Decision Analytics Journal 8 (2023) 100282

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

Enhancing human activity recognition using features reduction in IoT edge and Azure cloud

Ayman A. Wazwaz [a,*], Khalid M. Amin [b], Noura A. Semari [b], Tamer F. Ghanem [b]

[a] Computer Engineering Department, College of Information Technology and Computer Engineering, Palestine Polytechnic University, Hebron, Palestine
[b] Department of Information Technology, Faculty of Computers and Information, Menofiya University, Shebin El Kom, Egypt

ARTICLE INFO

Keywords:
Machine learning
Internet of Things (IoT)
IoT Edge
Cloud computing
Feature selection
Principal component analysis
Human activity recognition

ABSTRACT

The Internet of Things (IoT), cloud computing, and machine learning opened an opportunity for new smart systems. These technologies have triggered huge traffic and delay by continuously transmitting telemetry data to the cloud. IoT edge choice made decision-making closer to the environment, which decreases traffic and speeds up response time. Human activity recognition (HAR) systems, among other IoT applications, require systems with quick response time; reduce costs using constrained traffic to the cloud while maintaining accurate prediction results. This study proposes an application of HAR for predicting activities using up to three smartphone accelerometers. Three models are developed, trained, and deployed to achieve the necessary accuracy at the IoT edge and in the cloud, with an acceptable response time. Since each coordinate value from the three accelerometers has different importance in activity category prediction, focusing on the data from the most related values can help minimize the amount of information transferred from the edge to the cloud. Six models were trained in the cloud; three were deployed and tested at the edge with different features by selecting the most important ones using Principal Component Analysis (PCA). Different experiments showed that traffic and processing time decreased significantly based on the time required to predict HAR categories with acceptable accuracy. Since there is significant latency between the edge and the cloud and within the cloud, sending samples for verification save bandwidth, and processing requests locally at the edge speed up predictions. Results illustrate that the time required to serve one request from the environment where smartphones generate traffic through the internet connection to the cloud took about 5.8 s on average, including transmission delays and the prediction process. During this time, the model at the edge can serve 150 requests with the same accuracy using nine features. In addition, the edge can serve 286 requests in 5.8 s with 94.8 % accuracy when choosing the top four features at the edge.

1. Introduction

The Internet of Things (IoT) gives devices a platform to connect to the Internet and other devices, and allows them to gather data about their surroundings. IoT and Artificial Intelligence (AI) support smart systems like smart cities, smart healthcare, smart transportation, and smart energy management systems [1–3]. In recent years, modern devices like smartphones and smartwatches have gained popularity. These devices now have a large number of sensors, including microphones, accelerometers, and gyroscopes, which considerably increase their feature sets. These tools make it possible to create many user-centered applications, including Human Activity Recognition (HAR) systems [4–10].

The field of using machine learning with HAR is rapidly growing, where smartphones and machine learning are significant elements. There are several uses for the capacity to identify and categorize human activity utilizing smartphones and machine learning, including in healthcare, sports, and rehabilitation [11–13]. Also, HAR can be applied for automated observation to predict fall detection of elderly people that may happen [14–16].

Significant research has been done in recent years to create accurate and effective HAR systems employing smartphones and machine learning techniques [17–21]. Many HAR systems were surveyed [11,22–24], the authors focused on several daily life activities in different application domains.

Machine learning is used in the centralized cloud, where resource-intensive infrastructure is located, enabling it to constantly have as much processing, storage, and power as it requires to process data. However, there may be additional waiting time due to network delays from the device to the cloud and vice versa, as well the amount of data in applications with high traffic. This adds additional cost in terms of financial cost and delay [25–27].

* Corresponding author.
E-mail addresses: aymanw@ppu.edu (A.A. Wazwaz), k.amin@ci.menofia.edu.eg (K.M. Amin), noura.samri@ci.menofia.edu.eg (N.A. Semari), tamer.ghanem@ci.menofia.edu.eg (T.F. Ghanem).

https://doi.org/10.1016/j.dajour.2023.100282
Received 13 March 2023; Received in revised form 18 May 2023; Accepted 4 July 2023
Available online 11 July 2023

Fundamental modifications are needed in AI techniques and cloud-to-device design in order to deliver acceptable, efficient, and sustainable solutions for the anticipated future demands. So, there is a need to handle the escalating demand for processing, minimizing delays, and traffic. Cloud service providers began to offer edge solutions, such as Microsoft Azure IoT Edge, which extend their cloud capabilities and place the edge close to the sources of creating data [27].

At the IoT edge, resources are limited, and the applications that run AI systems demand more of these resources, which constrained the AI at the edge. The key issue preventing edge devices from performing to their full potential is the resource gap between those limited edge devices, and the resources required by AI applications. Edge devices typically have very small physical dimensions, processing capability, and power capacity [2,28].

When using cloud environments, HAR system must pass through various steps from smartphones to the cloud server through the internet connection, the prediction process on the server, and then back to the local environment. The proposed solution in this research suggests reducing the amount of data transferred by using fewer features, reducing the amount of overhead traffic processing using the IoT edge, and continuing with periodic server updates using sample values of prediction frequently.

HAR features reduction can be used to improve the efficiency and accuracy of activity recognition models by reducing the number of features used to represent the data. This can help to address issues such as overfitting, improve model performance, and reduce computational requirements [29]. Principal Component Analysis (PCA) is one of the most commonly used feature reduction methods in HAR. This is because PCA is a simple and effective method that can reduce the dimensionality of the data while retaining most of the relevant information [30,31]. PCA provides different strategies for reducing the dimensionality of feature space and preserves the maximum amount of variance of the original data [28]. In this proposed solution, reducing number of features helped in facilitating the prediction processing on the IoT edge where limited resources are used.

In this article, the following sections talk about related literature review, then datasets extraction, the methodology and experiments will be explained, results will be analyzed and compared, and lastly conclusions and future research will be discussed.

2. Literature review

Smart devices are being used in different applications; smartphones were used in [32] to detect human body's motion, the authors proposed HAR using deep learning and smartphone sensor data. The study used deep belief networks to train over the features extracted from smartphone inertial sensors to recognize multiple activities, including running, sitting, sleeping, etc., with transitional activities from one activity to another. The reported outcomes indicate reliable recognition of human activities. In [15], a dedicated cloud-based approach for fall detection and large scale monitoring of older adults. Fall detection deployed on smartphones based on accelerometer and gyroscope measurements collected and processed locally, and the data was transmitted to the cloud for classification purposes and for building a profile of the monitored person, Microsoft Azure was used as a cloud service.

Authors in [33] processed wearable sensors separately at the beginning, learning their features and performing the classification before fusing with the other sensors. They used an approach to extract patterns in multiple temporal scales of the data, using an ensemble of Deep Convolution Neural Networks (DCNN). Author in [22] used unsupervised learning to features from a given dataset remains and using data augmentation techniques, in order to increase the amount of available data for better performance. Wearable systems are designed using intelligent sensors, artificial intelligence, IoT, and big data, in such a way that it is possible to obtain information of interest from the human body as proposed in [23]. Microcontrollers, accelerometer sensors, and smart watches were used to detect human motion signals.

A proposed design and implementation of Convolution Neural Networks (CNN) at the IoT edge is provided in [34]. Authors suggested feature-less activity recognition system, with multi-channel 1-D convolutional neural network architecture, and substituted the manually designed feature extraction procedure in HAR by an automated feature-learning engine. Inference stage was enabled on four layer CNN model using pre-trained and optimized deep learning models on mobile devices using the Tensor-Flow lite. In [35], a two-level approach was proposed; at the first level, four primary physical activities are recognized. At the second level, the corresponding contexts are recognized for each primary activity. The proposed method is based on the accelerometer data and consists of four steps: data acquisition and pre-processing, feature extraction, activity recognition, and context recognition, like talking, in a meeting, shopping, etc.

In internet of healthcare field, authors in [36] proposed a three-dimensional inertia signals of thirteen timestamped human activities such as walking, walking upstairs, walking downstairs, writing, smoking, and others are registered. Here, HAR model is presented based on efficient handcrafted features and Random Forest as a classifier. In [31], researchers proposes an advanced machine learning (ML) approach to HAR systems that includes data collection, data cleaning, feature extraction, feature engineering, and modeling with classification algorithms for predicting human activities. They compare the performance of tree-based boosting algorithms with other traditional ML techniques for identifying human activities using motion sensors from smart devices.

In order to keep up with the needs of applications, AI at the edge is needed to transition from being processed in the cloud to being processed closer to end user devices at the edge [1,27]. IoT edge devices are used in smart systems through offloading tasks from the cloud to the IoT edge, where task processing will be processed locally. AI applications requires a lot of processing capability and consumes energy; and these requirements are normally beyond the capacity of a standalone IoT devices such as smartphones and microcontrollers. In this case, tasks are often offloaded to nearby devices residing between the IoT devices and the cloud [37].

Distributed Machine Learning (ML) at the edge gives computers and smart devices the ability to learn without being explicitly programmed, extracts patterns and dependencies from data, and use them either to gain an understanding of a phenomenon or to predict future outcomes [3,25,38].

The early mentioned publications showed that embedded sensors in smartphones are widely used in machine learning, they were used to recognize different categories of daily life activities. In recent research works, most of the articles focused mainly on enhancing inference accuracy, update algorithms to speed up processing [22,33–36] without real-time testing. In article [15], authors focused on fall detection, and used smartphone app to detect falls without considering real-time performance parameters, like processing time and delay. Accelerometers and other built-in sensors were used in these articles to detect daily activities, movements, falls, and other human monitoring systems. Additional sensors were used from smartphones, or wearable sensors [16,19,39–42] to assure the predicted results.

Table 1 compares between different articles that used smartphones to recognize different activities, the majority of the articles relied on signal analysis on various sensors, with the primary objective being to improve accuracy or accelerate prediction. In this article, the accelerometer coordinates from three smartphones were employed, and the accuracy reached 99.6%. The prediction time was greatly improved without significantly reducing accuracy level (98.8%) by reducing the amount of features and utilizing the IoT edge.

This article used Microsoft Azure cloud systems, smartphones, and an edge device to enhance the performance of human activity recognition using different models. Three smartphones were employed to recognize human activities using built-in accelerometers; the accelerometer values were used as features, and different datasets of features

Table 1
A list of related articles using smartphones in activity recognition.

Reference, year	Sensing method	Activities	Learning method	Accuracy
[15], 2020	Accelerometer and gyroscope	Fall detection	Boosted Decisions Trees	Cloud: 99.2% Edge: 98.2%
[31], 2021	Accelerometer, gyroscope, and magnetometer	Walking, sitting sleeping, standing, exercising	XGBoost, AdaBoost, Boosted C5.0	Sleeping: 96%
[43], 2023	Smartwatch wrist-worn accelerometer	Smoking recognition	CNN, LSTM, BiLSTM	98.6%
[44], 2023	Accelerometer and gyroscope	Walking, driving, inactive, active	Random Forest, XGboost	92.9%
[34], 2019	Accelerometer and gyroscope	Walking, sleeping, standing	CNN	Edge: 96.4%

were used according the importance of each feature. The datasets were extracted from a larger dataset using different sensors to recognize activities. Different experiments were implemented and tested on the IoT edge near the smartphones, in addition to the cloud. Processing time, network delay, and traffic were measured for different models on the edge and the cloud. It was found that the majority of the time was consumed in transmission through the internet connection, compared with the prediction process itself.

3. Methodology and experiments

In this section, the dataset is described, the methodology is explained step by step, and the results of the experiments are analyzed.

3.1. Dataset

Different datasets are available online for HAR systems using smartphones; these datasets were collected by recording activities using different sensors for different people. The dataset in our experiments is based on a dataset generated by Data and Web Science group (DWS) in university of Manheim in Germany [45]; they recorded human activities using smartphones for different people, and using different sensors, like accelerometer, magnetometer, gyroscope, and others. The activities are climbing stairs down and up, jumping, lying, and standing, sitting, running, and walking. They used seven smartphones distributed on chest, forearm, head, shin, thigh, upper arm, and waist.

To simplify this large dataset, we extracted a subset using only three smartphones, and four activities, for one sensor (the accelerometer) that generates three values (x, y, z) from each smartphone. Table 2 present 9 features and one predicted activity, three features from each sensor. The first three columns arm_x, arm_y, and arm_z came from the first smartphone on the arm; the rest six columns from the waist and the shin.

The selected activities are walking, running, standing, and sitting. The positions of the smartphones are shown in Fig. 1; three smartphones are fixed on the arm, the waist, and the shin respectively. Note that the smartphone on the waist is in the horizontal state, which swap the effect of x and y parameters for this accelerometer.

The three positions were chosen upon the nature of categories we are predicting. The chosen categories were not affected clearly by chest or head movements, the forearm movement is already covered by the upper arm movement, and thigh is covered by the shin as well. The three selected smartphones detected clearly the needed activities, and the number of smartphones were reduced to three instead of seven. Regarding sensors, gyroscope and magnetic field sensors, for example, have less impact on detecting walking than the accelerometer sensor, which was chosen in accordance with the four categories. Rotation is detected by a gyroscope, which is not common for walking, running, standing, and sitting.

The dataset contains 120 000 entries, 25% for each recorded category. In the training process at the cloud, 80% of the entries were used for training and the rest were used for validation and testing.

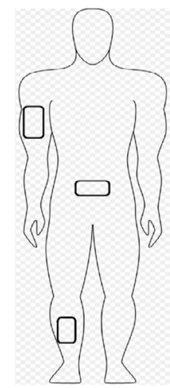

Fig. 1. Three smartphones are placed on the body: the arm, the waist, and the shin [33].

3.2. Methodology

This section outlines the process step-by-step, covering the methodology from extracting datasets to cloud-based training and real-time testing. The experiment were accomplished in three phases: training the models, real time testing for the edge, and real time testing to compare the edge and the cloud to compute the accuracy and the delay. The steps in sequence are:

1. The dataset for three smartphones (Table 2) was extracted from the original dataset [45].
2. The dataset was uploaded on Microsoft azure ML cloud for training and features analysis. Here, Azure tools was used to compute the importance of each feature based on Principal Component Analysis (PCA).
3. Six separate models were created using machine learning in Azure Auto ML to categorize activities; each model has its own subset of the dataset.
4. The IoT edge collect data from smartphones for the real-time tests, and then begin the prediction process after extracting the accelerometer values.
5. The category is predicted in real-time by a trained model, which add the predicted value and accelerometer readings to a JSON message before sending it to the cloud.
6. Telemetry values are provided to the cloud via Azure IoT hub, Azure Stream Analytics (ASA), and Azure machine learning, as shown in Fig. 2.
7. The outcomes of the prediction procedure and the telemetry data are kept in storage containers. Accuracy performance was compared and analyzed at this stage.
8. A web-service is activated to receive requests without going into other services in the cloud; the telemetry data is received by the web-service with machine learning model, which then generates predictions.

Table 2
A sample of the dataset of accelerometers' coordinates for 3 smartphones after features reduction.

arm_x	arm_y	arm_z	waist_x	waist_y	waist_z	shin_x	shin_y	shin_z	Category
−0.02753	9.977836	1.795053	9.941324	−0.42078	0.716465	−0.1227	10.04667	−0.48363	Running
−0.16101	10.0341	1.704672	9.877279	−0.44412	0.846949	−0.13827	10.04128	−0.42018	Running
−0.34058	10.10054	1.669357	9.985018	−0.3711	0.888249	−0.21189	10.05984	−0.48662	Running
−4.66391	8.535928	2.562394	9.944317	0.134075	0.64404	6.265626	7.480683	−2.74615	Sitting
−4.75189	8.607155	2.583942	9.943718	0.155623	0.637456	6.244078	7.475296	−2.73777	Sitting
−4.7495	8.542512	2.61387	9.944916	0.129885	0.612916	6.21834	7.484274	−2.71921	Sitting
−0.39085	10.13765	1.237802	9.932944	0.126294	0.387262	0.013168	10.11431	−0.38666	Standing
−0.37828	10.06463	1.238401	9.971252	0.240019	0.358532	−0.10714	10.06642	−0.44173	Standing
−0.42976	10.00118	1.182736	9.941922	0.172981	0.319626	−0.0808	10.13525	−0.36332	Standing
0.637456	12.07815	1.475427	8.647857	−0.35973	−1.59454	1.092355	7.026383	−4.89135	Walking
1.231218	10.5692	1.936909	6.501455	−1.49219	−0.26875	0.815226	7.100604	−5.06553	Walking
0.994791	8.868123	2.03148	5.975927	0.104148	0.696713	0.764349	7.782352	−4.12521	walking

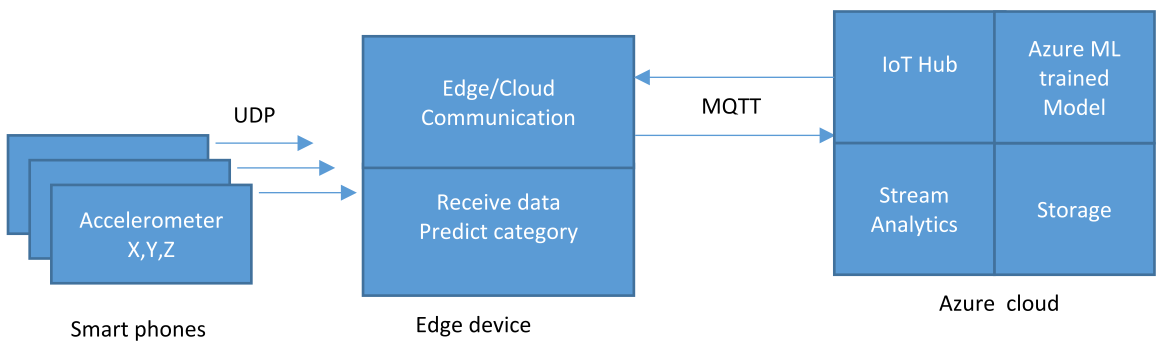

Fig. 2. HAR architecture from the environment to Microsoft Azure Cloud.

9. The results including telemetry, delay measurements, and predictions are sent to the cloud for storage and analysis. Network delay and prediction on the server delay are computed at this stage.

3.3. Experiments and results

Experiments are implemented in three phases training, real-time implementation, and web-service. The three phases are:

• The first phase is training and analysis in Microsoft Azure ML cloud to compute the accuracy, analyze the importance of the features, train the new models with different number of features, and generate the trained models. The generated models are stored to be used later.

• In the second phase, a real-time implementation using three smartphones connected with IoT edge using UDP protocol. This edge receive the values from smartphones' accelerometers, run the hosted prediction model to predict the category, and send the values and the predicted category at the edge to the cloud using MQTT protocol as shown in Fig. 2.

The IoT hub in the cloud receive the telemetry values, and forward those to the stream analytics service (ASA). The machine learning model is implemented on the telemetry data in real time, and store the result in Azure storage containers. Here, the prediction accuracy for different models is measured and compared.

• The third phase was building a web service in Azure cloud to host the prediction model; where a server hosted the machine learning model. Here the messages are sent directly from the edge to the web-service in the cloud, without passing through other cloud services. This reduce the time consumed in the cloud.

3.3.1. HAR model training

The data from the three smartphones' accelerometers was first used with the nine values model (HAR9), then PCA algorithm was used to analyze the features, and find the most related ones that affect the predicted category. Fig. 3 demonstrates how the significance of the features varies. It was noticed that five features were the most important among other features. Two models with top four features (Top4of9), and with top five features (Top5of9) were extracted and analyzed, and three models with one smartphone that create three features datasets, and three models with a combination of features from the three smartphones were among the six models that were employed, each with a different number of feature datasets.

It is clear from Fig. 3 that the (z) components are less important since the direction of the smartphones either in vertical or horizontal states, and none of the observed activities we are predicting depend on the (z) component significantly. Also, the (x) component on the waist (waist_x) is the most important feature, where different categories will affect the behavior of this feature, and the direction of the smartphone is horizontal that swaps the effect of x and y components on the waist.

The six models are:

1. HAR9: all values (9) from three smartphones
2. ARM: (arm_x, arm_y, arm_z) from one smart phone placed on the top of the arm
3. WAIST: (waist_x, waist_y, waist_z) from one smart phone placed on the waist horizontally
4. SHIN: (shin_x, shin_y, shin_z) from one smart phone placed on the shin
5. Top4of9: (waist_x, shin_y, arm_y, shin_x) from three smart phones
6. Top5of9: (waist_x, shin_y, arm_y, shin_x, arm_x) from three smart phones

The outcomes for six models are represented in Fig. 4, then after the least significant features from new datasets were eliminated, accuracy was measured for the different models. We noted from Fig. 4 that prediction accuracy levels for Top4of9 and Top5of9 are close to HAR9 accuracy level, with around half number of features, while using three features from a single smartphone have less accuracy.

Using Azure Auto ML, Azure cloud produced different trained models using different algorithms as shown in Table 3. Some of these algorithms are Light gradient-boosting machine (Light GBM), Extreme

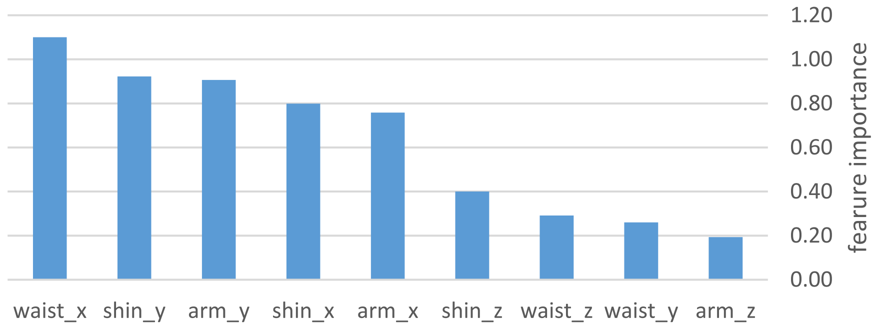

Fig. 3. Feature importance for three smartphones.

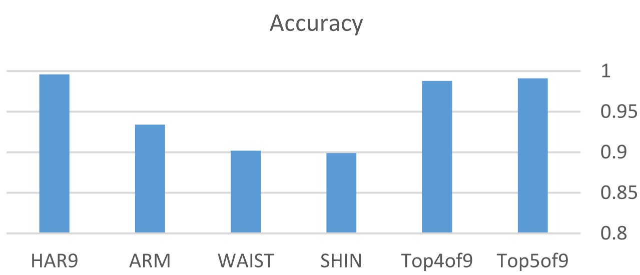

Fig. 4. Accuracy using Light GBM algorithm for six models.

Table 3
Prediction accuracy for HAR9 using different algorithms.

Algorithm	Accuracy
Light GBM	0.99657
XGBoost Classifier	0.98972
Random Forest	0.97741
Extreme Random Tree	0.98519
Logistic Regression	0.75741

Gradient Boosting (XGBoost) Classifier, Random Forest, Gradient Boosting, Extreme Random Trees, and Logistic Regression. Light GBM machine learning algorithm was chosen among these algorithms, because it produced the highest accuracy on the same dataset, then the light GBM model was employed with different number of features.

We selected the most popular machine learning metrics to compare between the different models in HAR training. The metrics are:

- Accuracy: It is defined as total correctly classified requests divided by the total number of classified request.
- AUC accuracy Area Under the Curve ROC (Receiver Operating Characteristics), It tells how much the model is capable of distinguishing between classes. The higher the AUC, the better the model is at distinguishing between categories in the training process.
- Precision: the actual correct prediction divided by total prediction made by a model.
- Recall (Sensitivity): It is a measure of how many of the positive cases the classifier correctly predicted, over all the positive cases in the data.
- F1 score: measure combining both precision and recall, it is generally described as the harmonic mean of the two.

Table 4 shows five metrics for the six experiments using Light GBM algorithm. The table demonstrates that whereas a single smartphone is used, the accuracy ranged between 0.899 and 0.934, HAR9's accuracy when considering all features approached 0.996. The accuracy of the last two models were 0.988 for the Top4of9 model, and 0.991 for the top5of9 model, these two models utilized the most crucial features. Here, it is noticed that increasing the number of features from 4 to 5 will not significantly affect accuracy, as the difference in accuracy between them was less than 0.003. Therefore, it is noted that removing one feature will decrease the amount of data that need to be sent and processed, while maintaining acceptable accuracy.

Table 4
HAR models accuracy.

Metric	HAR9	ARM	WAIST	SHIN	Top4of9	Top5of9
Accuracy	0.996	0.934	0.902	0.899	0.988	0.991
AUC accuracy	0.999	0.991	0.985	0.980	0.999	0.999
Precision	0.996	0.935	0.903	0.899	0.988	0.991
Recall	0.996	0.934	0.902	0.899	0.988	0.991
F1 score	0.996	0.934	0.901	0.899	0.988	0.991

It is also noted from Table 4 that using a single smartphone produced less accurate models, since movements will affect different parts of the body. Because moving the arm during activities is more common, the arm model performed better than the waist and shin models, with an accuracy of 0.93, whereas the accuracy of the other two single smartphone models was about 0.90 for either the waist alone or the shin alone. We used the top three models, HAR9, Top4of9, and Top5of9, in the remaining experiments because of their high accuracy since one sensor will not be able to detect the proper category with the same level of accuracy if two or three sensors are used.

Another parameter to employ in comparing the top three models is the confusion matrix, which measures how frequently the system was confused to determine the correct category. While there is nearly no confusion between sitting and standing in comparison to other activities, there is very little confusion between walking and running, and this confusion decreases as additional features are added as when nine features were used, as seen in Fig. 5.

In Fig. 5 (a and b), it is noticed that the percentage of confusion between predicted labels does not change significantly. For instance, top4of9 had 0.0074 confusion between walking and running, while top5of9 had 0.0067 confusion, where both are near each other. In the sitting and standing categories, there was absolutely little confusion in the three figures, even for 4 and 5 features, since these categories normally do not have frequent movements that affect the accelerometer values. Fig. 5(c) shows that this confusion is getting lesser, and there was almost no confusion in HAR9.

3.3.2. Edge real-time testing

The three models HAR9, Top4of9, and Top5of9 were used in different experiments to measure the performance on the edge, and compare the edge results with the cloud using three smartphones with nine features as a reference. Here, we chose the model with the best accuracy as the standard for all other models. The first experiment used nine values on both the edge and the cloud, followed by the top four values on the edge, and finally the top five values on the edge. The model on the cloud is the same for the three cases which is HAR9. For the three models, light GBM was employed, and the edge was on a PC with Intel i5-powered virtual machine.

Predicted Label

True Label	running	sitting	standing	walking
running	0.9826	0	0.0063	0.0111
sitting	0	0.99	0.0059	0.0041
standing	0	0.0007	0.9978	0.0015
walking	0.0074	0.0011	0.0074	0.9841

(a)

Predicted Label

True Label	running	sitting	standing	walking
running	0.9844	0.0004	0.0052	0.01
sitting	0.0004	0.9944	0.003	0.0022
standing	0	0.0004	0.997	0.0026
walking	0.0067	0.0004	0.0041	0.9889

(b)

Predicted Label

True Label	running	sitting	standing	walking
running	0.9915	0	0.0011	0.0074
sitting	0	0.9989	0.0004	0.0007
standing	0.0004	0	0.9985	0.0011
walking	0.0019	0	0.0007	0.9974

(c)

Fig. 5. Confusion matrix percentage for (a) top 4 0f 9 model, (b) top 5 of 9 model, and (c) HAR9 model.

Table 5 shows the results of three experiments, it shows the number of samples generated from the environment, the accuracy where the predicted category at the edge is the same as the category predicted on the cloud model. We noticed that 100% of the sample produced the same results in HAR9, here the same model was deployed on both on the cloud and the edge. Then comparing HAR9 in the cloud with 4 or 5 selected features, the accuracy was about 94.82% in the case of 4 features were sent to the edge, and 94.92% for the case of 5 features. It is clear that the accuracy in practice produced less accurate results compared with results during training, where the type of the smartphone and the accuracy of accelerometer change from one brand to another, and the number of movements captured affected results as well. In our case, we used different brands of smartphones and different rates of telemetry values.

On the otherhand, for the average processing time at the edge for the three models, Table 4 shows that HAR9 has the highest average prediction time with 38.612 ms per request, but conserve the highest accuracy at the edge. Top4of9 is the lowest in prediction time with 20.359 ms with accuracy over 94.8% accuracy. And top5of9 with 24.817 ms per request, this adds 4.5 ms difference, and the average increment in accuracy is only 0.1% in top5of9 as an advantage over top4of9. The trained models were packaged in pickeled (PKL) files and deployed on the local machine on the IoT edge, Table 5 shows that the prediction models are small in file size, and the difference in file size is about 20 kb.

3.3.3. *Cloud and edge real-time testing*

In this experiment, HAR9 was deployed at the edge, and the same model was deployed at the cloud using a webservice to compare processing time and delay in both cases. Here, The cloud server process requests using a container with a 1 GHz CPU and 1 GB of RAM. Faster processor containers with larger memory will produce faster processing time at the cloud, but the price per hour will increase as well. Table 6 demonstrates that the majority of the delay was consumed inside the cloud and in the network between the edge and the cloud. While processing a request in the cloud depends on the container's configuration. In this configuration, only 2% of the time was consumed in prediction process on the cloud; the remainder is spent in queuing, transmitting inside the cloud, and transmitting between the edge device and the cloud. While in the local environment, the average processing time at the edge was about 0.0386 s.

Here, we noticed that it took more than 5.83 s to process one request on the cloud. In that time, with processing speed of 0.0386 s per request, the system was able to handle 150 requests at the edge, instead of serving one request at the cloud with the same degree of accuracy using HAR9 on both the edge and cloud. When utilizing top4of9 at the edge with an average processing speed of 0.0203 per request, the edge could serve 286 requests with 94.82% accuracy compared to the cloud accuracy with HAR9.

For HAR9, the payload for sending one message in a JSON object with 9 values and one predicted category is 134 bytes. Sending 4220 samples at 25 requests per second will result in sending about 565 Kbytes excluding headers. Sending one request per second to the cloud instead of 25 requests will save 96% of the traffic. In this case, the edge will serve requests locally and one sample per second will be sent for analysis and storage at the cloud.

4. Conclusions

IoT devices produce telemetry data that is used in a variety of real-time applications. Machine learning is used to analyze and forecast results, and cloud computing is used to store data for various applications. Smart systems use these methods to provide services to their customers. IoT systems generate a lot of sensor data from smartphones and other devices, which require intensive computations. High-capability servers are used by cloud systems to process data in real-time, but because the servers are far from the location where the data is generated, it takes time for the data to travel across the internet lines and the servers. Using machine learning models close to the environment at the edge will reduce the amount of traffic transferred to the cloud, resulting in less latency and cost.

HAR systems are used in a variety of scenarios, such as healthcare and rehabilitation, where it is essential to continuously monitor individuals and where HAR systems should be able to act quickly when necessary. As a result, HAR must be quick and accurate, and it must use machine learning at the edge. With its vast capabilities, cloud computing can provide models with a high degree of accuracy, but the traffic and delay between the cloud and the end systems are high. In this work, we suggested adopting IoT edge computing with fewer features in order to speed up processing and reduce the amount of traffic that needs to be sent to the cloud.

By conserving accuracy, saving time, and lowering traffic to and from the cloud, IoT edge computing improves performance. It also helps when the right features and the suitable deployment locations are chosen. The following findings were drawn from this article:

- Using 4 out of 9 features resulted in training accuracy of more than 98 percent, and real-time testing accuracy of more than 94 percent.
- Using more features may add processing time with little or no benefit, as the case of moving from 4 features to 5 features. The time consumed in network transmission and inside the cloud is dominant compared with processing time at the edge or prediction process in the cloud.
- Cloud computing with machine learning may occasionally be utilized as a support to ensure edge processing, as well as to save results for later processing.
- At the edge, upgrading all the capabilities will enhance the performance, some edge devices might have more RAM and CPU power than some cloud-based containers, and then the accuracy and processing time would be better in this case.

Table 5
Real-time edge models performance for 9, 4, and 5 features.

Edge model	Number of samples	Accuracy (%)	Average processing time per request (ms)	PKL File size (kilo bytes)
HAR9	4220	4220/4220 (100%)	38.612	1374
Top4of9	4615	4376/4615 (94.82%)	20.359	1353
Top5of9	4095	3888/4095 (94.92%)	24.817	1359

Table 6
Processing time and network delay at edge and the cloud.

Edge	Cloud		
Local processing time at the edge per request using HAR9 on the edge	Total time from request to response using the cloud per request (100%)	Network delay from device to cloud and back per request (98%)	Processing time in the container at the cloud per request (2%)
0.0386 s	5.8347 s	5.7279 s	0.1068 s

- Feature reduction is application dependent, this technique may result in loss of information, which can negatively impact the accuracy of the model. The reduced feature set may not fully capture the complexity of the original data, which may lead to decreased performance. If this is the case, the edge might be used to speed up processing without sending all queries towards the cloud, edge devices can handle predictions and keep the cloud informed.

By comparing the results with the articles in the literature review, this work produced high accuracy models using one type of sensors, the accuracy approached 99.6% using light GBM if nine features utilized at the edge or the cloud, while the accuracy approached 99.2% at the cloud and 98.2% at the edge in article [15] for fall detection, and 96.4% at the edge in [34] for HAR. In article [31], the accuracy was 96% for sleeping category, and even lesser for the other categories using three type of sensors. In this work, utilizing lighter and faster models clearly increased response time with 94.82% accuracy using less than half the number of features.

There are several restrictions and reasons that led to the accuracy changing from training to real-time, including differences in sensor accuracy between different brands of smartphones, the nature of the body motions, and the number of samples produced over time. The orientations of smartphones on the human body also affect accuracy; improper arrangement will lead to incorrect results.

This research can be extended and deploy AI models on the smartphones, this will enable the smartphones to send less traffic, save time on the edge and the cloud, and run predictions locally. Local predictions might be used in different ways at the edge and the cloud, and facilitate integration of this distributed intelligence between the different parties of the system.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Mohammad Aazam, Sherali Zeadally, Eduardo Feo Flushing, Task offloading in edge computing for machine learning-based smart healthcare, Comput. Netw. (ISSN: 1389-1286) 191 (2021) 108019, http://dx.doi.org/10.1016/j.comnet.2021.108019.

[2] A.M. Ghosh, K. Grolinger, Edge-cloud computing for internet of things data analytics: Embedding intelligence in the edge with deep learning, IEEE Trans. Ind. Inform. 17 (3) (2021) 2191–2200, http://dx.doi.org/10.1109/TII.2020.3008711.

[3] V. Sze, Y.-H. Chen, J. Emer, A. Suleiman, Z. Zhang, Hardware for machine learning: Challenges and opportunities, in: 2018 IEEE Custom Integrated Circuits Conference, CICC, 2018, pp. 1–8, http://dx.doi.org/10.1109/CICC.2018.8357072.

[4] Sakorn Mekruksavanich, Ponnipa Jantawong, Anuchit Jitpattanakul, A deep learning-based model for human activity recognition using biosensors embedded into a smart knee bandage, Procedia Comput. Sci. (ISSN: 1877-0509) 214 (2022) 621–627, http://dx.doi.org/10.1016/j.procs.2022.11.220.

[5] Yi Liu, Weiqing Huang, Shang Jiang, Bobai Zhao, Shuai Wang, Siye Wang, Yanfang Zhang, TransTM: A device-free method based on time-streaming multiscale transformer for human activity recognition, Def. Technol. (ISSN: 2214-9147) (2023) http://dx.doi.org/10.1016/j.dt.2023.02.021.

[6] W. Qi, H. Su, C. Yang, G. Ferrigno, E. De Momi, A. Aliverti, A fast and robust deep convolutional neural networks for complex human activity recognition using smartphone, Sensors 19 (2019) 3731, http://dx.doi.org/10.3390/s19173731.

[7] E. Dhiravidachelvi, M. Kumar, L.D. Vijay Anand, D. Pritima, S. Kadry, et al., Intelligent deep learning enabled human activity recognition for improved medical services, Comput. Syst. Sci. Eng. 44 (2) (2023) 961–977, http://dx.doi.org/10.32604/csse.2023.024612.

[8] Z. Chen, C. Jiang, L. Xie, A novel ensemble ELM for human activity recognition using smartphone sensors, IEEE Trans. Ind. Inform. 15 (5) (2019) 2691–2699, http://dx.doi.org/10.1109/TII.2018.2869843.

[9] Preeti Agarwal, Mansaf Alam, A lightweight deep learning model for human activity recognition on edge devices, Procedia Comput. Sci. (ISSN: 1877-0509) 167 (2020) 2364–2373, http://dx.doi.org/10.1016/j.procs.2020.03.289.

[10] Mansoor Nasir, Khan Muhammad, Amin Ullah, Jamil Ahmad, Sung Wook Baik, Muhammad Sajjad, Enabling automation and edge intelligence over resource constraint IoT devices for smart home, Neurocomputing (ISSN: 0925-2312) 491 (2022) 494–506, http://dx.doi.org/10.1016/j.neucom.2021.04.138.

[11] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, Yunhao Liu, Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities, ACM Comput. Surv. 54 (4) (2021) 77, http://dx.doi.org/10.1145/3447744, (May 2022), 40 pages.

[12] T. Shaik, X. Tao, N. Higgins, L. Li, R. Gururajan, X. Zhou, U.R. Acharya, Remote patient monitoring using artificial intelligence: Current state, applications, and challenges, WIREs Data Min. Knowl. Discov. 13 (2) (2023) e1485, http://dx.doi.org/10.1002/widm.1485.

[13] Kei Tanigaki, Tze Chuin Teoh, Naoya Yoshimura, Takuya Maekawa, Takahiro Hara, Predicting performance improvement of human activity recognition model by additional data collection, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 6 (3) (2022) 142, http://dx.doi.org/10.1145/3550319, (September 2022), 33 pages.

[14] Christian Krupitzer, Timo Sztyler, Janick Edinger, Martin Breitbach, Heiner Stuckenschmidt, Christian Becker, Beyond position-awareness—Extending a self-adaptive fall detection system, Pervasive Mob. Comput. (ISSN: 1574-1192) 58 (2019) http://dx.doi.org/10.1016/j.pmcj.2019.05.007.

[15] Dariusz Mrozek, Anna Koczur, Bozena Małysiak-Mrozek, Fall detection in older adults with mobile IoT devices and machine learning in the cloud and on the edge, Inform. Sci. (ISSN: 0020-0255) 537 (2020) 132–147, http://dx.doi.org/10.1016/j.ins.2020.05.070.

[16] Nancy Gulati, Pankaj Deep Kaur, An argumentation enabled decision making approach for fall activity recognition in social IoT based ambient assisted living systems, Future Gener. Comput. Syst. (ISSN: 0167-739X) 122 (2021) 82–97, http://dx.doi.org/10.1016/j.future.2021.04.005.

[17] Abdul Wasay Sardar, Farman Ullah, Jamshid Bacha, Jebran Khan, Furqan Ali, Sungchang Lee, Mobile sensors based platform of human physical activities recognition for COVID-19 spread minimization, Comput. Biol. Med. (ISSN: 0010-4825) 146 (2022) 105662, http://dx.doi.org/10.1016/j.compbiomed.2022.105662.

[18] A. Zahin, L.T. Tan, R.Q. Hu, Sensor-based human activity recognition for smart healthcare: A semi-supervised machine learning, in: S. Han, L. Ye, W. Meng (Eds.), Artificial Intelligence for Communications and Networks. AICON 2019, in: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 287, Springer, Cham, 2019, http://dx.doi.org/10.1007/978-3-030-22971-9_39.

[19] S. Mekruksavanich, A. Jitpattanakul, Sport-related activity recognition from wearable sensors using bidirectional gru network, Intell. Autom. Soft Comput. 34 (3) (2022) 1907–1925, http://dx.doi.org/10.32604/iasc.2022.027233.

[20] M. Abdel-Basset, H. Hawash, R.K. Chakrabortty, M. Ryan, M. Elhoseny, H. Song, ST-DeepHAR: Deep learning model for human activity recognition in IoHT applications, IEEE Internet Things J. 8 (6) (2021) 4969–4979, http://dx.doi.org/10.1109/JIOT.2020.3033430.

[21] Y. Tang, L. Zhang, H. Wu, J. He, A. Song, Dual-branch interactive networks on multichannel time series for human activity recognition, IEEE J. Biomed. Health Inf. 26 (10) (2022) 5223–5234, http://dx.doi.org/10.1109/JBHI.2022.3193148.

[22] Sen Qiu, Hongkai Zhao, Nan Jiang, Zhelong Wang, Long Liu, Yi An, Hongyu Zhao, Xin Miao, Ruichen Liu, Giancarlo Fortino, Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges, Inf. Fusion (ISSN: 1566-2535) 80 (2022) 241–265, http://dx.doi.org/10.1016/j.inffus.2021.11.006.

[23] Damini Verma, Kshitij R.B. Singh, Amit K. Yadav, Vanya Nayak, Jay Singh, Pratima R. Solanki, Ravindra Pratap Singh, Internet of things (IoT) in nano-integrated wearable biosensor devices for healthcare applications, Biosens. Bioelectron. X (ISSN: 2590-1370) 11 (2022) 100153, http://dx.doi.org/10.1016/j.biosx.2022.100153.

[24] F. Demrozi, G. Pravadelli, A. Bihorac, P. Rashidi, Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey, IEEE Access 8 (2020) 210816–210836, http://dx.doi.org/10.1109/ACCESS.2020.3037715.

[25] Kangyang Chen, Comparative analysis of surface water quality prediction performance and identification of key water parameters using different machine learning models based on big data, BWater Res. (ISSN: 0043-1354) 171 (2020) 115454, http://dx.doi.org/10.1016/j.watres.2019.115454.

[26] W.Y.B. Lim, et al., Federated learning in mobile edge networks: A comprehensive survey, IEEE Commun. Surv. Tutor. 22 (3) (2020) 2031–2063, http://dx.doi.org/10.1109/COMST.2020.2986024.

[27] Christine Mwase, Yi Jin, Tomi Westerlund, Hannu Tenhunen, Zhuo Zou, Communication-efficient distributed AI strategies for the IoT edge, in: Future Generation Computer Systems, Vol. 131, Elsevier, 2022, pp. 292–308, http://dx.doi.org/10.1016/j.future.2022.01.013.

[28] Shreshth Tuli, et al., HealthFog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments, Future Gener. Comput. Syst. (ISSN: 0167-739X) 104 (2020) 187–200, http://dx.doi.org/10.1016/j.future.2019.10.043.

[29] Shaeela Ayesha, Muhammad Kashif Hanif, Ramzan Talib, Overview and comparative study of dimensionality reduction techniques for high dimensional data, Inf. Fusion (ISSN: 1566-2535) 59 (2020) 44–58, http://dx.doi.org/10.1016/j.inffus.2020.01.005.

[30] Hossein Shafizadeh-Moghadam, Fully component selection: An efficient combination of feature selection and principal component analysis to increase model performance, Expert Syst. Appl. (ISSN: 0957-4174) 186 (2021) 115678, http://dx.doi.org/10.1016/j.eswa.2021.115678.

[31] Pratik Tarafdar, Indranil Bose, Recognition of human activities for wellness management using a smartphone and a smartwatch: A boosting approach, Decis. Support Syst. (ISSN: 0167-9236) 140 (2021) http://dx.doi.org/10.1016/j.dss.2020.113426.

[32] Mohammed Mehedi Hassan, Md. Zia Uddin, Amr Mohamed, Ahmad Almogren, A robust human activity recognition system using smartphone sensors and deep learning, Future Gener. Comput. Syst. (ISSN: 0167-739X) 81 (2018) 307–313, http://dx.doi.org/10.1016/j.future.2017.11.029.

[33] Jessica Sena, Jesimon Barreto, Carlos Caetano, Guilherme Cramer, William Robson Schwartz, Human activity recognition based on smartphone and wearable sensors using multiscale DCNN ensemble, Neurocomputing (ISSN: 0925-2312) 444 (2021) 226–243, http://dx.doi.org/10.1016/j.neucom.2020.04.151.

[34] T. Zebin, P.J. Scully, N. Peek, A.J. Casson, K.B. Ozanyan, Design and implementation of a convolutional neural network on an edge computing smartphone for human activity recognition, IEEE Access 7 (2019) 133509–133520, http://dx.doi.org/10.1109/ACCESS.2019.2941836.

[35] Muhammad Ehatisham-ul Haq, Muhammad Awais Azam, Yusra Asim, Yasar Amin, Usman Naeem, Asra Khalid, Using smartphone accelerometer for human physical activity and context recognition in-the-wild, Procedia Comput. Sci. (ISSN: 1877-0509) 177 (2020) 24–31, http://dx.doi.org/10.1016/j.procs.2020.10.007.

[36] Mohamed E. Issa, Ahmed M. Helmi, Mohammed A.A. Al-Qaness, Abdelghani Dahou, Mohamed Abd Elaziz, Robertas Damaševičius, Human activity recognition based on embedded sensor data fusion for the internet of healthcare things, Healthcare 10 (6) (2022) 1084, http://dx.doi.org/10.3390/healthcare10061084.

[37] S. Wang, J. Xu, N. Zhang, Y. Liu, A survey on service migration in mobile edge computing, IEEE Access 6 (2018) 23511–23528, http://dx.doi.org/10.1109/ACCESS.2018.2828102.

[38] Talha Burak Alakus, Ibrahim Turkoglu, Comparison of deep learning approaches to predict COVID-19 infection, Chaos Solitons Fractals (ISSN: 0960-0779) 140 (2020) 110120, http://dx.doi.org/10.1016/j.chaos.2020.110120.

[39] V. Bianchi, M. Bassoli, G. Lombardo, P. Fornacciari, M. Mordonini, I. De Munari, IoT wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment, IEEE Internet Things J. 6 (5) (2019) 8553–8562, http://dx.doi.org/10.1109/JIOT.2019.2920283.

[40] Shruthi K. Hiremath, Thomas Plötz, Deriving effective human activity recognition systems through objective task complexity assessment, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 4 (4) (2020) 131, http://dx.doi.org/10.1145/3432227, (December 2020), 24 pages.

[41] Zhen Hong, Miao Hong, Ning Wang, Yong Ma, Xiaolong Zhou, Wei Wang, A wearable-based posture recognition system with AI-assisted approach for healthcare IoT, Future Gener. Comput. Syst. (ISSN: 0167-739X) 127 (2022) 286–296, http://dx.doi.org/10.1016/j.future.2021.08.030.

[42] Emanuele Lattanzi, Valerio Freschi, Evaluation of human standing balance using wearable inertial sensors: A machine learning approach, Eng. Appl. Artif. Intell. (ISSN: 0952-1976) 94 (2020) 103812, http://dx.doi.org/10.1016/j.engappai.2020.103812.

[43] N. Hnoohom, S. Mekruksavanich, A. Jitpattanakul, An efficient resnetse architecture for smoking activity recognition from smartwatch, Intell. Autom. Soft Comput. 35 (2023) 1245–1259, http://dx.doi.org/10.32604/iasc.2023.028290.

[44] Daniel Garcia-Gonzalez, Daniel Rivero, Enrique Fernandez-Blanco, Miguel R. Luaces, New machine learning approaches for real-life human activity recognition using smartphone sensor-based data, Knowl.-Based Syst. (ISSN: 0950-7051) 262 (2023) http://dx.doi.org/10.1016/j.knosys.2023.110260.

[45] Germany - Research Group Data and Web Science, DataSet - RealWorld, HAR, Mannheim University, https://sensor.informatik.uni-mannheim.de/#dataset_realworld.