JOURNAL LA MULTIAPP

VOL. 03, ISSUE 06 (276-286), 2022
DOI: 10.37899/journallamultiapp.v3i6.755

Human Activity Recognition Using Visibility Graph Features Coupled with Machine Learning Algorithm

Huda Jalil Ibrahim[1], Methaq Taleb Kata[1], Atheer Yousif Oudah[2]

[1]Department of Computer Science, Collage of Sciences, Mustansiriyah University, Iraq
[2]Computer Engineering Technology Department Al-Ayen University, Iraq

**Corresponding Author:*
Email:

Article Info

Article history:
Received 9 January 2023
Received in revised form 14 February 2023
Accepted 23 February 2023

Keywords:
Human Activity Recognition
Machine Learning
Visibility Graph
Least-Squares Support Vector Machine

Abstract

Human activities refer to the actions and behaviors of human beings. These activities can be physical, such as working, playing sports, or playing sports; or mentally, such as learning, problem-solving, or decision-making. Technical development and the emergence of mobile devices such as phones and smart watches, as well as wearable sensors, led to the emergence of many systems to recognize and classify human activities. These systems were developed using the data collected by these devices from a variety of individuals who volunteered to do several activities, such as downstairs, upstairs, sitting, running, standing, and more. Using the WISDM (Wireless Sensor Data Mining) dataset, a new machine learning model is proposed to recognize six different human activities: walking, jogging, going up stairs, going down stairs, sitting, and standing. For signal segmentation, the sliding window technique was used, along with two visibility graph techniques for feature extraction: mean degree and Jaccard coefficient. The Least-Squares Support Vector Machines (LS-SVM) used to classify these activities This model achieves 94% accuracy, demonstrating that the proposed model has a high classification rate.

Introduction

Human Activity Recognition has become a very important field in computer science to classify the activities of humans by using different strategies for recognition of a wide range of physical activities that humans perform (Gupta, 2021). Furthermore, there has been an increase in interest and research into how to create a HAR system capable of continuously discovering and learning different types of activities. Where these systems were used as a means of obtaining information about human behavior, to obtain this information, sensors such as peripheral and wearable sensors are used (Lytras, 2018). Most of the studies are interested in using built-in sensors to improve performance and achieve high accuracy in the classification process. Certain techniques are then used to process and extract features, and then these features are classified by artificial intelligence systems such as machine learning and deep learning (Pareek & Thakkar, 2021).

HAR systems can be applied in wide fields such as healthcare systems, for example, monitoring the patients' health status for rapid intervention and defining the right strategy at the right time, and fall detection of elderly people (Saeed et al., 2022). It can also be applied to monitoring public places by video monitoring and identifying any suspicious movement (Tripathi et al., 2018).

The activities include various movements, including running, walking, sitting, lying down, exercising, cooking, etc., and their division depends on the duration and complexity of each movement into short movements, such as the transition from stand to sit, simple movements such as speaking while sewing, and complex movements that include interaction with others (Mohamed et al., 2022). The accuracy of the classification of these activities is based on the sensors and algorithms used.

The rest of the paper demonstrates some important techniques that were used to extract and classify features, and then compares these algorithms with the accuracy obtained from each study. This paper deals with graph features and we note that the result of this method is satisfactory when compared to other methods.

Related Works

The 12 human activities proposed by Hassan et al. (2018) for recognition using a smartphone system include standing, sitting, lying down, walking, walking-upstairs, walking-downstairs, stand-to-sit, sit-to-stand, sit-to-lie, and lie-to-sit. The dataset was collected using smartphone sensors and then used to train and test the system. Linear Discriminant Analysis was used to analyze the features extracted by Kernel Principal Component Analysis. The retrieved features were classified using a deep belief network, which has a 95.85% accuracy.

Jia & Chen (2020) used UT-Interaction and CAVIAR datasets with 12 composite activities such as hand-shaking, hugging, kicking, pointing, punching, pushing, entering and exiting shops, walking, meeting, fighting, and passing out to propose a framework–based HAR system based on hierarchical activity structure to formalize hierarchical activity structure into formal syntactical logical formulas and rules.

Sikder & Nahid (2021) created the KU-HAR dataset, which includes 18 activities such as (stand, sit, talk-sit, talk-stand, stand-sit, lay, lay-stand, pick, jump, and push-up. Sit-up, walk, backward walk, walk-circle, run, stair-up, stair-down, table tennis). It was collected using the smartphone's built-in sensors (accelerometer and gyroscope). For classification, a Random Forest (RF) classifier was used, which resulted an accuracy of 89.67%.

The Deep Gated Recurrent Unit (DGRU) was used by Bokhari et al. (2021) to classify seven activities: running, sitting, walking, jumping, lying, falling, and NA. The datasets were collected using Channel State Information (CSI), while discrete wavelet transforms and linear discriminant analysis were used for features extraction and selection. The proposed system attained an accuracy of 98.12%.

Liu et al. (2021) they used five data sets, three of which contain various activities collected using various sensors, and two of which are citation network. MRDGCN was chosen because of its ability to extract and classify features effectively.

Jung & Chi, (2020) suggest a model based on sound recognition that can classify 10 types of daily human activities. The open-source online video and audio platforms Youtube-8M and AudioSet were used to collect sound datasets. The features of sound data are extracted using log Mel-band energies. These activities were classified using a neural network architecture (RNN), which achieved 87.2% accuracy.

The dataset ARAS used by Jethanandani et al. (2020) consisted of two real-world houses with multiple residents. The dataset was collected based on the activities of these residents in the smart homes. Sensors of various types are used to record activities in both smart homes. For classifying activities, Bernoulli-Naive Bayes, Decision Trees, Logistic Regression, and K-Nearest Neighbor were used. The features It was extracted using label encoding. The proposed system achieved an accuracy of 92%.

Jain & Kanhangad, (2017) designed a new model for HAR based on a two-descriptor centroid signature-based Fourier system with a gradient histogram to extract features. A multiclass SVM and a k-nearest classifier were also used to classify six activities such as standing, sitting, laying, walking, downstairs, and upstairs. The proposed approach was evaluated on two publicly available datasets, namely the UCI HAR dataset and physical activity sensor data collected with a smartphone. The achieved accuracy was 97.12%.

Jalloul et al. (2017) employed a complex network approach to HAR. The data was collected using wearable sensors comprised of six IMUs placed on the body at the following locations: ankle, thigh, wrist, arm, hip, and neck. Each IMU was outfitted with three channel measurements. Each of the two sensors' channels was correlated. The correlation was then represented using a connectivity matrix. These measurements were sent into the Random Forest classifier, which was used to identify four activities: walking, standing, lying, and sitting. Overall, the accuracy was 84.6%.

Dataset

The WISDM (Wireless Sensor Data Mining) dataset is a collection of human activity data collected from wireless accelerometers worn by study participants. The dataset was collected as part of a research project on human activity recognition using wearable sensors, and it contains data from over 50,000 instances of six different activities: walking, jogging, going up stairs, going down stairs, sitting, and standing.

The WISDM dataset includes data on the acceleration and angular velocity of the accelerometers in three dimensions (x, y, and z), as well as metadata on the participants, such as age and gender. It is a widely used dataset in the field of human activity recognition, and it has been used to evaluate and compare different machine learning algorithms for activity recognition.

The WISDM dataset is freely available for researchers to download and use for their own studies. It is a useful resource for researchers who are interested in developing machine learning algorithms for human activity recognition, or who are studying the use of wearable sensors in health and wellness applications. Table 1. shows the proportions of each human activity in the data set.

Table 1. Summary of WISDM dataset

Activities	No. Samples	Percentage
Walking	424,400	38.6%
Jogging	342,177	31.2%
Sitting	59,939	5.5%
Standing	48,395	4.4%
Upstairs	122,869	11.2%
Downstairs	100,427	9.1%

Signal Segmentation Techniques

Signal segmentation is the process of dividing a continuous signal into distinct segments or sections Bhat et al. (2018) signal segmentation techniques are used to identify different parts or features of a signal, and they are important in a variety of applications, including signal processing, speech recognition, and biomedical signal analysis.

There are several different techniques that can be used for signal segmentation, including Almotiri et al (2018); (1) Thresholding: Thresholding involves dividing the signal into different segments based on the value of the signal at a particular point in time. For example, a threshold might be set to separate the signal into two segments based on whether the signal is above or below a certain level; (2) Feature extraction: Feature extraction algorithms identify specific features or patterns in the signal, such as peaks or valleys, and use these features to segment the signal into different sections; (3) Clustering: Clustering algorithms group similar sections of the signal into different clusters based on their similarity in terms of the signal's amplitude, frequency, or other characteristics; (4) Dynamic programming: Dynamic programming algorithms use an optimization criterion to divide the signal into segments that are locally optimal in some sense; (5) Machine learning: Machine learning algorithms, such as support vector machines or decision trees, can be trained to classify different sections of the signal based on their characteristics.

Overall, signal segmentation is an important step in signal processing and analysis, and the choice of segmentation technique depends on the specific requirements and characteristics of the signal being segmented.

Table 2. Related works summary

Author	**Dataset**	**No. Activities**	**Features Extraction Methods**	**Classification Methods**	**Accuracy**
Hassan et al. [7]	Collected dataset	12	KPC Analysis	Deep Belief Network	95.85%
Jia et al. [8]	UT, CAVIAR	12	-	-	-
Sikder et al [9]	KU-HAR	18	-	RF	89.67%
Bokhari et al. [10]	Collected dataset	7	DWT, LDA	DGRU	98.12%
Liu et al. [11]	Multiple datasets	-	MRDGCN	MRDGCN	-
Jung et al. [12]	Collected dataset	10	log Mel-band	RNN	87.2%
Jethanandani, Manan et al. [13]	ARAS	27	Label encoding	BNB, DT, LR, and KNN	92%
Jain et al. [14]	UCI HAR	6	Fourier Descriptor centroid with a gradient histogram	SVM, KNN	97.12%
Jalloul, Nahed, et al. [15]	Collected dataset	4	-	RF	84.6%

Visibility Graph

A visibility graph is a mathematical construct used to represent the visibility of objects in a scene or environment. It is a graph data structure in which the nodes represent the objects in the scene, and the edges represent the visibility relationships between the objects (Du & Tang, 2019).

In a visibility graph, an edge exists between two objects if and only if one object is visible from the other (Feng et al., 2021). This means that if object A can see object B, and object B can see object A, there will be an edge connecting the two objects in the graph. The visibility graph is a useful tool for analyzing the visibility relationships between objects in a scene, and it has applications in computer graphics, robotics, and other fields.

Mean Degree

In a visibility graph, the mean degree refers to the average number of connections (edges) that a node (object) has to other nodes in the graph (Supriya et al., 2016). It is calculated by dividing the total number of edges in the graph by the total number of nodes (Eq. 1) (Supriya et al., 2016). For example, consider a visibility graph with 10 nodes and 20 edges. The mean degree of the graph would be 2, since each node has an average of 2 edges connecting it to other nodes in the graph.

$$\bar{\mathrm{d}} = \frac{1}{n}\sum_{i-1}^{n} Dn_i \qquad (\mathbf{1})$$

The mean degree of a visibility graph can provide important information about the connectivity of the objects in the scene. A high mean degree indicates that the objects in the scene are generally well-connected and have good visibility of each other, while a low mean degree indicates that the objects are less connected and may have less visibility of each other (Supriya et al., 2018).

The mean degree of a visibility graph can also be used to analyze the efficiency of a robot's path through a scene, or to identify areas where a person might have a clear view of the surrounding environment. It can also be used to optimize the layout of objects in a scene to improve visibility and connectivity.

Jaccard Coefficient

The Jaccard coefficient is a measure of similarity between two sets. It is defined as the size of the intersection of the two sets divided by the size of the union of the two sets. In a visibility graph, the Jaccard coefficient can be used to measure the similarity between two nodes (objects) in terms of their visibility relationships with other nodes in the graph (Ceballos-Escalera et al., 2020).

To calculate the Jaccard coefficient for two nodes in a visibility graph (Eq. 2) (Ceballos-Escalera et al., 2020), first, the intersection of the two nodes is calculated by identifying the nodes that are visible from both nodes. The size of this intersection is then divided by the size of the union of the two nodes, which is the total number of nodes that are visible from either of the two nodes.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \qquad (\mathbf{2})$$

For example, consider a visibility graph with four nodes (A, B, C, and D) and the following visibility relationships:

- Node A is visible from nodes B and C.
- Node B is visible from nodes A and C.
- Node C is visible from nodes A, B, and D.
- Node D is visible from node C.

The Jaccard coefficient for nodes A and B can be calculated as follows:

- The intersection of A and B is {A, B, C}, which has a size of 3.
- The union of A and B is {A, B, C, D}, which has a size of 4.
- The Jaccard coefficient for A and B is 3/4, or 0.75.

The Jaccard coefficient can be used to analyze the similarity of objects in a scene based on their visibility relationships with other objects. It can also be used to identify clusters of objects in a scene that have similar visibility patterns.

Least-Squares Support Vector Machines

Least-Squares Support Vector Machines (LS-SVMs) are a type of machine learning algorithm that can be used to classify data into different categories. LS-SVMs are a variant of Support Vector Machines (SVMs), which are a type of linear classifier that uses a hyperplane to separate different categories of data (Ceballos-Escalera et al., 2020).

LS-SVMs differ from traditional SVMs in that they use a least-squares optimization criterion to find the optimal hyperplane. This allows LS-SVMs to handle non-linearly separable data and to handle cases where the data may be noisy or have outliers (Asare et al., 2017).

To classify data using an LS-SVM, the algorithm first divides the data into different categories based on their labels. It then determines the optimal hyperplane that best separates the data into these categories. Once the hyperplane is determined, the LS-SVM can be used to classify new data points based on which side of the hyperplane they fall on (Asare et al., 2017).

LS-SVMs are often used in applications where it is important to accurately classify data, such as in image and speech recognition, text classification, and fraud detection. They are also useful in situations where the data may be noisy or have outliers, as they are able to handle these cases more effectively than traditional SVMs (Nkengfack et al., 2021).

The steps involved in training and using an LS-SVM are as follows:

Step 1. Collect and preprocess the data: The first step in training an LS-SVM is to collect a set of labeled training data. The data should be cleaned and preprocessed to remove any noise or errors, and to ensure that it is in a suitable format for the LS-SVM to use.

Step 2. Choose the kernel function: The kernel function is a mathematical function that is used to transform the data into a higher-dimensional space, where it may be more easily separated into different categories. The choice of kernel function depends on the characteristics of the data and the specific classification task.

Step 3. Train the LS-SVM: The LS-SVM is trained by finding the hyperplane that best separates the data into the different categories. This is done using a least-squares optimization criterion, which allows the LS-SVM to handle non-linearly separable data and to handle cases where the data may be noisy or have outliers.

Step 4. Test the LS-SVM: Once the LS-SVM has been trained, it can be tested on a set of labeled test data to evaluate its performance. The accuracy of the LS-SVM's predictions can be measured using a variety of metrics, such as precision, recall, and F1 score.

Step 5. Use the LS-SVM: Once the LS-SVM has been trained and tested, it can be used to classify new data points based on which side of the hyperplane they fall on. The LS-SVM can be used to classify data in real-time as it is received, or it can be used to classify a batch of data all at once.

Overall, training and using an LS-SVM involves several steps, including collecting and preprocessing the data, choosing a kernel function, training the LS-SVM, testing its performance, and using it to classify new data.

Proposed System

In this paper, we presented a new model for human activity recognition based on visibility graph coupled with LS-SVM. Fig.1 shows the proposed model for human activity classification.

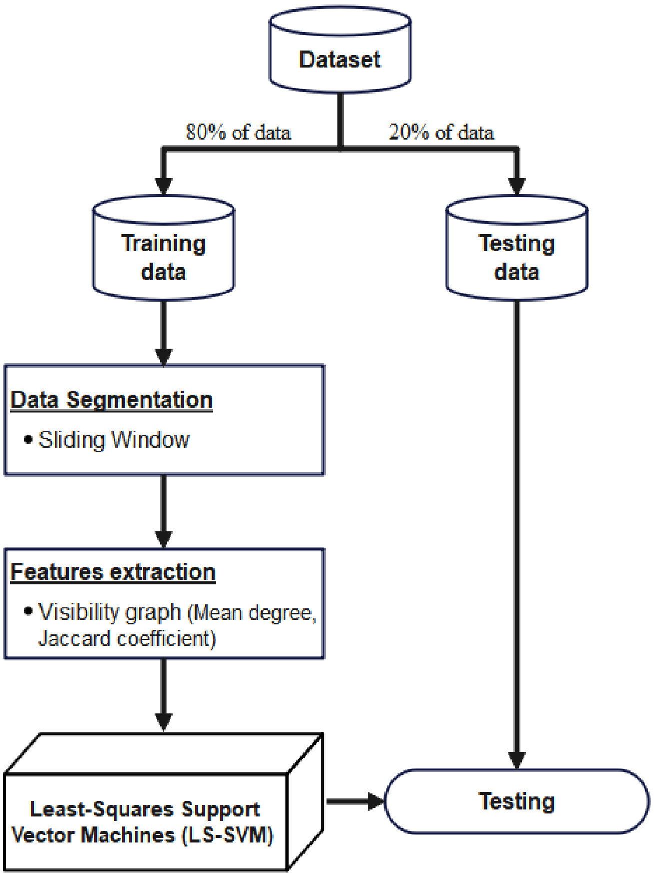

Figure 1. Proposed model Block diagram

Data Distribution

The WISDM dataset contains data from over 50,000 instances of six different activities: walking, jogging, going up stairs, going down stairs, sitting, and standing. The overall number of records is 1,098,207. These records are divided into two parts: 80% for training total (878,566) records and 20% for testing total (219,641) records.

Data Segmentation

Sliding window segmentation is a technique for dividing a signal or sequence into overlapping segments or windows. It involves moving a window of fixed size over the signal, and extracting a segment from the signal at each position of the window.

Sliding window segmentation is often used in signal processing and machine learning applications, where it can be used to analyze the characteristics of the signal within each segment or window. For example, it can be used to identify patterns or features within the signal, or to classify the signal based on its characteristics.

To implement sliding window segmentation, the following steps are typically followed:

Step 1. Define the size of the window (Window = 15) and the overlap between consecutive windows (Overlap = 50%).

Step 2. Initialize the starting (position=1) of the window.

Step 3. Extract the segment of the signal within the window.

Step 4. Extract visibility graph features from segment.

Step 5. Slide the window to the next position by a certain amount, which is determined by the overlap between consecutive windows.

Step 6. Repeat steps 3-5 until the end of the signal is reached.

Features Extraction

The visibility graph technique was used to map each segment of HAR data in a time series into a graph by representing each data point as a node and connecting any two nodes by edge. Then the adjacency matrix was represented for each graph. As shown in Figure 2. two features were extracted from a graph. The first feature is *mean degree* (the number of edges that are represented for each node). The second feature is the *Jaccard coefficient* (a node is linked with a number of neighbors). Then, the features were sent to the LS-SVM for classification.

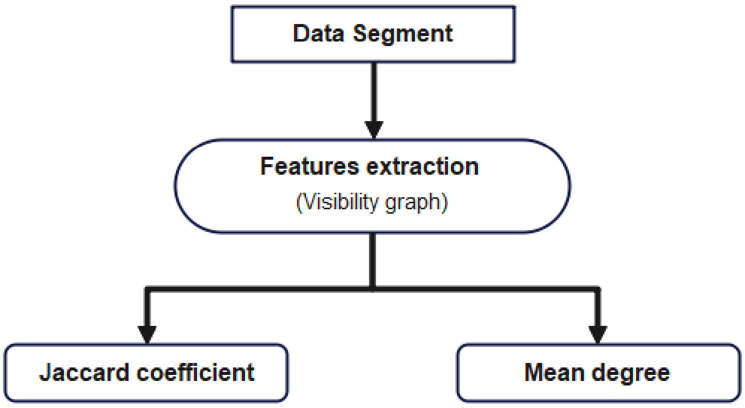

Figure 2. Graph features extraction

To extract the mean degree feature from a graph, the following steps can be followed:

Step 1. Identify the nodes and edges in the graph: The first step is to identify the nodes and edges in the graph. The nodes represent the objects or entities in the graph, and the edges represent the relationships between these objects.

Step 2. Calculate the degree of each node: The degree of a node is the number of edges that are connected to it.

Step 3. um the degrees of all the nodes: Next, sum the degrees of all the nodes in the graph. This will give the total number of edges in the graph.

Step 4. Divide the total number of edges by the number of nodes: Finally, divide the total number of edges by the number of nodes in the graph to calculate the mean degree. This will give the average number of edges that each node has in the graph.

To extract the Jaccard coefficient feature from a graph, the following steps can be followed:

Step 1. Identify the nodes and edges in the graph: The first step is to identify the nodes and edges in the graph. The nodes represent the objects or entities in the graph, and the edges represent the relationships between these objects.

Step 2. Select two nodes: Next, select two nodes in the graph for which you want to calculate the Jaccard coefficient.

Step 3. Identify the nodes that are visible from both nodes: The intersection of the two nodes is the set of nodes that are visible from both of the selected nodes. To find this intersection, you can iterate over the edges connected to each node and record the nodes that are connected to both nodes.

Step 4. Calculate the size of the intersection and the size of the union: The size of the intersection is the number of nodes that are visible from both of the selected nodes. The size of the union is the total number of nodes that are visible from either of the two selected nodes.

Step 5. Calculate the Jaccard coefficient: Finally, calculate the Jaccard coefficient by dividing the size of the intersection by the size of the union. The Jaccard coefficient will be a value between 0 and 1, with higher values indicating greater similarity between the two nodes.

Training LS-SVM

Due of the many properties of the algorithm and the great accuracy of the classification process. The LS-SVM technique was employed as the primary classifier of human activities in this work. We tried three kernels: a linear kernel, a polynomial kernel, and a radial basis function (RBF) kernel. The RBF kernel produced the best results.

In a Least-Squares Support Vector Machine (LS-SVM), the parameters γ and σ are used in the radial basis function (RBF) kernel, which is a type of kernel function that can be used to classify non-linearly separable data.

The parameter γ determines the width of the RBF kernel, which controls the influence of each training example on the decision boundary. A larger value of γ results in a narrower kernel, which means that each training example has a greater influence on the decision boundary. A smaller value of γ results in a wider kernel, which means that each training example has a smaller influence on the decision boundary.

The parameter σ is a measure of the spread or variance of the RBF kernel. It determines how smooth the decision boundary will be, with a larger value of σ resulting in a smoother boundary and a smaller value of σ resulting in a more jagged boundary.

Both γ and σ are hyperparameters, which means that they are not directly learned from the training data, but rather are set manually. The values of γ and σ can have a significant impact on the performance of the LS-SVM, and they are often chosen through trial and error or through the use of a grid search or other optimization method.

To achieve high classification accuracy, two essential parameters γ and σ, should be carefully chosen in the LS-SVM. These two parameters may have an impact on classifier performance. Several experiments were carried out in this research to determine the best value for each parameter. As a result, the two parameters were set to $\gamma = 1$ and $\sigma = 1$ respectively.

Results and Discussion

Using (219,641) records of testing data, the performance of the proposed model was assessed in terms of accuracy and confusion matrix. Table 3 reports the confusion matrix results of the proposed model.

Table 3. Confusion matrix for the proposed model

Predict/ground truth	Downstairs	Jogging	Walking	Upstairs	Sitting	Standing
Standing	0.00	0.00	0.01	0.00	0.00	**0.94**
Sitting	0.00	0.01	0.00	0.03	**0.94**	0.00
Upstairs	0.06	0.00	0.00	**0.94**	0.00	0.05
Walking	0.01	0.00	**0.94**	0.02	0.03	0.00
Jogging	0.00	**0.93**	0.01	0.00	0.01	0.00
Downstairs	**0.94**	0.06	0.04	0.04	0.02	0.01

Six activities were classified in this experiment. The proposed model obtained an average of accuracy of 94%.

After obtained the results of our proposal HAR model, we found some of limitations that make difficult to create system capable of accurately distinguishing and classifying human activities, this reason come from; (1) Lack of comprehensive dataset to carries out all human activities;

(2) Doing the activity varies from one person to another. Different sensors used to collect data on human activities.

Conclusion

Despite the significant effort made for developing HAR system, several issues have not been addressed for example, the relationship between features type and classification rate. In this paper, we completely evaluate the type of features that extracted by using a type of a graph on HAR system. This study was based on visibility graph to extract mean degree and Jaccard coefficient features from WISDM dataset. The results showed that the graph features gave a high classification rate. Further experiments are needed to test the proposed model with a larger dataset that could make it more robust. In addition, further classifiers are needed to be tested in HAR systems. Finally, the proposed HAR model achieve 94% accuracy in classifying Six human activities.

Acknowledgment

The authors thank the Department of Computer Science / College of Science, Al-Mustansiryia University, for supporting this work.

References

Almotiri, J., Elleithy, K., & Elleithy, A. (2018). Retinal vessels segmentation techniques and algorithms: a survey. *Applied Sciences*, *8*(2), 155.

Asare, A. T., Mensah, F., Acheampong, S., Asare-Bediako, E., & Armah, J. (2017). Effects of gamma irradiation on agromorphological characteristics of okra (Abelmoschus esculentus L. Moench.). *Advances in Agriculture*, *2017*.

Bhat, G., Deb, R., Chaurasia, V. V., Shill, H., & Ogras, U. Y. (2018, November). Online human activity recognition using low-power wearable devices. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (pp. 1-8). ACM.

Bokhari, S. M., Sohaib, S., Khan, A. R., & Shafi, M. (2021). DGRU based human activity recognition using channel state information. *Measurement*, *167*, 108245.

Ceballos-Escalera, A., Molognoni, D., Bosch-Jimenez, P., Shahparasti, M., Bouchakour, S., Luna, A., ... & Della Pirriera, M. (2020). Bioelectrochemical systems for energy storage: A scaled-up power-to-gas approach. *Applied energy*, *260*, 114138.

Ceballos-Escalera, A., Molognoni, D., Bosch-Jimenez, P., Shahparasti, M., Bouchakour, S., Luna, A., ... & Della Pirriera, M. (2020). Bioelectrochemical systems for energy storage: A scaled-up power-to-gas approach. *Applied energy*, *260*, 114138.

Du, C., & Tang, B. (2019). Novel unconventional-active-jamming recognition method for wideband radars based on visibility graphs. *Sensors*, *19*(10), 2344.

Feng, Q., Wei, H., Hu, J., Xu, W., Li, F., Lv, P., & Wu, P. (2021). Analysis of the attention to COVID-19 epidemic based on visibility graph network. *Modern Physics Letters B*, *35*(19), 2150316.

Gupta, S. (2021). Deep learning based human activity recognition (HAR) using wearable sensor data. *International Journal of Information Management Data Insights*, *1*(2), 100046.

Hassan, M. M., Uddin, M. Z., Mohamed, A., & Almogren, A. (2018). A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, *81*, 307-313.

Jain, A., & Kanhangad, V. (2017). Human activity classification in smartphones using accelerometer and gyroscope sensors. *IEEE Sensors Journal*, *18*(3), 1169-1177.

Jalloul, N., Porée, F., Viardot, G., L'Hostis, P., & Carrault, G. (2017). Activity recognition using complex network analysis. *IEEE journal of biomedical and health informatics*, *22*(4), 989-1000.

Jethanandani, M., Sharma, A., Perumal, T., & Chang, J. R. (2020). Multi-label classification based ensemble learning for human activity recognition in smart home. *Internet of Things*, *12*, 100324.

Jia, H., & Chen, S. (2020). Integrated data and knowledge driven methodology for human activity recognition. *Information Sciences*, *536*, 409-430.

Jung, M., & Chi, S. (2020). Human activity classification based on sound recognition and residual convolutional neural network. *Automation in Construction*, *114*, 103177.

Liu, W., Fu, S., Zhou, Y., Zha, Z. J., & Nie, L. (2021). Human activity recognition by manifold regularization based dynamic graph convolutional networks. *Neurocomputing*, *444*, 217-225.

Lytras, M. (2018). Visvizi A Daniela L Sarirete A Ordonez De Pablos P. *Social networks research for sustainable smart education Sustainability*, *10*(9), 2974.

Mohamed, M., El-Kilany, A., & El-Tazi, N. (2022). Future Activities Prediction Framework in Smart Homes Environment. *IEEE Access*, *10*, 85154-85169.

Nkengfack, L. C. D., Tchiotsop, D., Atangana, R., Louis-Door, V., & Wolf, D. (2021). Classification of EEG signals for epileptic seizures detection and eye states identification using Jacobi polynomial transforms-based measures of complexity and least-square support vector machine. *Informatics in Medicine Unlocked*, *23*, 100536.

Pareek, P., & Thakkar, A. (2021). A survey on video-based human action recognition: recent updates, datasets, challenges, and applications. *Artificial Intelligence Review*, *54*, 2259-2322.

Saeed, U., Shah, S. Y., Ahmad, J., Imran, M. A., Abbasi, Q. H., & Shah, S. A. (2022). Machine learning empowered COVID-19 patient monitoring using non-contact sensing: An extensive review. *Journal of pharmaceutical analysis*.

Sikder, N., & Nahid, A. A. (2021). KU-HAR: An open dataset for heterogeneous human activity recognition. *Pattern Recognition Letters*, *146*, 46-54.

Supriya, S., Siuly, S., Wang, H., & Zhang, Y. (2018). EEG sleep stages analysis and classification based on weighed complex network features. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *5*(2), 236-246.

Supriya, S., Siuly, S., Wang, H., Cao, J., & Zhang, Y. (2016). Weighted visibility graph with complex network features in the detection of epilepsy. *IEEE access*, *4*, 6554-6566.

Tripathi, R. K., Jalal, A. S., & Agrawal, S. C. (2018). Suspicious human activity recognition: a review. *Artificial Intelligence Review*, *50*, 283-339.