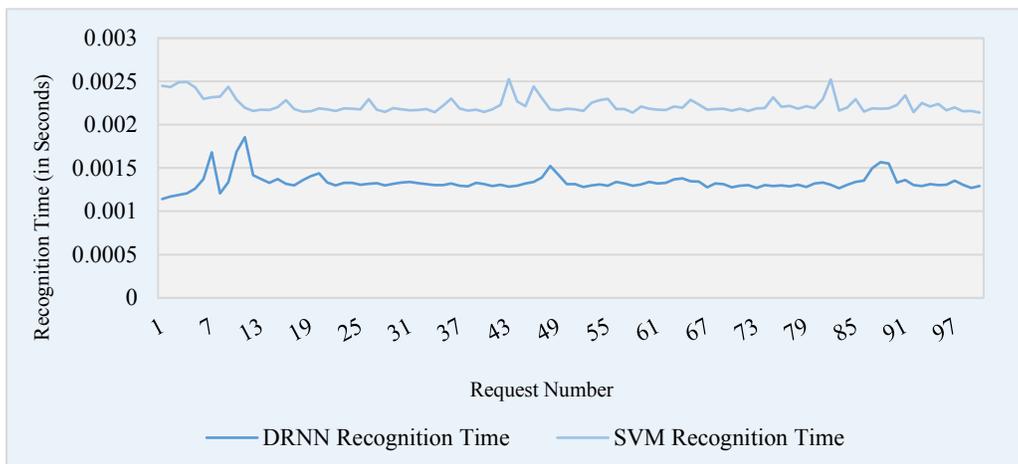


Figure 11: Recognition time of DRNN and SVM models to recognize one activity per request for 100 requests

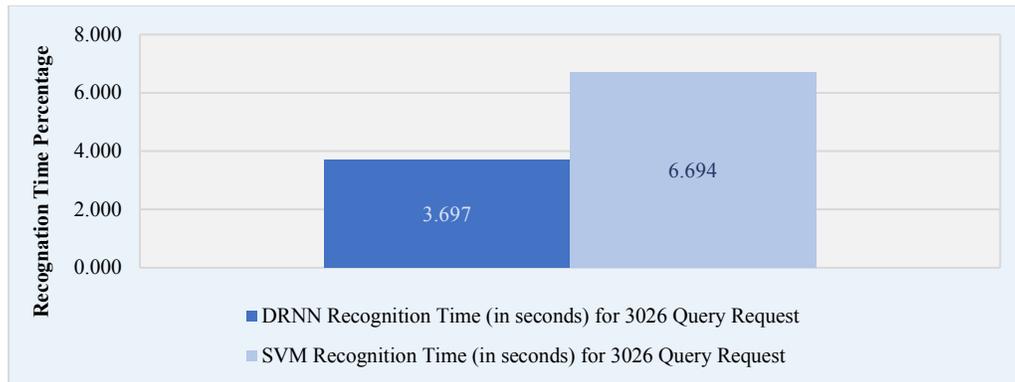


Figure 12: Recognition time of DRNN and SVM models to recognize all activities in the testing set of the WISDM dataset

From Fig. 11, we notice that the recognition time of the DRNN model is lower than the SVM model. In addition to that, Fig. 12 shows that the recognition time of the DRNN model on the testing set of the WISDM dataset is almost 36% of the recognition time taken by the SVM model. Even though the training time of the DRNN model is higher than the training time of SVM, the recognition time of the DRNN model is lower than the recognition time of SVM. This situation can be exploited to train the RDNN model in off-line mode and deploy it for recognition in on-line mode, utilizing this efficiency for real-time activity recognition. The last step in the experiment is conducted to prove the effectiveness of the proposed model as a lightweight model in terms of CPU utilization and memory utilization. To support this, we executed the model on an edge device, which was a third-generation Raspberry Pi3 with a 1.2 GHz 64-bit quad-core CPU and 1 GB RAM at different numbers of hidden neurons. Fig. 13 demonstrates the CPU utilization and memory utilization of the model at different loads.

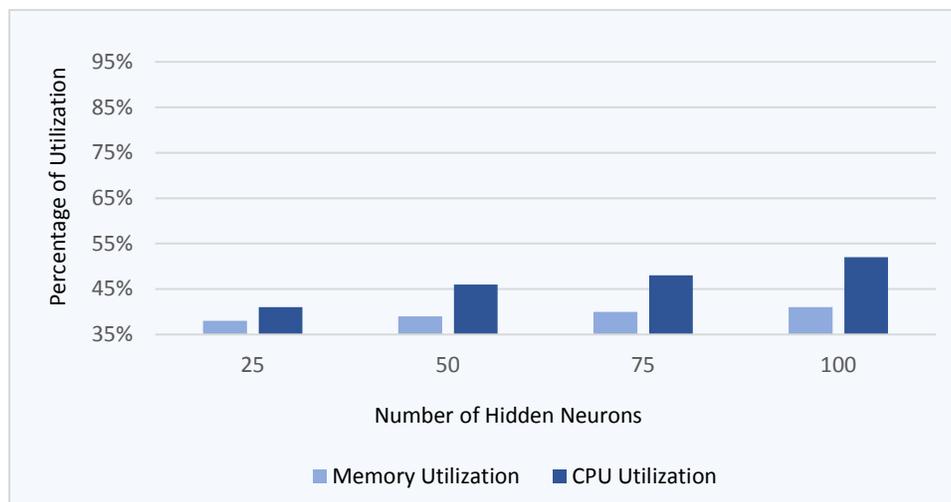


Figure 13: The CPU utilization and memory utilization of executing the DRNN model on the edge device at different numbers of hidden neurons

During the computation of the CPU utilization and memory utilization, we noticed that they were 10% and 36%, respectively, because some tasks of the operating system were running. In general, from Fig. 13, we can see that the CPU utilization and memory utilization of the DRNN model are acceptable and are not significantly affected when the number of hidden neurons is in the range between 25 and 100.

To conclude, our proposed deep learning-based human activity recognition framework based on edge computing aims to increase the speed of decision making. It uses a deep learning algorithm to reduce communication to the Cloud servers, thus reducing delays and round trips that may be unnecessary. Based on experimental study discussed in this section, the potentials of edge computing distributed framework utilization is effective and efficient. We examined the accuracy and recognition time performance measures during the experiment to show how it outperformed the state-of-the-art models.

4.5 Statistical analysis of performance results

Statistical analysis of performance results on different random test data samples is very important to show the effectiveness of the proposed DRNN model. To achieve this statistical analysis, we repeated the evaluation of the DRNN model twelve times on different random data samples from the two datasets. Then, for each dataset, we used a one-sample t test of the SPSS software to compare these F1-score results with the F1-score result obtained in the previous experimental results. The one-sample t test is a parametric statistical analysis method used to compare the mean of a random sample to a hypothesized or known mean value and test for a deviation from that value, calculated using Eq. (6):

$$t = \frac{\mu - \bar{x}}{\bar{s}} \quad (6)$$

where \bar{x} represents the hypothesized mean value, μ denotes the sample mean, and \bar{s} is the estimated standard error, which is computed based on s , the sample standard deviation of the mean, and n , the sample size, using the Eq. (7):

$$\bar{s} = \frac{s}{\sqrt{n}} \quad (7)$$

In our case, the one-sample t test is used to determine whether the mean of F1-score results for the proposed DRNN model evaluated on different random test data samples is statistically different from the F1-score result obtained in the comparisons of other works' models or not. In other words, it is used here to see if there are statistically significant differences in the F1-score results of the DRNN model when it is tested on a different set of data samples.

The hypotheses of the one-sample t test are expressed as follows:

- Null hypothesis (H_0): $\mu = \bar{x}$ ("the mean of the F1-score results for the twelve times of evaluation is equal to the hypothesized F1-score value, which is 97% on the WISDM dataset, and 99% on the PUC-Rio dataset").
- Alternative hypothesis (H_1): $\mu \neq \bar{x}$ ("the mean of the F1-score results for the twelve times of evaluation is not equal to the hypothesized F1-score value, which is 97% on the WISDM dataset, and 99% on the PUC-Rio dataset").

Tab. 10 shows the percentage values of F1-scores that are rounded to the nearest integer values for the twelve times of evaluation on the different random data samples of WISDM and PUC-Rio datasets.

Table 10: F1-score results of the proposed DRNN model for the twelve times of evaluation on different random data samples of WISDM and PUC-Rio datasets

Evaluation no.	F1-Score (%)	
	WISDM dataset	PUC-Rio dataset
1	97.00	99.00
2	97.00	98.00
3	97.00	99.00
4	97.00	99.00
5	98.00	99.00
6	98.00	99.00
7	97.00	99.00
8	97.00	98.00
9	97.00	99.00
10	97.00	99.00
11	98.00	98.00
12	97.00	99.00

After applying the, Tabs. 11 and 12 show the statistics and results of the one-sample t test for the different evaluations on the WISDM dataset. Tabs. 13 and 14 illustrate the results of the one-sample t test for the different evaluations on the PUC-Rio dataset.

Table 11: One-sample t test statistics of the twelve F1-score results for the different random data samples of the WISDM dataset

	N	Mean	Standard Deviation	Standard Error Mean
F1-Score	12	97.2500%	0.45227	0.13056

Table 12: One-Sample t test results of the twelve F1-score results for the different random data samples of the WISDM dataset

Test Value=97%						
t	Degrees of freedom	Significance (2-tailed)	Mean Difference	95% Confidence Interval of the Difference		
				Lower	Upper	
F1-Score	1.915	11	0.082	0.25000	-0.0374	0.5374

Table 13: One-sample t test statistics of the twelve F1-score results for the different random data samples of the PUC-Rio dataset

	N	Mean	Standard Deviation	Standard Error Mean
F1-Score	12	98.7500%	0.45227	0.13056

Table 14: One-Sample t test results of the twelve F1-score results for the different random data samples of the PUC-Rio dataset

Test Value=99%						
t	Degrees of freedom	Significance (2-tailed)	Mean Difference	95% Confidence Interval of the Difference		
				Lower	Upper	
F1-Score	-1.915	11	0.082	-0.25000	-0.5374	0.0374

As shown in Tab. 12, since the t test value is equal to 1.915 and p -value > 0.082 that is greater than the selected interval level of the difference ($\alpha=0.05$), we accept the null hypothesis (H_0). Similarly, from the results of Tab. 14, since the t test value is equal to -1.915 and p -value > 0.082 , which is greater than 0.05, we also accept the null hypothesis (H_0). Therefore, there are no statistically significant differences in the F1-score results of the DRNN model when it is tested twelve times on a different set of data samples.

5 Conclusions

The past few years have seen a rise in popularity of the use of mobile and wearable sensors in Human Activity Recognition (HAR). The application has been experienced in the fields of health, education, entertainment, and also human surveillance. Presently, edge computing has successful applications in minimizing communication latency and congestion in networks. In this research work, we proposed a new deep learning-based human activity recognition (DL-HAR) framework for edge computing. This work aimed to improve the primary care of patients in an uncritical stage but who are in need of continuous monitoring by a healthcare professional. The motivation behind this work is the fast evolution in sensor design which can allow one to collect information and transmit it in real time. Basically, we proposed a lightweight and effective DRNN architecture using LSMT cells which contain three hidden layers as well as the input and output layers. Each hidden layer consists of 50 neurons. The framework starts by training the DRNN model at the server side due its high capability and sends the image of the trained DRNN model to Docker containers on the Raspberry Pi3 edge devices to recognize the activity time-series data coming from sensors or smartphone devices. A WISDM dataset was used in our experiment to show the applicability of the proposed framework and evaluate its deep learning model based on a set of evaluation measures. The experimental results show the effectiveness and efficiency of the DRNN model. In summary, we can say that the proposed framework is a lightweight and effective solution for human activity recognition utilizing edge computing technology. Even though the

proposed framework is evaluated by the experiments on two real representative public datasets that are collected from accelerometer sensors, conducting the experiments on real-time live noisy data is one limitation of this study and needs to be investigated in future research. Another limitation of the study is related to the changes in the instruments of the framework that may lead to changes in the results. Therefore, the validity of the results using different instruments can be also verified in another future work. Moreover, in the future work, we will apply our framework to detect human activities from time-series data of online connected sensors. We also plan to use the proposed framework for solving the scalability issue in the edge computing environment.

Acknowledgement: This work is financially supported by the Deanship of Scientific Research, Research Chair of Pervasive and Mobile Computing at King Saud University.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- Agarwal, P.; Alam, M.** (2020): A lightweight deep learning model for human activity recognition on edge devices. *Procedia Computer Science*, vol. 167, pp. 2364-2373.
- Alemdar, H.; Ertan, H.; Incel, O. D.; Ersoy, C.** (2013): ARAS human activity datasets in multiple homes with multiple residents. *7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops*, pp. 232-235
- Almaslakh, B.; Al Muhtadi, J.; Artoli, A. M.** (2018): A robust convolutional neural network for online smartphone-based human activity recognition. *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 2, pp. 1609-1620.
- Al-Rakhami, M.; Alsahli, M., Hassan, M. M.; Alamri, A.; Guerrieri, A. et al.** (2018): Cost efficient edge intelligence framework using docker containers. *IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th Intl Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress*.
- Al-Rakhami, M.; Gumaei, A.; Alsahli, M.; Hassan, M. M.; Alamri, A. et al.** (2020): A lightweight and cost effective edge intelligence architecture based on containerization technology. *World Wide Web*, vol. 23, pp. 1341-1360.
- Bayer, J. S.** (2015): *Learning Sequence Representations*, Technische Universität MÜNchen (Ph.D. Dissertation). Technische Universität München, Arcisstraße, München, Germany.
- Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S.** (2012): Fog computing and its role in the internet of things. *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, pp. 13-16.

- Cheng, L.; Guan, Y.; Zhu, K.; Li, Y.** (2017): Recognition of human activities using machine learning methods with wearable sensors. *IEEE 7th Annual Computing and Communication Workshop and Conference*, pp. 1-7.
- Cook, D. J.** (2010): Learning setting-generalized activity models for smart spaces. *IEEE Intelligent Systems*, vol. 27, no. 1, pp. 32-38.
- Esteva, A.; Robicquet, A.; Ramsundar, B.; Kuleshov, V.; DePristo, M. et al.** (2019): A guide to deep learning in healthcare. *Nature medicine*, vol. 25, no. 1, pp. 24-29.
- Ferdowsi, A.; Challita, U.; Saad, W.** (2019): Deep learning for reliable mobile edge analytics in intelligent transportation systems: an overview. *IEEE Vehicular Technology Magazine*, vol. 14, no. 1, pp. 62-70.
- Fernández-Caramés, T. M.; Fraga-Lamas, P.** (2019): Towards next generation teaching, learning, and context-aware applications for higher education: a review on blockchain, IoT, fog and edge computing enabled smart campuses and universities. *Applied Sciences*, vol. 9, no. 21, pp. 4479.
- Goliatt, L.; Capriles, P.; Duarte, G.** (2018): Modeling heating and cooling loads in buildings using gaussian processes. *IEEE Congress on Evolutionary Computation*, pp. 1-6
- Greco, L.; Ritrovato, P.; Xhafa, F.** (2019): An edge-stream computing infrastructure for real-time analysis of wearable sensors data. *Future Generation Computer Systems*, vol. 93, pp. 515-528.
- Gumaei, A.; Hassan, M. M.; Alelaiwi, A.; Alsaman, H.** (2019): A hybrid deep learning model for human activity recognition using multimodal body sensing data. *IEEE Access*, vol. 7, pp. 99152-99160.
- H De La Iglesia, D.; Villarrubia González, G.; Sales Mendes, A.; Jiménez-Bravo, D.; Barriuso, A. L.** (2019): Architecture to embed software agents in resource constrained internet of things devices. *Sensors*, vol. 19, no. 1, pp. 100.
- Hammerla, N. Y.; Halloran, S.; Ploetz, T.** (2016): Deep, convolutional, and recurrent models for human activity recognition using wearables. arXiv:1604.08880.
- Hassan, M. M.; Gumaei, A.; Aloji, G.; Fortino, G.; Zhou, S. M.** (2019). A Smartphone-enabled fall detection framework for elderly people in connected home healthcare. *IEEE Network*, vol. 33, no. 6, pp. 58-63.
- Ignatov, A.** (2018): Real-time human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing*, vol. 62, pp. 915-922.
- Jan, B.; Farman, H.; Khan, M.; Imran, M.; Islam, I. U. et al.** (2019): Deep learning in big data analytics: a comparative study. *Computers & Electrical Engineering*, vol. 75, pp. 275-287.
- Katsikeas, S.; Fysarakis, K.; Miaoudakis, A.; Van Bemten, A.; Askoxylakis, I. et al.** (2017): Lightweight & secure industrial IoT communications via the MQ telemetry transport protocol. *IEEE Symposium on Computers and Communications*, pp. 1193-1200.
- Kolosnjaji, B.; Eckert, C.** (2015): Neural network-based user-independent physical activity recognition for mobile devices. *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 378-386 Springer, Cham.

- Kratzke, N.** (2018): A brief history of cloud application architectures. *Applied Sciences*, vol. 8, no. 8, pp. 1368.
- Kratzke, N.; Quint, P.** (2017): Understanding cloud-native applications after 10 years of cloud computing-a systematic mapping study. *Journal of Systems and Software*, vol. 126, pp. 1-16.
- Kwapisz, J. R.; Weiss, G. M.; Moore, S. A.** (2011): Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74-82.
- Li, H.; Ota, K.; Dong, M.** (2018). Learning IoT in edge: deep learning for the internet of things with edge computing. *IEEE Network*, vol. 32, no. 1, pp. 96-101.
- Li, X.; Huang, X.; Li, C.; Yu, R.; Shu, L.** (2019): EdgeCare: leveraging edge computing for collaborative data management in mobile healthcare systems. *IEEE Access*, vol. 7, pp. 22011-22025.
- Liu, Z.; Qiu, X.; Zhang, S.; Deng, S.; Liu, G.** (2020): Service scheduling based on edge computing for power distribution IoT. *Computers, Materials & Continua*, vol. 62, no. 3, pp. 1351-1364.
- Merelli, I.; Fornari, F.; Tordini, F.; D'Agostino, D.; Aldinucci, M. et al.** (2019): Exploiting docker containers over grid computing for a comprehensive study of chromatin conformation in different cell types. *Journal of Parallel and Distributed Computing*, vol. 134, pp. 116-127.
- Mishra, A.; Tripathi, K.; Gupta, L.; Singh, K. P.** (2019): Long short-term memory recurrent neural network architectures for melody generation. *Soft Computing for Problem Solving*, pp. 41-55.
- Mohammadi, M.; Al-Fuqaha, A.; Sorour, S. M.; Guizani, M.** (2018): Deep learning for IoT big data and streaming analytics: a survey. *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923-2960.
- Morabito, R.** (2017): Virtualization on internet of things edge devices with container technologies: a performance evaluation. *IEEE Access*, vol. 5, pp. 8835-8850.
- Morabito, R.; Petrolo, R.; Loscri, V.; Mitton, N.** (2018): LEGIoT: a lightweight edge gateway for the internet of things. *Future Generation Computer Systems*, vol. 81, pp. 1-15.
- Noble, W. S.** (2006): What is a support vector machine? *Nature Biotechnology*, vol. 24, no. 12, pp. 1565- 1567.
- Ordóñez, F. J.; Roggen, D.** (2016): Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, vol. 16, no. 1, pp. 115.
- Pahl, C.; El Ioini, N.; Helmer, S.; Lee, B.** (2019): A semantic pattern for trusted orchestration in IoT edge clouds. *Internet Technology Letters*, vol. 2, no. 3, pp. 95.
- Pahl, C.; Lee, B.** (2015): Containers and clusters for edge cloud architectures-a technology review. *3rd International Conference on Future Internet of Things and Cloud*, pp. 379-386.
- Park, D.; Kim, S.; An, Y.; Jung, J. Y.** (2018): Lired: a light-weight real-time fault detection system for edge computing using LSTM recurrent neural networks. *Sensors*, vol. 18, no. 7, pp. 2110.

- Park, H. D.; Min, O. G.; Lee, Y. J.** (2017): Scalable architecture for an automated surveillance system using edge computing. *Journal of Supercomputing*, vol. 73, no. 3, pp. 926-939.
- Pascanu, R.; Mikolov, T.; Bengio, Y.** (2013): On the difficulty of training recurrent neural networks. International conference on machine learning. *International Conference on Machine Learning*, pp. 1310-1318.
- Ravi, D.; Wong, C.; Lo, B.; Yang, G. Z.** (2016): Deep learning for human activity recognition: a resource efficient implementation on low-power devices. *IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks*, pp. 71-76.
- Ronao, C. A.; Cho, S. B.** (2016): Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications*, vol. 59, pp. 235-244.
- Sahinovic, S.; Dzebo, A.; Ustundag, B. C.; Golubovic, E.; Uzunovic, T.** (2018): An open and extensible data acquisition and processing platform for rehabilitation applications. *International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies*, pp. 394-406.
- Samarah, S.; Zamil, M. G. A.; Rawashdeh, M.; Hossain, M. S.; Muhammad, G. et al.** (2018): Transferring activity recognition models in FOG computing architecture. *Journal of Parallel and Distributed Computing*, vol. 122, pp. 122-130.
- Tapia, E. M.; Intille, S. S.; Larson, K.** (2004): Activity recognition in the home using simple and ubiquitous sensors. *International Conference on Pervasive Computing*, Springer, pp. 158-175. Springer, Berlin, Heidelberg,
- Teerapittayanon, S.; McDanel, B.; Kung, H.** (2017): Distributed deep neural networks over the cloud, the edge and end devices. *The Distributed Computing Systems, IEEE 37th International Conference*, pp. 328-339.
- Trihinas, D.; Pallis, G.; Dikaiakos, M.** (2018): Low-cost adaptive monitoring techniques for the internet of things. *IEEE Transactions on Services Computing*, pp. 1.
- Uddin, M. Z.** (2019): A wearable sensor-based activity prediction system to facilitate edge computing in smart healthcare system. *Journal of Parallel and Distributed Computing*, vol. 123, pp. 46-53.
- Ugulino, W.; Cardador, D.; Vega, K.; Velloso, E.; Milidiú, R. et al.** (2012): Wearable computing: accelerometers' data classification of body postures and movements. *Advances in Artificial Intelligence-SBIA*, pp. 52-61. Springer.
- Uzunovic, T.; Golubovic, E.; Tucakovic, Z.; Acikmese, Y.; Sabanovic, A.** (2018): Task-based control and human activity recognition for human-robot collaboration. *IECON 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 5110-5115.
- Van Kasteren, T.; Noulas A.; Englebienne G.; Kröse B.** (2008): Accurate activity recognition in a home setting. *Proceedings of the 10th International Conference on Ubiquitous Computing*, pp. 1-9.
- Yin, Y.; Zeng, Y.; Chen, X.; Fan, Y.** (2016): The internet of things in healthcare: an overview. *Journal of Industrial Information Integration*, vol. 1, pp. 3-13.