BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 07/07 pp124–138
Volume 4, Number 2, June 2021
DOI: 10.26599/BDMA.2020.9020022

Sensitive Integration of Multilevel Optimization Model in Human Activity Recognition for Smartphone and Smartwatch Applications

Samaher Al-Janabi* and Ali Hamza Salman

Abstract: This study proposes an intelligent data analysis model for finding optimal patterns in human activities on the basis of biometric features obtained from four sensors installed on smartphone and smartwatch devices. The proposed model, referred to as Scheduling Activities of smartphone and smartwatch based on Optimal Pattern Model (SA-OPM), consists of four main stages. The first stage relates to the collection of data from four sensors in real time (i.e., two smartphone sensors called accelerometer and gyroscope and two smartwatch sensors of the same name). The second stage involves the preprocessing of the data by converting them into graphs. As graphs are difficult to deal with directly, a deterministic selection algorithm is proposed as a new method to find the optimal root to split the graphs into multiple subgraphs. The third stage entails determining the number of samples related to each subgraph by using the optimization technique called the lion optimization algorithm. The final stage involves the generation of patterns from the optimal subgraph by using the association pattern algorithm called gSpan. The pattern finder based on Forward-Backward Rules (FBR) generates the optimal patterns and thus aids humans in organizing their activities. Results indicate that the proposed SA-OPM model generates robust and authentic patterns of human activities.

Key words: optimization; Ant Lion Optimization (ALO); gSpan; Forward-Backward Rules (FBR); Internet of Things (IoT); smartwatch; smartphone

1 Introduction

The Internet of Things (IoT) is a system of interconnected hardware and software technologies that produce data from connected devices and sensors in the Internet. Simply put, IoT is any device that can connect to the Internet and convey data through the network without any interaction between users and computers[1].

Smartphones and smartwatches. Smartphones are compact computers running on down Ant Lion Optimization (ALO) software, including operating systems, which can run other applications. Smartphones have become an integral part of people's daily life. These devices are widely used in communication and in accessing Internet platforms and social media[2].

Smartwatches have become increasingly widespread, and the competition in designing these devices with superior features has intensified. Smartwatches are devices worn on the wrist that can perform multiple tasks, such as accessing accounts, playing games, answering calls, playing audio files, and even taking images for those equipped with cameras. Smartwatches may be installed with several applications, such as those for measuring one's pulse rate and blood pressure[3,4].

Human biometrics featuring intelligent computation. The term "biometrics" is a combination of two Greek words, namely, "bio" (life) and "metric" (measurement). Biometrics can be defined as the science and technology of statistically analyzing and measuring biological data[5].

Biometrics is used to measure physiological and

• Samaher Al-Janabi and Ali Hamza Salman are with the Department of Computer Science, Faculty of Science for Women, University of Babylon, Babylon 964, Iraq. E-mail: samaher@itnet.uobabylon.edu.iq; ali.portit@gmail.com.
* To whom correspondence should be addressed.
Manuscript received: 2020-06-03; accepted: 2020-09-22

behavioral characteristics that make the identity of a person distinct. Hence, it is also the science of measuring behavioral and/or physical characteristics that are unique to each person.

Biometrics means "life measurement". The unique physiological characteristics of an individual are related to this term. Several aspects of biometrics have evolved and have thus been used to verify the identities of individuals. In identifying a person, this person's unique features are used, including the iris, face, signature, and fingerprint[5].

Intelligent computation involves the actions of agents in an environment. Agents in this sense may be the society, dogs, and humans.

An intelligent agent is a system that plays intelligently. It is flexible to change in the environment and aim to realize its goals in a smart manner. The goal of smart computing is to understand smart behavior and facilitate its application to industrial and natural systems[6].

Optimization is part of machine learning under supervised learning. To optimize something is to change it for the better. It is a way to achieve continuous improvement, and it can be applied to all aspects of life. Optimization aims to continuously develop processes and activities related to individuals and production paths[7].

2 Literature Review

Mi et al.[8] proposed an algorithm that learns people's activities while using smartphones. The algorithm identifies missing information and pattern changes and manages anomalies well. An advanced mobile phone client recognizable proof framework depends on conduct action cycle, which can be duplicated in other social examinations. In our proposed method, two sensors are used to collect data, and they differ in the techniques used.

Smartphones are important tools for applying modern technology, as they contain sensors suitable for mobility control applications. In Ref. [9], the system for Human Activity Recognition (HAR) was evaluated. A BlackBerry Z10 smartphone was used to collect gyroscope and acceleration data. The Decision Tree (DT) technique and F-score were employed to evaluate the performance of the HAR classifier. Similarly, the proposed method collects data from the gyroscopes and accelerometers in smartphones and smartwatches, but it uses different techniques.

In the field of mobile computing, most researchers have focused on mobile devices, including mobile phones. Such level of attention is understandable because of these devices' internet access and their sensors, including gyroscopes and accelerometers, which are important sensors used to distinguish activities. Reference [10] is based on the analysis of mobile biological research. A total of 18 activities were considered, and 9 sensors were used for biometric events (acceleration sensors, smart gyroscopes, and smart watches). Similarly, the proposed method collects data from the gyroscopes and accelerometers in smartphones and smartwatches, but it uses different techniques.

In the study of Ref. [11], the lion algorithm was used in conjunction with the evolutionary swarm algorithm to solve a variety of optimization issues. Four tests were conducted. The first test solved the problem of performance improvement. The results of the first test were compared with those of published evolutionary improvement algorithms. The second test involved a comparison of modern improvement algorithms. The third test dealt with improvement problems from a broad side. The fourth dealt with standard engineering problems. The results showed that the lion algorithm is equivalent to some and outperforms most optimization algorithms and thus can be used as an alternative. Hence, the lion algorithm is considered in the current work to achieve satisfactory results.

Smartphones and smartwatches are equipped with powerful sensors that provide a platform for implementing behavioral biometrics. To distinguish biometrics-based motions, Weiss et al.[12] used the accelerometer in identifying the activity of walking. In the study, an accelerometer sensor and a gyroscope were used on smart devices (watch and phone) to determine which of them works best in distinguishing activities. Similarly, the proposed method collects data from accelerometer and gyroscope sensors, but it uses different techniques.

Although the field of IoT has become increasingly important and is continuously evolving, we must not forget several aspects, including energy consumption, emission of radiation, and the generation of electronic waste. Alsamhi et al.[13] used an algorithm called Ant Colony System (ACS) to determine a near-optimal solution. The results showed reduced European Commission (EC) and CO_2 emissions.

Ray et al.[14] suggested using an effective algorithm to find recurring subgraphs in a large flow graph. Five large graph datasets were employed to determine the frequent subgraphs. The well-known Chernoff bounds were used. In the proposed method, sensors in smartphones and smartwatches are used to collect data. Moreover, the Deterministic Selection Algorithm (DSA) is used to detect the frequently recurring subgraphs, and different techniques are used to generate patterns.

Previous works are further summarized in Table 1.

3 Material and Methodology

In this part, the main principle used to identify and resolve the problem is discussed.

3.1 Big Data Analysis (BDA)

Big data refer to the set of data in multi format, such as structured, unstructured, and semistracture, sized from 1 TB to 1 ZB, and handle through combination between cloud computing and machin learning techniques. BDA involves advanced architecture and technologies that are designed to extract information from a large amount of data by allowing analysis, high-velocity capture, and/or discovery. Three main features characterize big data[15]:

- Quick preparation: Big data extricate recent experiences from a large volume of information almost always in real time.
- Forecasting: Exact prediction is used to implement statistical and numerical strategies in voluminous information and thereby revealing likelihoods for the extrication of valuable insights.
- Connection: Users piece data together to derive meaning.

3.2 IoT

IoT is a computing idea that describes the concept of connecting physical objects to the internet and introducing them to other devices. Objects are closely identified, with Radio-Frequency IDentification (RFID) used as the method of connection. IoT involves different sensors, techniques, and wireless technologies. As an object can be represented digitally in IoT, IoT is significant and is able to improve the object. In such a case, the object is no longer linked to its user only, but it is connected to surrounding objects and the database[1,4].

3.3 Types of sensors related to smartphones and smartwatches

Several types of sensors can be used in smart devices, such as smartphones and smartwatches.

A gyroscope provides direction, such as left/right and up/down directions, and orientation details with great precision; for example, it can detect the degree of deviation of a device. A gyroscope can also measure rotation and is different from the acceleration sensor. With this feature, the gyroscope can detect the direction of a smartphone and the amount of rotation it experiences. Popular applications, such as Google Sky Map and Pokémon Go, take advantage of the gyroscope to determine the direction of a user's phone[16].

An accelerometer detects vibration and tilt and is used to determine the direction and movement of three axes (x, y, and z). It is also capable of determining the direction of a phone (i.e., vertical, oblique, or horizontal direction). This sensor can determine the speed of movement of the phone in any linear direction[17].

3.4 Biometric features

Biometrics has become widely used to identify people or their characteristics. It is used in various applications, such as forensic medicine through Deoxyribo Nucleic Acid (DNA) and fingerprints. The use of biometric features has succeeded in various fields in terms of security. Biometric characteristics are divided into two groups[16].

3.4.1 Physiological biometrics

Physiological biometrics find the unique bodily functions of people and are used in multiple fields for the purpose of person identification and analysis. With such type of biometrics, one can measure brain activity and monitor a person's health, heartbeat, and ability to breathe in a wide range of applications, such as fingerprints, iris recognition, DNA, face recognition, hand geometry, palmprint, and odor/scent.

3.4.2 Behavioral biometrics

Behavioral biometrics is a field of study related to the measurement of unique identification patterns in human activities, such as typing rhythm, voice, and gait. It is the opposite of physical measurements.

3.5 Optimal recognition of activities

In recent years, the research on learning human activities has undergone significant growth. The results are of great importance as learning the daily activities of people may aid in the monitoring of their health, as well as in disease prevention and issuance of danger alerts. Extracting and discovering patterns of behavior can provide early and useful information about the person concerned. The goal is to analyze the various factors that affect the nature of identifying daily human activities with the data collected from wearable sensors[18].

4 Association Pattern

Association patterns provide supporting information for modeling connections in the real world that occur between objects when they are implemented in computer

Table 1 Comparison of previous works.

Researcher	Database/Dataset	Methodology	Evaluation measure	Advantage
Mi et al.[8]	UniMiB SHAR	Prophet algorithm	Cross validation; Mean Squared Error (MSE)	• This algorithm has solid strength for missing information and pattern changes and can manage anomalies well. • An advanced mobile phone client recognizable proof framework depends on conduct action cycle, which can be duplicated in other social examinations. • The framework adjusts among vitality utilization, information amount, and fitting exactness.
Capela et al.[9]	Information was gathered with a Blackberry Z10 cellphone utilizing the TOHRC Data Logger.	DT	Sensitivity, specificity, and F-score	HAR allows of understandable level of attention because of these devices' internet access and their sensors, including gyroscopes and accelerometers, which are important sensors used to distinguish activities.
Yoneda and Weiss[10]	Using data collected from four sensors	k-neighbors, DT, and random-forest	Equal Error Rate (EER), False Acceptance Rate (FAR), and False Rejection Rate (FRR)	• Assess a bigger number of sensors and gadgets than most other research by considering the accelerometer and gyrator sensors on both the cellphone and smartwatch, independently and together (a sum of nine sensor configurations are assessed). • This exploration can be applied to fabricate a real time biometric framework, as in the past we have built constant action acknowledgment frameworks.
Booth-alingam[11]	https://sites.google.com/view/lionalgorithm/test-suite	Lion optimization algorithm, nature-inspired optimization algorithms, swarm intelligence, and crossover mutation	Classical Evolutionary Programming (CEP), Fast Evolutionary Programming (FEP), Canonical Evolutionary Strategies (CES), and Fast Evolutionary Strategies (FES)	Skirt the neighborhood ideal arrangements, search for arrangements on a huge scale, maintain a strategic distance from the nearby ideal point, and distinguish different arrangements with comparable wellness.
Weiss et al.[12]	Smartphone and smartwatch sensors were used to collect data	k-neighbors, DT, and random forest	10-fold cross validation, EER, FAR, and FRR	Combine the data from nine sensors, use these (k-neighbors, DT, and random-forest) methods with fixed dimensions and multi methods to evaluate authentication.
Alsamhi et al.[13]	Collect data from sensors	ACS	Energy-efficient algorithm	• Determine a near-optimal solution that is a benefit of using ACS. • Reduce energy consumption EC. • Reduce CO_2 emissions.
Ray et al.[14]	Using five large graph datasets (Artificial Graph1 (10 000 nodes and 15 000 edges), Artificial Graph2 (10 000 nodes and 16 911 edges), Twitter, Hetrec, and ArnetMiner)	Streaming graphs sampling algorithm	Chernoff bounds	• Be quicker than the cutting edge enormous chart excavators while keeping up their exactness. • Be quicker and fit for finding more kinds of examples. • Give hypothetical assurances of calculation's exactness utilizing the notable Chernoff bounds, just as an investigation of the computational unpredictability of the methodology.

Note: TOHRC Data Logger is an easy-to-use tool for capturing sensor data.

applications. Patterns support designers in defining and understanding the implications of these associations accurately and thus help these experts in the smooth realization of their objectives. The classic problem of extracting association patterns between data purchased by customers and groups of items is referred to as transactions.

The most common model for pattern correlation uses the frequencies of sets of elements to quantify the correlation level.

4.1 Types of association patterns

Several types of algorithms are used to generate association patterns, as explained in Table 2. Some of these algorithms are applicable to datasets while others are applicable to graphs.

4.1.1 Frequency association pattern

Finding the correlation pattern from a frequency-based model is a simple and common approach. The importance of the raw frequency of a particular pattern does not differ from the primary importance of statistical relationships. Assume that database T contains a set of n transactions and is denoted by $T_1, \ldots, T_n$. Each transaction T_i is drawn on the universe of items U and can be represented as a multidimensional record of dimensionality, $d = |U|$, containing only binary attributes. The binary attribute represents a specific element in the record. The value is 1 in the record when the items are present and 0 when they are not in the transaction. On the practical side, comparing the universe of items U and the number of items T_i in the transaction reveals that the size is considerably large. To build a repeated mining algorithm, we take advantage of this feature, as shown in Algorithm 1. Some algorithms for the recurring approach are based on the pursuit and tree algorithms.

Algorithm 1 Scheduling Activities of smartphone and smartwatch based on Optimal Pattern Model (SA-OPM)

```
Input:  Real-time dataset taken from smartphone and
        smartwatch sensors
Output: Optimal patterns
// Collection & preprocessing stage
For each row in datasets
    For each column in datasets
        Call split based on activity related to smartphone (SP) and smartwatch (SW);
    End for
End for
For each row in SP
  For each column in SP
    Call draw graph;
  End for
End for
For each row in SW
  For each column in SW
    Call draw graph;
  End for
End for
// Find optimal graph based on Dselect-ALO stage
// Determine the pivot
For each id_dataset
    Find the pivot according to DSA;
End for
For each subgraph from multi subgraphs
  Call ALO;
  Call gSpan;
End for
// Evaluation stage
For all patterns generated from optimal graph
  Call FBR;
End for
End SA-OPM
```

Note: Dselect means deterministic selection algorithm; FBR means Forward-Backward Rules.

4.1.2 Frequent graph patterns

A graph has recurring patterns found through graphs or large graphs with no less than a support threshold set by the user. Recurring charts are useful in distinguishing and grouping different charts and in building and compiling indicators. Mining techniques for repetitive graphs have been successfully applied to study protein

Table 2 Advantage and disadvantage of existing techniques[20].

Techniques association pattern	Advantage	Disadvantage
Apriori algorithm	Use large itemset property; easily parallelized; easy to implement	Assume transaction database is memory resident, require many database scans.
Frequency Pattern (FP) growth algorithm	Only 2 passes over dataset; compressed dataset; no candidate generation; much faster than Apriori	FP-tree may not fit in memory and is expensive to build.
Graph-based substructure pattern mining (gSpan) algorithm	Lower memory requirement; no candidate generation; lexicographic ordering minimizes search tree false positives pruning.	Divide dataset each once in two groups because it is based on heretical clustering.

structures. Recurring patterns have also been used as indexing features to perform quick searches in graphs. The method is superior to traditional path-based indexing. Comparisons among that techniques are shown in Table 2.

4.2 Association pattern techniques

4.2.1 Apriori algorithm

The apriori algorithm was suggested by Agarwl for mining association liaison rules[19,20]. This algorithm depends on the bottom-up approach in terms of breadth. It works on the base of frequency principle and summarization of the dataset, in which case the group of elements and the subgroup do not exceed zero. This feature is called the anti-monotype feature of support. What is distinctive about this method is that it performs repeated surveys of a group of candidates. However, it is time consuming to use when producing candidates, and its accounting cost is high[19,21].

4.2.2 FP growth algorithm

This method does not depend on the generation of a filter when extracting repeating sets of elements. It builds highly compressed data (an FP-tree) to compress the original transaction database. It focuses on the growth of the refined part, and thus, it does not need to generate an expensive filter. These features lead to a high efficiency. We can thus regard the FP growth algorithm as a good mathematical method for use with pairing rules[20,22].

4.2.3 gSpan

gSpan is a popular algorithm (shown in Algorithm 2) that is used in Depth First-level Search (DFS) in repeating subgraphs. It has two DFS glossary arrangements and a minimum code, which establishes a useful and effective way to set DFS search support signals. This method finds repeated subcharts without having to generate a filter and prune errors. It combines checking subcharts and growth in one business. Hence, its use leads to a rapid mining process that reveals frequent infrastructure. gSpan also creates a new lexical arrangement between charts and assigns a unique DFS code to each histogram. On the basis of this arrangement, the method uses the search technology at the first level to find subdrawings related to an effective situation. It is designed to reduce duplicate graphs[21,23].

Algorithm 2 gSpan

Input: s is a DFS code; D is a graph data set; $\min_{\text{sup}}$ is the minimum support threshold
Output: Set of patterns S
1: $S = \varnothing$;
2: Call gSpan(s, D, $\min_{\text{sup}}$, S); procedure pattern growth graph (s, D, $\min_{\text{sup}}$, S);
1: If $s \neq$ dfs (s), then return;
2: Insert s into S;
3: Set $\boldsymbol{C}$ to $\varnothing$;
4: Scan D once; find all the edges, such that s can be right-most extended to s_r e; insert s_r e into $\boldsymbol{C}$ and count its frequency;
5: Sort $\boldsymbol{C}$ in DFS lexicographic order;
6: For each frequent s_r e in $\boldsymbol{C}$ do
7: gSpan(s_r e, D, $\min_{\text{sup}}$, S);
8: Return S

5 Proposed System

The proposed system uses a raw dataset collected from two types of sensors (gyroscope and accelerometer) found in smartphones and smartwatches. Then, preprocessing is performed to convert the data into graphs. The DSA is used to find the optimal root for each graph. Subsequently, the ALO algorithm is implemented to find the frequently occurring subgraphs and generate patterns. In the final step, the system is evaluated, as illustrated in Fig. 1 and Algorithm 1.

Step 1: Apply deterministic selection algorithm (Dselect).

The proposed system is implemented on a raw dataset. The DSA is used to determine the best seed for each cluster from the dataset, as shown in Algorithm 3.

Algorithm 3 Dselect

Input: Complex Graph (CG)
Output: Set of pivot to split the complex graph
//Split Dataset_ID into multi groups and sort each group
1: Split Dataset_ID into 5 groups, sort each group;
2: Create empty matrix called $\boldsymbol{C}$, number of elements is $N\backslash 5$;
// N is total number of samples of Dataset_ID
3: $P =$ Dselect ($\boldsymbol{C}$, $N\backslash 5$, $N\backslash 10$);
// Recursively computer median of $\boldsymbol{C}$
4: Partition Dataset_ID around P;
5: For I in range 1 to $N - J$
6: For j in range $N - I$ to N
7: If $I = J$
8: Return P;
9: End if
10: If $J < I$
11: Return Dselect (1st part of Dataset_ID, $J - 1, I$);
12: Else
13: Return Dselect (2nd part of Dataset_ID, $N - J, I - J$);
14: End if
15: End for
16: End for
End Dselect

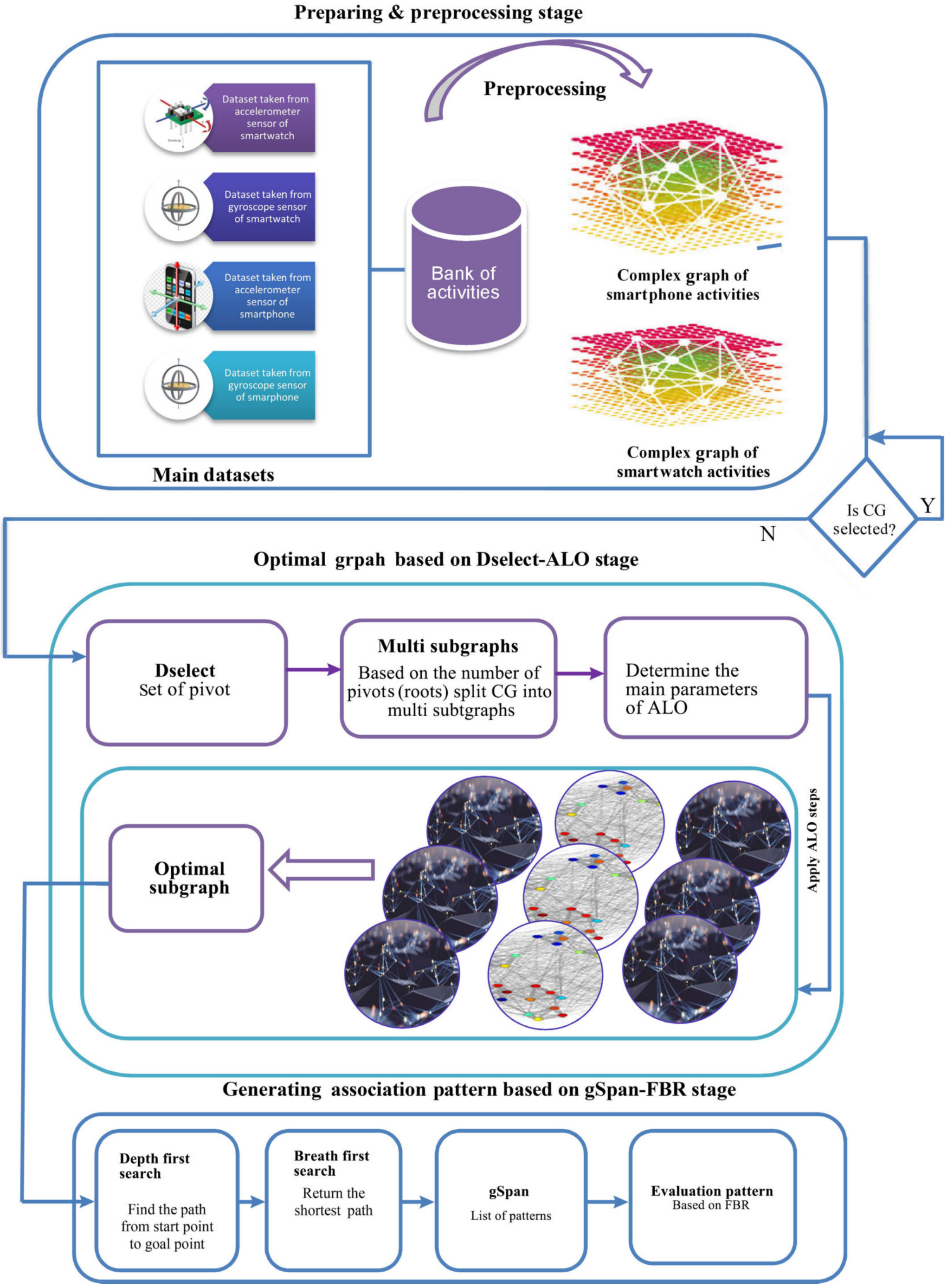


Fig. 1 Block diagram of SA-OPM.

Step 2: Apply ALO.

The ALO is one of the main optimization algorithms shown in Algorithm 4. For further details, refer to Ref. [24].

Step 3: Generate patterns and verify them using gSpan-FBR, as explained in Algorithm 5.

In this section, we present the main measures used to validate the association patterns, including the support and confidence, and the forward edges/backward edges with gSpan measures, as is explained in Algorithms 4 and 5.

• Support and confidence

Let Na and Nb be the respective numbers of A and B transactions, and let Nab be the number of transactions, where A and B items appear simultaneously. The support of the rule $A \rightarrow B$ is the proportion of joint A and B transactions:

$$\mathrm{SUP}(A \rightarrow B) = \mathrm{Pab} = \mathrm{Nab}/N \tag{1}$$

Algorithm 4 ALO

Input: Multi subgraphs
Output: Optimal subgraph
1: Initialize the first population of ants and antlions randomly
2: Calculate the fitness of ants and antlions
//Find the best antlions and assume it as the elite (determined optimum)
3: While the end criterion is not satisfied
4: For every ant
5: Select an antlion using Roulette wheel;
6: Update c^t and d^t,
$$c^t = \frac{c^t}{I}, \quad d^t = \frac{d^t}{I};$$
7: Create a random walk and normalize it,
$$X(t) = [0, \mathrm{cumsum}(2r(t_1) - 1), \mathrm{cumsum}(2r(t_2) - 1), \ldots, \mathrm{cumsum}(2r(t_n) - 1)],$$
$$x_i^t = \frac{(x_i^t - a_i) \times (d_i - c_i^t)}{(d_i^t - a_i)} + c_i;$$
8: Update the position of ant,
$$\mathrm{Ant}_i^t = \frac{R_A^t + R_E^t}{2};$$
9: End for
Calculate the fitness of all ants, replace an antlion with its corresponding ant if becomes fitter,
$$\mathrm{Antlion}_j^t = \mathrm{Ant}_i^t \text{ if } f(\mathrm{Ant}_i^t) > f(\mathrm{Antlion}_j^t);$$
10: Update elite if an antlion becomes fitter than the elite;
11: End while

By contrast, confidence is the proportion of *B* transactions among *A* transactions, that is, the conditional frequency of *B* given *A*:

$$\mathrm{CONF}(A \to B) = \mathrm{Pab/Pa} = \mathrm{Nab/Na} = 1 - \mathrm{Nab/Na} \quad (2)$$

• Forward edges/backward edges with gSpan

These rules govern the order in which edges are added to the list when creating codes for a graph.

(1) If the first vertex of the current edge is less than the second vertex of the current edge (forward edge),

(A) If the first vertex of the next edge is less than the second vertex of the next edge (forward edge),

(a) If the first vertex of the next edge is less than or equal to the second vertex of the current edge,

(b) And if the second vertex of the next edge is equal to the second vertex of the current edge plus 1, the next edge is acceptable,

(c) Otherwise, the next edge being considered is not valid.

(B) Otherwise, the next edge is a backward edge,

(a) If the first vertex of the next edge is equal to the second vertex of the current edge,

(b) And if the second vertex of the next edge is less than the first vertex of the current edge, the next edge is acceptable,

Algorithm 5 gSpan-FBR

Input: Optimal cluster
Output: Set of patterns *S*
1: $S = \varnothing$
2: For each optimal cluster
3: s = Call DFS;
4: Insert s into S;
5: Call BFS;
6: Find all the edges;
7: If the trees have max number of edges
8: Remove duplication tree;
9: Else
10: Kept the tree has minimal number of edges;
11: End if
12: End for
//Forward & backward rules
13: For each value of *S*
14: If v_1 of $e_1 < v_2$ of e_1
15: If v_1 of $e_2 < v_2$ of e_2
16: If v_1 of $e_2 <= v_2$ of e_1 & v_2 of $e_2 == v_2$ of $e_1 + 1$
17: this is an acceptable e_2;
18: Else
19: e_2 being considered is not valid;
20: Else
21: e_2 is a backward edge,
22: If v_1 of $e_2 == v_2$ of e_1 & v_2 of $e_2 < v_1$ of e_1
23: this is an acceptable e_2;
24: Else
25: e_2 being considered is not valid;
26: Else
27: e_1 is a backward edge;
28: If v_1 of $e_2 < v_2$ of e_2 (forward edge)
29: If v_1 of $e_2 <= v_1$ of e_1 & v_2 of $e_2 == v_1$ of $e_1 + 1$
30: this is an acceptable e_2;
31: Else
32: e_2 being considered is not valid;
33: Else
34: e_2 is a backward edge;
35: If v_1 of $e_2 == v_1$ of e_1 & v_2 of $e_1 < v_2$ of e_2
36: this is an acceptable e_2;
37: Else
38: e_2 being considered is not valid;
39: End forward/backward edge;
End gSpan-FBR

(c) Otherwise, the next edge being considered is not valid.

(2) Otherwise, the current edge is a backward edge,

(A) If the first vertex of the next edge is less than the second vertex of the next edge (forward edge),

(a) If the first vertex of the next edge is less than or equal to the first vertex of the current edge,

(b) And if the second vertex of the next edge is equal to the first vertex of the current edge plus 1, the next edge is acceptable,

(c) Otherwise, the next edge being considered is not valid.

(B) Otherwise, the next edge is a backward edge,

(a) If the first vertex of the next edge is equal to the first vertex of the current edge,

(b) And if the second vertex of the current edge is less than the second vertex of the next edge, the next edge is acceptable,

(c) Otherwise, the next edge being considered is not valid.

(C) Otherwise, the next edge is a backward edge,

(a) If the first vertex of the next edge is equal to the first vertex of the current edge,

(b) And if the second vertex of the current edge is less than the Second vertex of the next edge, the next edge is acceptable,

(c) Otherwise, the next edge being considered is not valid.

6 Result

In this section, we explain the results from a practical perspective.

A comparison is performed to verify the stability of the results. The comparison focuses on the results of 3 out of 7 classes, as explained in Fig. 2, that shows the classification tree. Class 4 clarifies the case before reaching the stability stage, Class 5 explains the state of stability, and Class 6 shows the stability in the resulting values and the absence of any changes in them. Tables 3–8 and Figs. 3–5 (stability graph diagram) shown below present the results of each class in detail.

Table 5 provides the rules resulting from the data analysis. It shows that activity E is not important, and thus, no rule is generated.

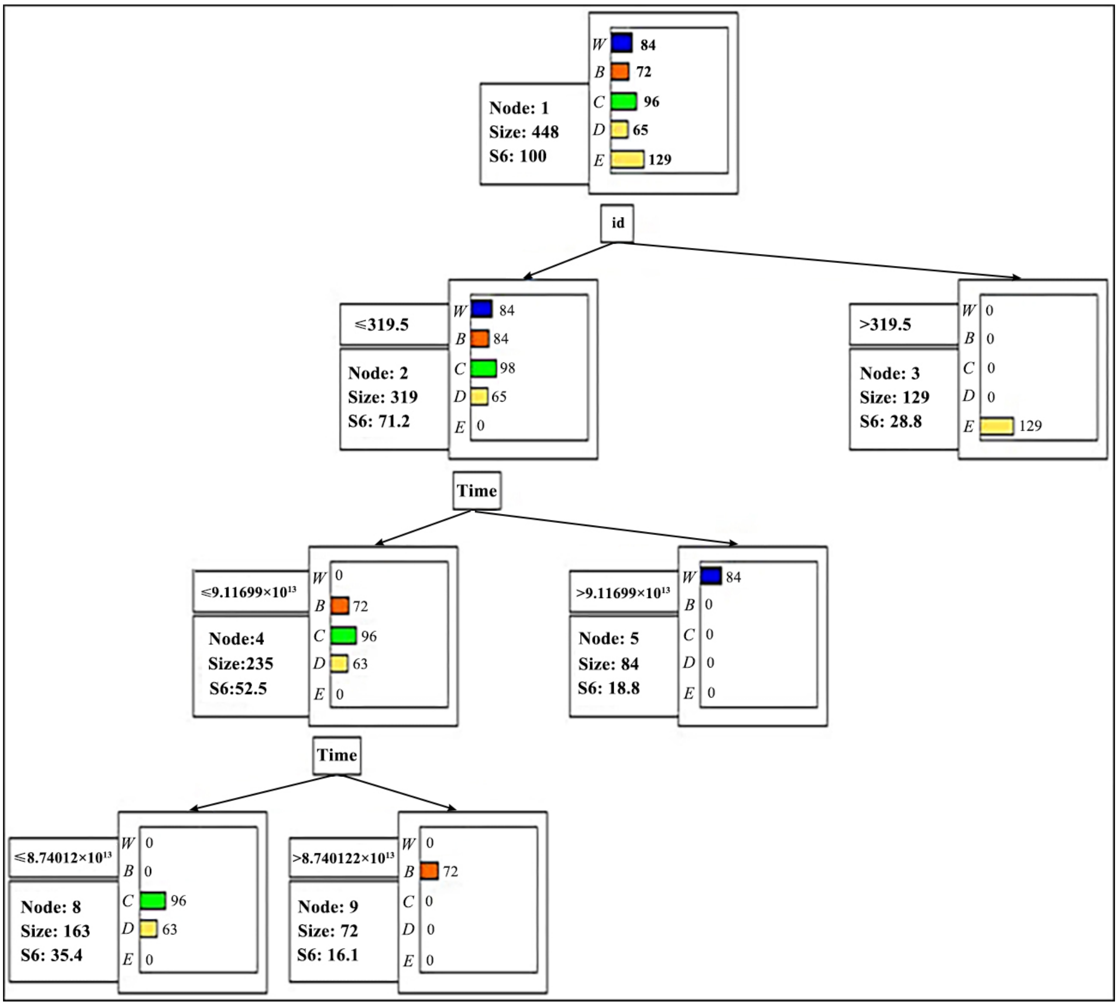


Fig. 2 Classification tree.

Table 3 Correlation matrix.

From/to	Subject_id	Time	X	Y	Z
Subject_id	**1.000**	−0.931	0.004	0.030	−0.053
Time	−0.931	**1.000**	−0.347	−0.027	0.067
X	0.004	−0.015	**1.000**	−0.332	0.272
Y	0.030	−0.027	−0.332	**1.000**	−0.259
Z	−0.053	0.067	0.272	−0.259	**1.000**

Table 4 Data of tree structures.

Node	Number of objects	%	Improvement (%)	Purity (%)	Split variable	Value	Parent node	Son node	Predicted value
1	448	100.00	114.870	28.79				2; 3	E
2	319	71.21	82.648	30.72	id	$\leqslant 319.5$	1	4; 5	C
3	129	28.79		100.00	id	> 319.5	1		E
4	235	52.46	75.157	41.70	time	$\leqslant 9.11699 \times 10^{13}$	2	8; 9	C
5	84	18.75		100.00	time	$> 9.11699 \times 10^{13}$	2		H
8	163	36.38		60.12	time	$\leqslant 8.74012 \times 10^{13}$	4		C
9	72	16.07		100.00	time	$> 8.74012 \times 10^{13}$	4		B

Table 5 Resulting rule.

Node	Activity (predicted value)	Rule
1	E	
2	C	If id $\leqslant$ 319.5, activity $= C$ in 71.2% of cases
3	E	If id $>$ 319.5, activity $= E$ in 28.8% of cases
4	C	If id $\leqslant$ 319.5 and time $\leqslant 9.11699\times10^{13}$, activity $= C$ in 52.5% of cases
5	H	If id $\leqslant$ 319.5 and time $> 9.11699\times10^{13}$, activity $= H$ in 18.8% of cases
8	C	If id $\leqslant$ 319.5, time $\leqslant 9.11699\times10^{13}$, and time $\leqslant 8.74012\times10^{13}$, activity $= C$ in 36.4% of cases
9	B	If id $\leqslant$ 319.5, time $\leqslant 9.11699\times10^{13}$, and time $> 8.74012\times10^{13}$, activity $= B$ in 16.1% of cases

Table (6-1) Class centroids (number of classes = 4).

Class	Subject_id	Time (min)	X	Y	Z
1	78.500	91 238 505 068 110.600	−0.048	−0.081	0.209
2	301.541	81 599 582 348 022.700	0.044	−0.051	0.002
3	440.000	80 225 016 245 326.600	−0.394	0.104	−0.002
4	443.000	80 225 166 031 727.000	−0.204	0.088	0.019

Table (6-2) Distances between class centroids (number of classes = 4).

	1	2	3	4
1	0	9 638 922 720 087.860	11 013 488 822 784.000	11 013 339 036 383.600
2	9 638 922 720 087.860	0	1 374 566 102 696.120	1 374 416 316 295.750
3	11 013 488 822 784.000	1 374 566 102 696.120	0	149 786 400.375
4	11 013 339 036 383.600	1 374 416 316 295.750	149 786 400.375	0

Table (6-3) Central objects (number of classes = 4).

Class	Subject_id	Time (min)	X	Y	Z
1 (H)	1	92 058 023 774 665	−1.121	0.045	0.063
2 (E)	448	80 225 415 675 729	0.090	−0.008	0.025
3 (E)	440	80 225 016 245 329	−0.394	0.104	−0.002
4 (E)	443	80 225 166 031 729	−0.204	0.088	0.019

In Fig. 3, the stability status is clarified when several classifications are taken. The result is the stability status of Class 5; no change occurs in the data stability condition, as shown in Table 9.

Attention is paid to the time factor when conducting the second experiment on all persons (51 persons)

Table (6-4) Distances between central objects (number of classes = 4).

Class	1 (*H*)	2 (*E*)	3 (*E*)	4 (*E*)
1 (*H*)	0	11 832 608 098 936	11 833 007 529 336	11 832 857 742 936
2 (*E*)	11 832 608 098 936	0	399 430 400	249 644 000
3 (*E*)	11 833 007 529 336	399 430 400	0	149 786 400
4 (*E*)	11 832 857 742 936	249 644 000	149 786 400	0

Table (7-1) Class centroids (number of classes = 5).

Class	Subject_id	Time (min)	X	Y	Z
1	78.500	91 238 505 068 110.800	−0.048	−0.081	0.209
2	205.500	84 521 758 474 900.400	0.108	−0.168	0.016
3	350.542	80 108 053 909 745.600	0.011	0.008	−0.006
4	443.000	80 225 166 031 724.100	−0.204	0.088	0.019
5	444.000	80 225 215 960 527.700	−0.277	−0.001	0.049

Table (7-2) Distances between class centroids (number of classes = 5).

Class	1	2	3	4	5
1	0	6716 746 593 210.450	11 130 451 158 365.300	11 013 339 036 386.700	11 013 289 107 583.100
2	6 716 746 593 210.450	0	4 413 704 565 154.810	4 296 592 443 176.250	4 296 542 514 372.620
3	11 130 451 158 365.300	4 413 704 565 154.810	0	117 112 121 978.562	117 162 050 782.187
4	11 013 339 036 386.700	4 296 592 443 176.250	117 112 121 978.562	0	49 928 803.625
5	11 013 289 107 583.100	4 296 542 514 372.620	117 162 050 782.187	49 928 803.625	0

Table (7-3) Central objects (number of classes = 5).

Class	Subject_id	Time (min)	X	Y	Z
1 (*H*)	1	92 058 023 774 665	−1.121	0.045	0.063
2 (*C*)	205	84 521 733 513 454	−4.397	1.194	−0.568
3 (*E*)	320	80 219 025 130 299	0.061	−0.011	−0.022
4 (*E*)	443	80 225 166 031 729	−0.204	0.088	0.019
5 (*E*)	444	80 225 215 960 529	−0.277	−0.001	0.049

Table (7-4) Distances between central objects (number of classes = 5).

	1 (*H*)	2 (*C*)	3 (*E*)	4 (*E*)	5 (*E*)
1 (*H*)	0	7 536 290 261 211	11 838 998 644 366	11 832 857 742 936	11 832 807 814 136
2 (*C*)	7 536 290 261 211	0	4 302 708 383 155	4 296 567 481 725	4 296 517 552 925
3 (*E*)	11 838 998 644 366	4 302 708 383 155	0	6 140 901 430	6 190 830 230
4 (*E*)	11 832 857 742 936	4 296 567 481 725	6 140 901 430	0	49 928 800
5 (*E*)	11 832 807 814 136	4 296 517 552 925	6 190 830 230	49 928 800	0

Table (8-1) Class centroids (number of classes = 6).

Class	Subject_id	Time (min)	X	Y	Z
1	78.500	91 238 505 068 110.700	−0.048	−0.081	0.209
2	205.500	84 521 758 474 900.300	0.108	−0.168	0.016
3	350.105	80 107 443 109 940.400	0.014	0.010	−0.006
4	437.000	80 224 866 458 925.500	−0.498	−0.363	0.003
5	440.000	80 225 016 245 331.600	−0.394	0.104	−0.002
6	444.000	80 225 215 960 528.500	−0.277	−0.001	0.049

and every person performing all daily activities (18 activities). The results are analyzed using the K-min classification algorithm, which reveals eight of the most important implemented activities; the other ten activities are classified as insignificant. The results are clarified below.

Figure 4 shows the stability status, fluctuation in data, and the implementation of activities.

Figure 5 shows the relationship among the time factor, activity, and the person for 18 activities.

Table (8-2) Distances between class centroids (number of classes = 6).

Class	1	2	3	4	5	6
1	0	6 716 746 593 210.440	11 131 061 958 170.300	11 013 638 609 185.200	11 013 488 822 779.100	11 013 289 107 582.200
2	6 716 746 593 210.440	0	4 414 315 364 959.890	4 296 892 015 974.780	4 296 742 229 568.660	4 296 542 514 371.780
3	11 131 061 958 170.300	4 414 315 364 959.890	0	117 423 348 985.109	117 573 135 391.234	117 772 850 588.109
4	11 013 638 609 185.200	4 296 892 015 974.780	117 423 348 985.109	0	149 786 406.125	349 501 603.000
5	11 013 488 822 779.100	4 296 742 229 568.660	117 573 135 391.234	149 786 406.125	0	199 715 196.875
6	11 013 289 107 582.200	4 296 542 514 371.780	117 772 850 588.109	349 501 603.000	199 715 196.875	0

Table (8-3) Central objects (number of classes = 6).

Class	Subject_id	Time (min)	*X*	*Y*	*Z*
1 (*H*)	1	92 058 023 774 665	−1.121	0.045	0.063
2 (*C*)	205	84 521 733 513 454	−4.397	1.194	−0.568
3 (*E*)	320	80 219 025 130 299	0.061	−0.011	−0.022
4 (*E*)	437	80 224 866 458 929	−0.498	−0.363	0.003
5 (*E*)	440	80 225 016 245 329	−0.394	0.104	−0.002
6 (*E*)	444	80 225 215 960 529	−0.277	−0.001	0.049

Table (8-4) Distances between central objects (number of classes = 6).

Class	1 (*H*)	2 (*C*)	3 (*E*)	4 (*E*)	5 (*E*)	6 (*E*)
1 (*H*)	0	7 536 290 261 211	11 838 998 644 366	11 833 157 315 736	11 833 007 529 336	11 832 807 814 136
2 (*C*)	7 536 290 261 211	0	4 302 708 383 155	429 6867 054 525	4 296 717 268 125	4 296 517 552 925
3 (*E*)	11 838 998 644 366	4 302 708 383 155	0	5 841 328 630	5 991 115 030	6 190 830 230
4 (*E*)	11 833 157 315 736	4 296 867 054 525	5 841 328 630	0	149 786 400	349 501 600
5 (*E*)	11 833 007 529 336	4 296 717 268 125	5 991 115 030	149 786 400	0	199 715 200
6 (*E*)	11 832 807 814 136	4 296 517 552 925	6 190 830 230	349 501 600	199 715 200	0

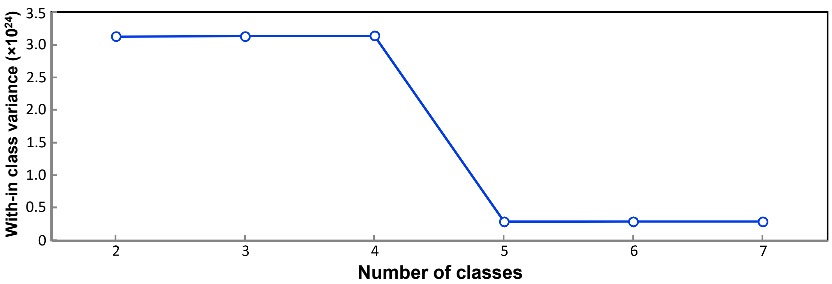


Fig. 3 Stable state.

After determining the best seed through the DSA and finding the optimal routing from the goal through the ALO (i.e., the stable state of the optimal number of clusters), gSpan-FBR is applied to the optimal subgraph (i.e., cluster) to find the optimal rules to achieve all events within adequate time. The results are shown in Table 10.

7 Discussion and Conclusion

The world has witnessed the rapid development of the internet and smart environments and devices, which have become widely applicable to various areas, including medicine and other industries. In this research, we attempt to find a way to generate optimal patterns to distinguish human activities by using a data analysis algorithm based on smart methods. Data from four sensors were collected in real time. The data obtained were preprocessed and converted into a graph. The optimal root for each subgraph was then identified using the DSA algorithm. A total of 18 seeds (root) were generated. Then, the ALO was applied to find the optimal subgraph among the 18 subgraphs. The main parameters of the ALO were as follows: Npop denotes the number of populations, *X* is the input dataset, *Y* is the goal, *F* is the activation function, and *S* is the seed. The secondary parameters were as follows: *M* denotes mating, UB is the upper dimension, and LB is the lower dimension. The accuracy within the best subgraph was 1.19% while that between optimal

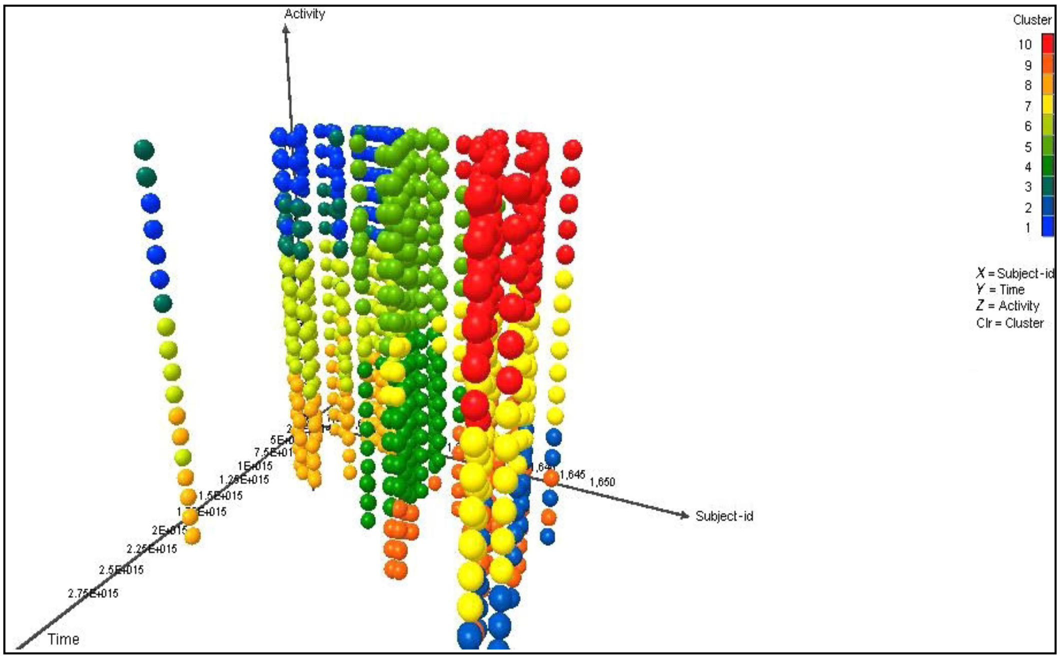


Fig. 4 Best status.

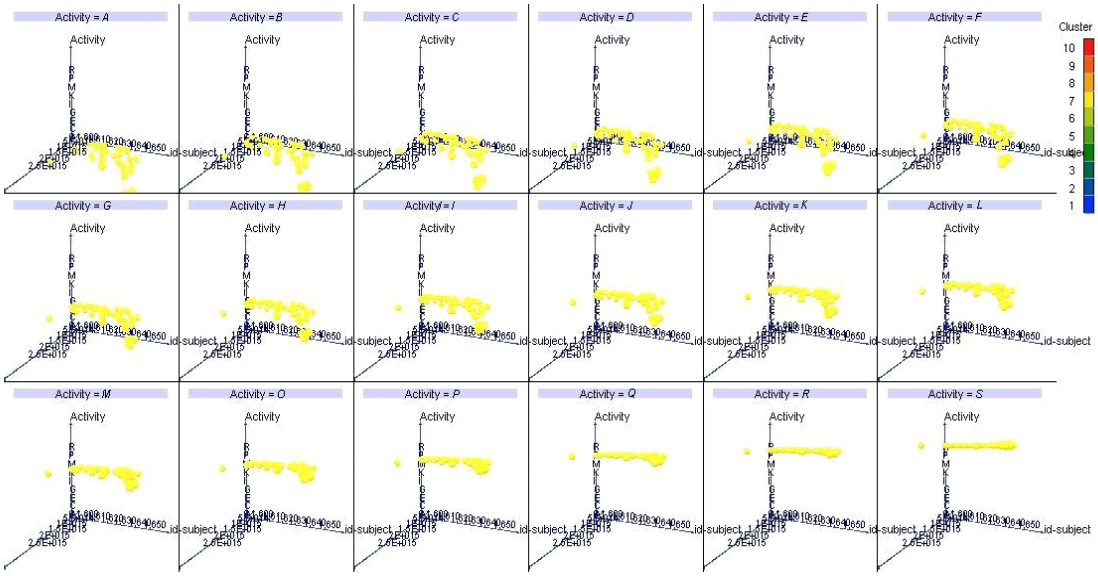


Fig. 5 Relationship among time, activity, and person.

Table 9 Class state.

Class	With-in classes (%)	Between classes (%)	Total (%)
4	12.90	87.10	100
5	1.19	98.81	100
6	1.19	98.81	100

subgraph and others was 98.81%. When applying gSpan-FBR on the best subgraph generated through the ALO, we found 10 activity patterns, with the remaining 8 being passive ones. The method was characterized by mathematical complications and considerable time consumption, but it proved to be the best in terms of accuracy. Hence, the algorithm satisfies the goal of optimization. Previously, dealing with a large volume of data through preprocessing was impossible. In this research, we converted the stream data into a graph. In

Table 10 Rules generated from the application of gSpan-FBR to the ALO results.

Node	Activity (predicated value)	Rule
1	A	
2	H	If $X \leqslant 0.076\,247\,3$, Activity $= H$ in 64.1% of cases
3	M	If $X > 0.076\,247\,3$, Activity $= M$ in 35.9% of cases
4	G	If $X \leqslant 0.076\,247\,3$ and $X > -5.565\,47$, Activity $= G$ in 10.9% of cases
5	H	If $X \leqslant 0.076\,247\,3$ and $X > -5.565\,47$, Activity $= H$ in 53.2% of cases
6	M	If $X > 0.076\,247\,3$ and $Z \leqslant 5.8432$, Activity $= M$ in 30.2% of cases
7	D	If $X > 0.076\,247\,3$ and $Z > 5.8432$, Activity $= D$ in 5.8% of cases
8	G	If $X \leqslant 0.076\,247\,3$, $X \leqslant -5.565\,47$, and $Z \leqslant 1.624\,54$, Activity $= G$ in 4.9% of cases
9	G	If $X \leqslant 0.076\,247\,3$, $X > -5.565\,47$, and $Z > 1.624\,54$, Activity $= G$ in 6.0% of cases
10	Q	If $X \leqslant 0.076\,247\,3$, $X > -5.565\,47$, and $Y \leqslant -8.6498$, Activity $= Q$ in 5.9% of cases
11	F	If $X \leqslant 0.076\,247\,3$, $X > -5.565\,47$, and $Y > -8.6498$, Activity $= F$ in 47.3% of cases
12	P	If $X > 0.076\,247\,3$, $Z \leqslant 5.8432$, and $X \leqslant 5.537\,04$, Activity $= P$ in 16.1% of cases
13	E	If $X > 0.076\,247\,3$, $Z \leqslant 5.8432$, and $X > 5.537\,04$, Activity $= E$ in 14.1% of cases
14	P	If $X > 0.076\,247\,3$, $Z > 5.8432$, and $Y \leqslant -9.5673$, Activity $= P$ in 0.8% of cases
15	D	If $X > 0.076\,247\,3$, $Z > 5.8432$, and $Y > -9.5673$, Activity $= D$ in 5.0% of cases

addition, the seed was not chosen randomly as the best seed (root), because each graph was selected on the basis of the DSA. This feature helps increase the accuracy rate and reduce mathematical operations.

8 Declaration

Conflict of interest: The authors declared that they have no conflict of interest.

Ethical approval: This article did not contain any studies with human participants or animals performed by any of the author.

References

[1] S. Al_Janabi, Smart system to create an optimal higher education environment using IDA and IOTs, *Int. J. Comput. Appl.*, vol. 42, no. 3, pp. 244–259, 2018.

[2] S. H. Ali, Miner for OACCR: Case of medical data analysis in knowledge discovery, in *Proc. 2012 6th Int. Conf. Sciences of Electronics, Technologies of Information and Telecommunications* (*SETIT*), Sousse, Tunisia, 2012, pp. 962–975.

[3] S. Al-Janabi, I. Al-Shourbaji, M. Shojafar, and M. Abdelhag, Mobile cloud computing: Challenges and future research directions, in *Prof. of IEEE 2017 10th International Conference on Developments in eSystems Engineering* (*DeSE*), Paris, France, 2017, pp. 62–67.

[4] S. Shahrestani, Assistive IoT: Enhancing human experiences, https://doi.org/10.1007/978-3-319-60164-9_2.

[5] W. S. Lima, E. Souto, K. El-Khatib, R. Jalali, and J. Gama, Human activity recognition using inertial sensors in a smartphone: An overview, *Sensors*, doi: 10.3390/s19143213.

[6] S. Al-Janabi and M. A. Mahdi, Evaluation prediction techniques to achievement an optimal biomedical analysis, *Int. J. Grid Utility Comput.*, vol. 10, no. 5, pp. 512–527, 2019.

[7] S. Al-Janabi, A. F. Alkaim, and Z. Adel, An innovative synthesis of deep learning techniques (DCapsNet & DCOM) for generation electrical renewable energy from wind energy, *Soft Comput.*, vol. 24, no. 14, pp. 10 943–10 962, 2020.

[8] S. Muhammad, B. Stephan, I. Ozlem, S. Hans, and H. Paul, Complex human activity recognition using smartphone and wrist-worn motion sensors. *Sensors*, vol. 16, no. 4, p. 426, 2016.

[9] N. A. Capela, E. D. Lemaire, N. Baddour, M. Rudolf, N. Goljar, and H. Burger, Evaluation of a smartphone human activity recognition application with able-bodied and stroke participants, *J. NeuroEng. Rehabil.*, vol. 13, no. 1, pp. 5, 2016.

[10] K. Yoneda and G. M. Weiss, Mobile sensor-based biometrics using common daily activities. in *Proc. 2017 IEEE 8th Annu. Ubiquitous Computing, Electronics Mobile Communication Conf.* (*UEMCON*), 2017, pp. 584–590.

[11] R. Boothalingam, Optimization using lion algorithm: A biological inspiration from lion's social behavior, *Evol. Intell.*, vol. 11, no. 1, pp. 31–52, 2019.

[12] G. M. Weiss, K. Yoneda, and T. Hayajneh, Smartphone and smartwatch-based biometrics using activities of daily living, *IEEE Acces*, vol. 7, pp. 133 190–133 202, 2019.

[13] S. H. Alsamhi, O. Ma, M. S. Ansari, and Q. Meng, Greening internet of things for greener and smarter cities: A survey and future prospects, *Telecommun. Syst.*, vol, 72, no. 4, pp. 609–632, 2019.

[14] A. Ray, L. B. Holder, and A. Bifet, Efficient frequent subgraph mining on large streaming graphs, *Intell. Data Anal.*, vol. 23, no. 1, pp. 103–132, 2019.

[15] S. Al-Janabi, M. A. Salman, and M. Mohammed, Pragmatic text mining method to find the topics of citation network, in *Big Data and Networks Technologies*, Y. Farhaoui, ed. Cham, Germany: Springer, 2020.

[16] F. Ramos, A. Moreira, A. Costa, R. Rolim, H. O. De Almeida, and A. Perkusich, Combining smartphone and smartwatch sensor data in activity recognition approaches: An experimental evaluation, in *Proc. of Int. Conf. Software*

Engineering & Knowledge Engineering, 2016, pp. 267–272.

[17] M. Masoud, Y. Jaradat, A. Manasrah, and I. Jannoud, Sensors of smart devices in the internet of everything (IOE) era: Big opportunities and massive doubts, *J. Sens.*, vol. 2019, p. 6 514 520.

[18] N. H. Kaghed, T. A. Abbas, and S. H. Ali, Design and implementation of classification system for satellite images based on soft computing techniques, doi: 10.1109/ICTTA.2006.1684408.

[19] S. H. Ali, A novel tool (FP-KC) for handle the three main dimensions reduction and association rule mining. in *Proc. 2012 6th Int. Conf. Sciences of Electronics, Technologies of Information and Telecommunications* (*SETIT*), Sousse, Tunisia, 2012, pp. 951–961.

[20] A. F. Alkaim, S. Al_Janabi, Multi objectives optimization to gas flaring reduction from oil production, doi: https://doi.org/10.1007/978-3-030-23672-4_10.

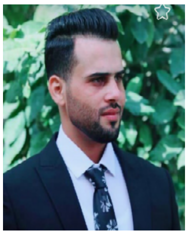
[21] S. Al_Janabi and N. Kadiam, Recommendation system of big data based on PageRank clustering algorithm. in *Big Data and Networks Technologies*, Y. Farhaoui, ed. Cham, Germany: Springer, 2020.

[22] S. Al-Janabi, M. A. Salman, and A. Fanfakh, Recommendation system to improve time management for people in education environments, *J. Eng. Appl. Sci.*, vol. 13, no. 24, pp. 10 182–10 193, 2018.

[23] L. F. Lu, X. X. Ren, L. Y. Qi, C. M. Cui, and Y. C. Jiao, Target gene mining algorithm based on gSpan, in *Collaborative Computing: Networking, Applications and Worksharing*, H. Gao, X. Wang, Y. Yin, M. Iqbal, eds. Cham, Germany: Springer, 2019, pp. 518–528.

[24] E. Hamouda, S. El-Metwally, M. Tarek, Ant Lion Optimization algorithm for kidney exchanges, *PLoS One*, vol. 13, no. 5, p. e0196707, 2018.

Samaher Al-Janabi received the BS, MS, and PhD degrees in computer science from Science College, University of Babylon, Iraq. Through years of study, she is specialized in the design, implementation, performance of measurement and intelligent analysis of huge/bigdata databases. She is a lecturer at Departments of Software, Computer, and Information Network, University of Babylon. Her research interests span topics concerning intelligent data analysis, knowledge discovery in database, soft computing techniques and applications, data & graph mining techniques (i.e., clustering, classification, predication, and optimization and association rules), cooperative computing (cluster, grid, and cloud), Internet of things, and mobile services & applications. She has published well over fifty-eight scientific papers and authored three books; one on new trends of KDD toward intelligent data analysis, one on soft computing techniques, and the third on intelligent miner of huge medical database. She is one of five women winners of the L'Oreal–UNESCO for Women in Science Levant and Egypt Regional Fellowships 2014. She has gotten one patent for research under title "Smart system to create optimal higher education environment in Iraq using IDA and IOTs" in 2018, and the other patent for research under title "A novel method to design smart station to prediction and control of concentrations of air pollution (PM2.5, PM10, NO_2, CO, O_3, SO_2) based on intelligent big data analysis" in 2020. She is one of four persons on the ACM-W Rising Star Award Selection Committee in 2019 and 2020. She is a reviewer of several local and international journals.



Ali Hamza Salman received the BS degree in computer science from Science College, University of Babylon, Iraq, and now he is a master student at Department of Computer Science, University of Babylon. His research interests span topics concerning intelligent data analysis, big data, and graph mining techniques (i.e., clustering, classification, predication, and optimization and association rules).