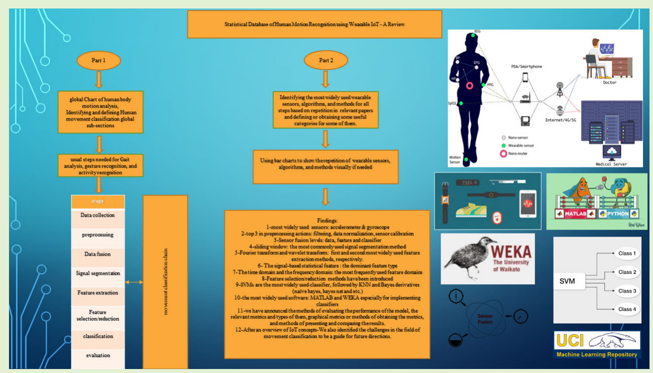
Statistical Database of Human Motion Recognition Using Wearable IoT—A Review

Eghbal Foroughi Asl, Saeed Ebadollahi, Reza Vahidnia, *Member, IEEE*, and Aliakbar Jalali, *Member, IEEE*



***Abstract*—Wearable sensors and the Internet of Things (IoT) will be two buzzwords that will be heard commonly in the coming decades. The combination of these two technologies soon will create a great revolution in applications that require motion recognition, such as health care, sports, and entertainment. The development of technology has made wearable sensors one of the most basic tools for human motion analysis. We believe that IoT is the most powerful complement to the use of wearable sensors in the analysis of human body motion. Using wearable IoT, all necessary human data will be collected and delivered via the Internet to the experts who can make accurate decisions about the type of activity, falling situations, freezing of gait (fog), and so on. In this article, the human motion analysis is presented in a chart and is divided into two parts: movement measurement and movement classification. However, this article focuses on movement classification that includes three subsections, gait analysis (GA), gesture recognition (GR), and human activity recognition (HAR), and is closely related to human motion recognition. In this article, our goal is to first acquaint the reader with the important steps required to classify the movement of the human body by wearable sensors and then by using tables to determine the most used algorithms and methods for each step. After briefly reviewing IoT concepts, directions for further research will be provided.**

***Index Terms*— Activity recognition, gait analysis (GA), gesture recognition (GR), wearable sensors.**

I. Introduction

DEPENDING on the definition of technology, the first wearable technologies were invented in the thirteenth century, and this technology was the same as glasses. Later, in the 16th century, the first portable, wearable watch, the Nuremberg eggs, was invented. They were designed to be worn around the neck and were a popular symbol until the invention of pocket watches and wristwatches. Another prototype of technology was an abacus ring made in China in the 17th century. The first wearable computer was created by mathematics professor Edward Thorp in the 1960s. In his book *Beat the Dealer: A Winning Strategy for the Game of Twenty-One*, Thorp reveals that he built a computer small enough to fit in a shoe to cheat in roulette. A timer predicted where the ball would land from the roulette table and overall helped Thorp and his colleague Claude Shannon by 44%in the game [1]. Over the next few decades, newer products will make wearable technologies more popular and modern. Reducing the cost of living, saving time, fast and immediate detection, and many other factors force us to move toward wearable sensors. The widespread deployment of sensors and the Internet of Things (IoT) facilitates this. Combining wearable sensors or wearable equipment in general with the IoT makes the equipment smarter and increases its efficiency [2]. Dian et al. [3] have stated that "IoT-enabled wearables are smart devices that can be worn as external accessories, embedded in clothing and garments, implanted in the body, or even adhered to or tattooed on the skin. These devices are

Manuscript received 25 January 2023; revised 21 March 2023 and 28 April 2023; accepted 20 May 2023. Date of publication 6 June 2023; date of current version 14 July 2023. The associate editor coordinating the review of this article and approving it for publication was Prof. Rosario Morello. *(Corresponding author: Saeed Ebadollahi.)*

Eghbal Foroughi Asl and Saeed Ebadollahi are with the Department of Electrical Engineering, Iran University of Science and Technology (IUST), Tehran 16846-13114, Iran (e-mail: eghbalforoughiasl@gmail.com; s_ebadollahi@iust.ac.ir).

Reza Vahidnia is with Rogers Communications Inc., Toronto, ON M4Y 2Y5, Canada, also with TELUS, Vancouver, BC V6E 0A7, Canada, and also with the Department of Electrical and Computer Engineering, British Columbia Institute of Technology, Vancouver, BC V5G 3H2, Canada (e-mail: rvahidnia@bcit.ca).

Aliakbar Jalali was with the School of Electrical Engineering, Iran University of Science and Technology, Tehran 16846-13114, Iran. He is now with the Faculty of Cybersecurity, University of Maryland, College Park, MD 20742 USA, and also with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506 USA (e-mail: drjalali@iust.ac.ir).

Digital Object Identifier 10.1109/JSEN.2023.3282171

able to connect to the Internet in order to collect, send, and receive the information that can be used for smart decision-making." Wearable sensors have different types with different applications. One of the most widely used types of wearable sensors is inertial sensors. These sensors mainly include a gyroscope, an accelerometer, and a magnetometer. These sensors are used either individually or in groups to report the specific force of the body, angular velocity, and orientation of the human body [4]. As was mentioned, these sensors have various applications, one of the most important of which is in the human body motion analysis. Other types of sensors that are used in human body motion analysis are introduced in this article, such as force sensors, pressure sensors, and electrodes. This article, while providing a general definition of the human body motion analysis, defines a general and universal chart of it and, while defining each part of this chart, focuses on the classification section. When defining each subsection of the chart, it introduces the steps to perform the project related to each subsection of movement classification [human activity recognition (HAR), gesture recognition (GR), and gait analysis (GA)]. This article will give the reader the idea to adapt the subject matter to one of the subsections of movement classification and then know the steps. Finally, the algorithms for classification, feature extraction, feature reduction, feature selection, validation, segmentation, the actions desired by the authors for preprocessing the wearable sensor signals, and the type of sensors are fully identified and reported based on repeated use in papers. Also, the most widely used software in this field is fully identified.

II. Main Body of This Article

One of the most important applications of wearable sensors is in the human body motion analysis. Human body motion analysis is defined as any method that involves any means to obtain a quantitative or qualitative measurement of it [5]. Human body motion analysis has many applications in various fields including: 1) medical evaluation; 2) monitoring people; and 3) activity recognition (activity recognition has subsections of fitness, health care, entertainment, and games) [6]. Human body motion analysis by wearable sensors is divided into two parts: 1) movement measurement and 2) movement classification that has three subsections: GA, GR, and activity recognition [6] (see Fig. 1).

In the first category, due to the specificity of the body part, only the estimation of movement parameters (e.g., the orientation and position of each joint) is needed, and in the second category, in addition to estimating the uncertainties, there is a need to classify features or data related to movement classification, which will require algorithms for classification, feature extraction, feature reduction, validation, and so on. The first category focuses on measurements of specific parts of the body such as the neck, head, torso, and upper and lower limbs [6], [7]. This category has been briefly reviewed in several papers. However, the second part is very extensive and has been studied in many papers in various ways, which require complex algorithms. This section estimates spatiotemporal gait parameters, assesses gait abnormalities, recognizes meaningful human expressions, including hand, arm, and face, and also includes activity recognition, fall detection, classification of daily activities, and so on [8], [9], [10], [11], [12], [13]. As it was mentioned earlier, this section includes three subsections: HAR, GA, and GR. In HAR, various human activities, such as walking, running, sitting, sleeping, standing, showering, cooking, driving, opening doors, and abnormal activities, will be detected. Data can be collected from wearable sensors or through video frames or images [14]. An interesting definition of activity recognition subsets is provided in the paper [15]: activity recognition can be referred to as the process of describing and classifying actions, pinpointing specific movements, and extracting unique patterns from the dataset using heterogeneous sensing modalities. The GA subsection is also described in [16] as follows: GA is a measure that can be easily translated from animals to humans, especially in the case of motor diseases, such as PD. This study involves quantifying (introducing and analyzing measurable gait parameters) and interpreting [e.g., different conclusions about mobility (health, age, weight, speed, and so on)] using the gait pattern. Wearable sensors play a key role in data collection in this area. GR with wearable sensors is an active research field that seeks to integrate the motor channel into human–computer interaction. GR is a computational process that attempts to detect and interpret human movements using mathematical algorithms. The program is also used in virtual environment control, sign language translation, robot remote control, or music creation [17]. GR with wearable sensors has received so much attention from researchers in recent years. In the following, we will extract the steps taken by scientists and researchers to carry out the project in each of the above fields, respectively, and identify the most widely used algorithms for each stage.

A. Data Collection

The first step for all three subsections is the collection of motion data or, in general, the data related to the movement classification operation by wearable sensors so that the wearable sensors are attached either directly to the relevant parts or by other wearable devices, such as gloves, shoes, smartphones, and glasses, which are attached to the body of the person. In the following, we will see that the available data in this field can also be used; in fact, many companies and unions have created their own motion datasets for these issues, and this dataset can also be used. Now, in this part of this article, using the studied papers, we will determine the type of sensors used for data collection, and we will also get acquainted with some of the available datasets in this field.

1) Wearable Sensors in the Literature: As it was mentioned, wearable sensors can connect to various parts of the body such as arms, legs, thighs, chest, head, and neck directly or can be worn on wearable devices, such as glasses, shoes, smartphones, and gloves. Wearable sensors available in the references are categorized and presented in a table (these sensors are sometimes used individually and sometimes as a combination of several sensors, and in the meantime, the fusion of sensors should not be neglected). In the bar chart, you can see the most commonly used sensors. To avoid increasing the number of pages of the paper, more information is available in the table named sensors. For more information, refer to the

TABLE I
SENSOR CATEGORIES

sensor	category
Accelerometer, gyroscope, magnetometer, orientation, compass, goniometer, …	Motion sensor
Ultrasonic sensor, passive infrared, …	motion detector
Ventilation sensor, temperature, EMG, PPG, ECG, Heart rate, Humidity, EOG, Carbon monoxide, Galvanic skin response sensors, Oximetry, EEG, …	Bio sensors & chemical sensors
Pressure, strain, Force, PVDF,…	Pressure & force sensors
GPS, Bluetooth beacon, UWB TAG, …	Positioning & tracking sensors
Microphone, camera, headsets, …	Audio & visual sensors
Light sensors, optical sensors, …	Optical & light sensors
Flex sensors, bend sensors, …	Bend sensors
Capacitive sensor for measuring distance, infrared, electric field sensor, …	Proximity sensors

attached table. The category of sensors used and the sensors in each category are presented in Table I. How to select a category is based on the type of application or the type of signal measured by the sensor, and in another bar chart, the most commonly used sensor categories are identified. Of course, before presenting the bar charts, we will define each sensor category, specify the sensors that are included in this category, will have a short and general overview of some sensors, and then we will announce a quantitative analysis. In this section, we try to introduce the main or lesser-known sensors, and the sensors whose function can be understood from their names will not be introduced.

Motion Sensors: Wearable motion sensor data can describe the type, quantity, and quality of motion-related activities in the community [18]. This category includes these sensors: accelerometer, gyroscope, magnetometer, compass, orientation, and goniometer. As you may know, an accelerometer detects linear motion and gravitational forces by measuring the acceleration in three axes (x, y, and z) [19]. A gyroscope measures the rotation rate [19]. The magnetometer detects and measures the magnetic fields of the Earth [19]. The goniometer is generally used to measure angular changes caused by body movements, as it is known from its ancient name. A flexible goniometer can be used to measure the relative rotation between two parts of the human body [20].

Bio and Chemical Sensors: These sensors are sensors that provide measurable signals proportional to the analyte concentration [21], [22]. The sensors that fall into this category are very wide. These sensors include EMG, temperature, humidity, EOG, ventilation sensor, PPG, ECG, heart rate, carbon monoxide (CO), galvanic skin response (GSR) sensor, oximetry, MMG, and EEG. Electromyography (EMG) measures the electrical signals produced by muscle movement and contraction. The EMG sensor has two types: surface EMG or sEMG, and intramuscular EMG (iEMG) [19]. iEMG may also be referred to as inserted EMG; in any case, this type of sensor is placed directly in the muscle under the skin, unlike the first sensor that is placed on the surface of the skin. Although iEMG may have a better performance in measuring the electrical signals of muscles, it is used less than sEMG due to its invasive nature and difficulty of use. In most papers, what is meant by EMG is sEMG. The temperature sensor is one of the most famous sensors in this category because temperature is one of the most important physiological parameters of the human body, which can be used as a reference value to monitor human health [23]. A flexible humidity sensor measures precise humidity by enabling humidity measurement with very fast and consistent resistance changes depending on the atmospheric humidity [24]. An electrooculography (EOG) sensor records eye movement by detecting a voltage difference between the cornea and retina [25]. The ventilation sensor attached to the abdomen measures the expansion and contraction associated with breathing rate and volume, representing the physiological response to bodily movement [26], [27]. Electrocardiography (ECG) and photoplethysmography (PPG) are sensors for heart rate monitoring [19]. ECG can also be called EKG. Wearable CO sensors are employed to monitor odorless and colorless CO [28]. GSR is a change in the electrical properties of the skin. The signal is then used to capture autonomic nerve responses as a parameter of sweat gland function [29]. Oximetry sensors are used for long-term, noninvasive monitoring of SpO_2 in human blood [30]. Mechanomyography (MMG) acts like EMG, but, instead of using electrodes, it uses a microphone or accelerometer to measure low-frequency muscle contractions and vibrations [19]. Electroencephalography (EEG) sensors are placed on a participant's head, and then, the electrodes detect brainwaves from the subject. EEG measures the electrical activity of the brain as the voltage at different points of the brain acts as the basis of EEG. It can be used for freezing of gait (fog) detection [31]. Now, we will introduce the next category.

Pressure and Force Sensors: This category includes these sensors: pressure sensor, strain sensor, force, and polyvinylidene fluoride (PVDF). Force and pressure are two sides of the same coin and have a similar nature. A force sensor helps to measure the amount of force applied to an object. These sensors can be either worn in shoes or placed on the ground using a cover and used. A force-sensing resistor (FSR) is a good example of these sensors [32]. Pressure sensors can also be directly attached to body parts, e.g., head or back, or used in wearable textiles. A pressure sensor detects, monitors pressure, and converts the data into an electronic signal. A strain sensor can measure electrical/optical and other responses to strain in a material [33]. It can be said that the type of signal measured by this sensor is similar to the previous two sensors. PVDF material is the basis of the PVDF sensor that measures pressure with a relatively high range [34].

Now, we come to the next category.

Audio and Visual Sensors: The sensors of this category include microphones, cameras, audio, and audio input and output devices. As it is clear from the last two names, these two names do not clearly define the sensor used and, therefore, only sit in the definition of the category. In the row related to the audio name in the attached table, [35] shows that this article has not specified this sensor precisely. Regarding the

TABLE II

PREPREPARED DATASET. ACC = ACCELEROMETER, GYRO = GYROSCOPE, COMP = COMPASS, FSR = FORCE SENSITIVE RESISTOR, UWB TAG = ULTRAWIDEBAND TAG, MFS = MAGNETIC FIELD SENSOR, UST = ULTRASONIC TRANSMITTER (SENSOR), MAG = MAGNETOMETER, ECG = ELECTROCARDIOGRAPHY, PPG = PHOTOPLETHYSMOGRAPHY, TEMP = TEMPERATURE, ORIENT = ORIENTATION SENSOR, HR = HEART RATE MONITOR, TOU = TOUCH SENSORS, CAM = CAMERA, LAC = LINEAR ACCELERATION SENSOR, MIC = MICROPHONE, RTC = REAL-TIME CLOCK, TI-SW = TILT SWITCHES, LIGHT = LIGHT SENSOR, PROXI = PROXIMITY SENSOR, AUDIO = AUDIO SENSORS, IR/V LIGHT = INFRARED/VISIBLE LIGHT SENSORS, HF LIGHT = HIGH-FREQUENCY LIGHT, PRESSURE = PRESSURE SENSORS, HUMID = HUMIDITY SENSORS, AND GPS = GPS SENSORS

Reference number	Datasets	Application	Sensors used	reference	repository
51	ActRecTut	HAR, GR	Acc, Gyro	51	-
54	THE CAR QUALITY INSPECTION	HAR, GR	Acc, Gyro, MFS, UST	53	-
54	THE WOODSHOP	HAR, GR	Acc, Gyro, Mag	54	-
54	THE DRINK AND WORK	HAR, GR	Acc, Gyro, Mag	54	-
55	M HEALTH	HAR	Acc, Gyro, Mag, ECG	-	UCI machine learning repository
56	DAPHNET	GA	Acc	-	UCI machine learning repository
57	Wrist PPG During Exercise	HAR	PPG, ECG, Acc, Gyro	-	PHYSIONET
58	Gait in Parkinson's Disease	GA	Pressure	-	PHYSIONET
59	PAMP2	HAR	Acc, Gyro, Mag, Temp, Orient, HR	-	UCI machine learning repository
59	M HEALTH	HAR	Acc, Gyro, Mag, ECG	-	UCI machine learning repository
60	Daily and Sports Activities Data Set	HAR	Acc, Gyro, Mag	-	UCI machine learning repository
62	HASC	HAR	Acc, TOU	61	-
62	Human Activity Recognition Using Smartphones Data Set	HAR	Acc, Gyro	289	UCI machine learning repository
63	OPENPOSE	HAR, GR	Cam	290	-
64	Human Activity Recognition Using Smartphones Data Set	HAR	Acc, Gyro	289	UCI machine learning repository
64	USC-HAD	HAR	Acc, Gyro, Mag	65	-
64	SHO	HAR	Acc, Gyro, Mag, LAC	66	-
65	MIT Place Lab Dataset	HAR	Acc, HR	291	-
65	UC Berkeley WARD Dataset	HAR	Acc, Gyro	292	-
65	CMU Multi-Modal Activity Database (CMU-MMAC)	HAR	Cam, Acc, Gyro	75	-
67	opportunity	HAR	Acc, Gyro, Mag, Mic, Cam	-	UCI machine learning repository
67	Skoda Mini Checkpoint dataset	HAR, GR	Acc	68	-
67	ACTITRACKER	HAR	Acc	69	-
67	DARMSTADT daily routines	HAR	Acc, RTC, TI-SW, Temp, Light	70	-
70	DARMSTADT daily routines	HAR	Acc, RTC, TI-SW, Temp, Light	70	-
71	DARMSTADT daily routines	HAR	Acc, RTC, TI-SW, Temp, Light	70	-
72	Ubicomp08	HAR	Proxi	72	-
73	BOOKSHELF	HAR, GR	Acc, Gyro, Mag	73	-

TABLE II

(Continued.) PREPREPARED DATASET. ACC = ACCELEROMETER, GYRO = GYROSCOPE, COMP = COMPASS, FSR = FORCE SENSITIVE RESISTOR, UWB TAG = ULTRAWIDEBAND TAG, MFS = MAGNETIC FIELD SENSOR, UST = ULTRASONIC TRANSMITTER (SENSOR), MAG = MAGNETOMETER, ECG = ELECTROCARDIOGRAPHY, PPG = PHOTOPLETHYSMOGRAPHY, TEMP = TEMPERATURE, ORIENT = ORIENTATION SENSOR, HR = HEART RATE MONITOR, TOU = TOUCH SENSORS, CAM = CAMERA, LAC = LINEAR ACCELERATION SENSOR, MIC = MICROPHONE, RTC = REAL-TIME CLOCK, TI-SW = TILT SWITCHES, LIGHT = LIGHT SENSOR, PROXI = PROXIMITY SENSOR, AUDIO = AUDIO SENSORS, IR/V LIGHT = INFRARED/VISIBLE LIGHT SENSORS, HF LIGHT = HIGH-FREQUENCY LIGHT, PRESSURE = PRESSURE SENSORS, HUMID = HUMIDITY SENSORS, AND GPS = GPS SENSORS

73	MIRROR	HAR, GR	Acc, Gyro, Mag	73	-
74	INTEL RESEARCH	HAR	Acc, Audio, IR/V Light, HF Light, Pressure, Humid, Temp, Comp	35	-
76	WISDM	HAR	Acc	76	-
77	UTD-MHAD	HAR, GR	Acc, Gyro	77	-
78	HHAR	HAR	Acc	78	-
79	SHL	HAR	Acc, Gyro, Mag	79	-
80	SARD	HAR	Acc, Gyro, Mag, GPS	80	-
81	UniMiB SHAR	HAR	Acc	81	-
82	ExtraSensory	HAR	Acc, Gyro, Mag, GPS, Mic, Comp	82	-
83	FIC	HAR, GR	Acc	83	-
84	WHARF	HAR, GR	Acc	84	UCI machine learning repository
85	SBRHA	HAR	Acc, Gyro	85	UCI machine learning repository
86	uWave	GR	Acc	86	-
87	M HEALTH	HAR	Acc, Gyro, Mag, ECG	-	UCI machine learning repository
87	OU-ISIR	HAR	Acc, Gyro	87	-
87	HAPT	HAR	Acc, Gyro	87	UCI machine learning repository
88	M HEALTH	HAR	Acc, Gyro, Mag, ECG	-	UCI machine learning repository
88	HAG	HAR	Acc, Gyro	88	-
89	HAG2	HAR	Acc	89	-
89	WISDM	HAR	Acc	76	-
90	Robita-gait	GA	Acc	90	-
91	HAG3	HAR	Acc	91	-
91	WISDM	HAR	Acc	76	-
92	CASIA-B	HAR	Cam	92	-
92	CASIA-C	HAR	Cam	92	-
125	CAR QUALITY CONTROL DATA SET	HAR, GR	Acc, Gyro, Comp, FSR, UWB Tag	52	-
126	CAR QUALITY CONTROL DATA SET	HAR, GR	Acc, Gyro, Comp, FSR, UWB Tag	52	-
167	Human Activity Recognition Using Smartphones Data Set	HAR	Acc, Gyro	289	UCI machine learning repository
202	Human Activity Recognition Using Smartphones Data Set	HAR	Acc, Gyro	289	UCI machine learning repository
292	DAPHNET	GA	Acc	-	UCI machine learning repository
294	DAPHNET	GA	Acc	-	UCI machine learning repository
295	Human Activity Recognition Using Smartphones Data Set	HAR	Acc, Gyro	289	UCI machine learning repository
303	M HEALTH	HAR	Acc, Gyro, Mag, ECG	-	UCI machine learning repository
303	PAMP2	HAR	Acc, Gyro, Mag, Temp, Orient, HR	-	UCI machine learning repository
305	Human Activity Recognition Using Smartphones Data Set	HAR	Acc, Gyro	289	UCI machine learning repository
305	WISDM	HAR	Acc	76	-
305	PAMP2	HAR	Acc, Gyro, Mag, Temp, Orient, HR	-	UCI machine learning repository
306	M HEALTH	HAR	Acc, Gyro, Mag, ECG	-	UCI machine learning repository
306	OU-ISIR	HAR	Acc, Gyro	87	-
306	HAPT	HAR	Acc, Gyro	87	-
306	UTD-MHAD	HAR, GR	Acc, Gyro	77	-
307	WISDM	HAR	Acc	76	-
307	Human Activity Recognition Using Smartphones Data Set	HAR	Acc, Gyro	289	-
307	PAMP2	HAR	Acc, Gyro, Mag, Temp, Orient, HR	-	UCI machine learning repository

name of audio input and output devices, the same condition is established, and this sensor is used in the paper [36].

Positioning and Tracking Sensors: This category includes sensors such as global positioning system (GPS) that uses different approaches such as position detection and location tracking [37]. These sensors are mostly used in location context pattern recognition and location-based movement classification. GPS sensors, Bluetooth beacons, UWB tags, and tracking sensors are the sensors placed in this category. GPS has been developed to enable accurate positioning and navigation anywhere on or near the surface of the Earth [38]. GPS sensors are often used for positioning and localization, providing geographical longitude, latitude, and height. However, their application in movement classification is more limited due to challenges such as the difference in sampling frequency with other sensors and occlusion in indoor environments, such as buildings and malls. It is expected that, due to the increasing growth of smartphones, this sensor will be used more in the field of activity recognition, GA, and GR. Devices that provide the ability to communicate with smart devices via Bluetooth are called beacons. Beacons are only signal transmitters; they only send signals to smart devices, such as smartphones. Many applications of Bluetooth beacons have been proposed for interested ones. These applications include indoor localization, proximity detection, and activity recognition [39]. They have been used for activity recognition according to Table XVII. Their signal strength reads are used for activity recognition via smartphones [40]. Ultrawideband (UWB) is an indoor positioning technology with several advantages over other related methods; one important advantage is providing long-term data on movement patterns without the influence of the presence of the observer [41]. Other sensors in this category are position-tracking sensors or simply trackers, which can calculate the positions and orientations of wearer subjects. These sensors are used in two papers according to the table. The paper [42] used two cyber gloves and two 3SPACE-position trackers for each hand to recognize Chinese sign language and perform some kind of GR. Tao et al. [20] have used an electromagnetic tracking system (ETS) that can calculate the positions and orientations of an object in the field of GA.

Optical and Light Sensors: These sensors detect and quantify various properties of light, such as intensity, frequency, wavelength, and polarization [43]. These sensors convert the light energy into an electrical signal output and include light and linear optical gesture sensor rows in Table XVII.

Proximity Sensors: A proximity sensor is a sensor that detects information about an object's presentation [44]. Capacitive sensors for measuring height and distance, infrared, and electric field sensors are the sensors that fit in this category.

Bend/Flex Sensors: Bend/flex sensors include bend sensors that are composed of a flexible substrate and a conductive layer, which changes its electrical resistance with the bending or the angular displacement [45].

Motion Detectors: These sensors include passive infrared and ultrasonic sensors in Table XVII. A motion detector is a sensor that detects nearby motion. Ogris et al. [46] have demonstrated how ultrasonic hand tracking can be used to improve the performance of a wearable, accelerometer, and gyroscope-based activity recognition system. In the paper [47], this sensor is used along with other sensors to learn the activities of the user with minimal user attention. If you check the table in the attachment carefully, you will notice that, in two papers [47] and [48], touch and light detection and ranging (LiDAR) sensors are used, respectively, and we have considered the category of these sensors as other categories. The first sensor is used to detect the physical touch of the object, and the second sensor is used to detect different ranges. Even though these sensors have been used much less than other sensors, it is not without grace to introduce them and get to know their performance in the three mentioned areas. Now, we introduce a quantitative analysis in general, and with approximate numbers, we introduce the most used sensors in the field of human motion analysis and the most used category in this field. With these numbers, it becomes easier to understand Fig. 2(a) and (b). In these two figures, the vertical axis represents the frequency of use based on the number of repetitions in the papers, and the horizontal axis is the sensor name and sensor categories, respectively. This contract is also valid for other similar figures. The names of the sensors in this article have been used a total of 358 times in the studied papers. Now, the sensors have been used in combination with each other, alone, or they have been mentioned as widely used sensors, and so on. The accelerometer with 138 repetitions is the most popular sensor name and, in a way, the most widely used wearable sensor. The percentage of accelerometer usage is about 39%. The gyroscope has been mentioned 71 times in all the studied papers, which is about 20% of the total value. The next sensor that is used the most is EMG, which has about 7% of the total value with 24 repetitions. Force and pressure sensors are ranked next with 5% and 4% of the total amount, respectively. The force sensor has been used in 17 papers and the pressure sensor in 14 papers. The next sensor is the temperature sensor, which is examined in nine papers and constitutes 3% of the total value. The magnetometer holds the same rank, and it is mentioned in nine papers; this sensor has something like 3% of the total amount, too. The light sensor and the microphone are used in seven and six papers, respectively, and have an equal share of about 2%. Other sensors are used less than six times and have been excluded from Fig. 2(a). Now, we present a detailed numerical analysis of the existing categories. The motion sensors category has been used 225 times and has a share of about 63% of all available categories. Bio and chemical sensors have been used 54 times and have a share of about 15%. The third category is pressure and force sensors, which has a share of 10% with 37 repetitions. Audio and visual sensors with nine repetitions have 3% of the total share. The positioning and tracking sensors category has the same number of repetitions in the papers and has a share similar to the previous category. Optical and light sensors with eight repetitions and proximity sensors with seven repetitions have an almost equal share of 2%. Other categories are used less than seven times and have been excluded from Fig. 2(b).

2) Available Datasets: The preprepared datasets in the studied papers are also presented in this article. Of course, these datasets are not necessarily related to IoT, but they are generally public datasets for use in movement classification

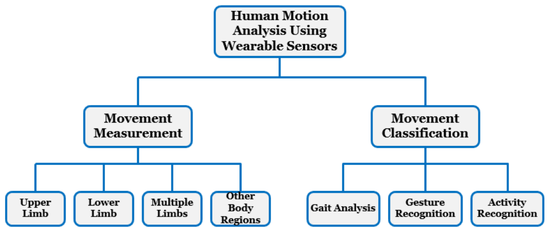


Fig. 1. Chart of human body motion analysis [6].

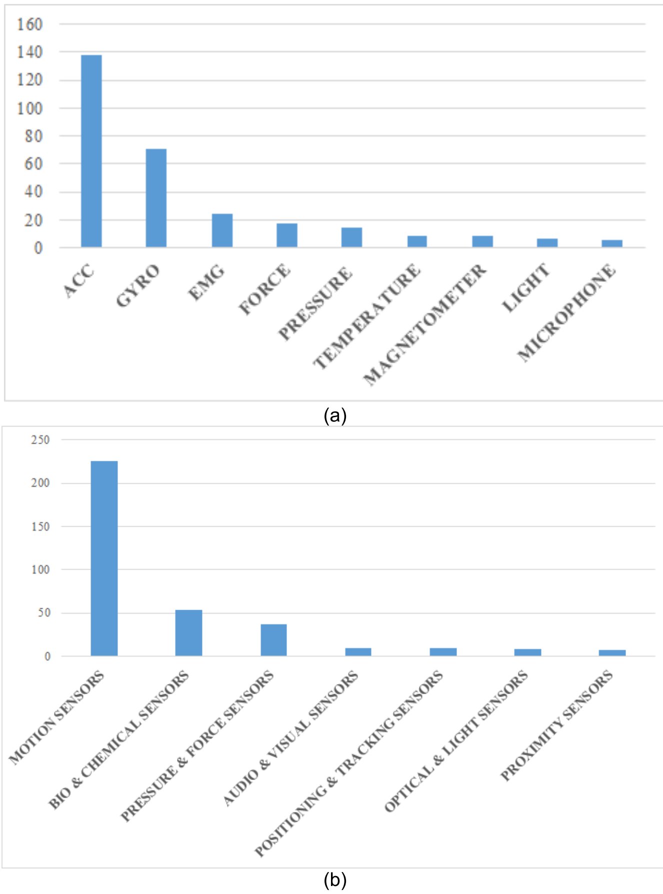


Fig. 2. (a) Most commonly used sensors based on the number of repetitions in the papers. Vertical axis: the number of times the sensors are used. Horizontal axis: the names of the sensors. (b) Most commonly used sensor categories based on the number of repetitions in the papers. Vertical axis: the number of times each sensor category is used. Horizontal axis: the names of the sensor categories.

by wearable sensors. Researchers can even find papers about preprepared datasets [49], [50].

As you can see in Table II, the UCI machine learning repository is one of the most important available data sources, and generating a dedicated dataset for a paper can also be a significant issue. The current explanations provided are very qualitative, and we need to describe the table related to this section in a little more detail. A total of 46 papers have used the specific dataset or have named their prepared dataset and made it publicly available. Some of the papers in this section are only for presenting the dataset and have not been used in the general reviews of the paper. This is because these datasets are very popular with those interested in the field and have a very high number of citations. The presentation of their other information does not have much effect on the charts, other tables, and numerical analyzes presented in this article. Before presenting the numerical analysis, we try to provide brief information about each dataset. First, we must announce that Table II has six columns that have names (reference number, datasets, application, sensors used, reference, and repository). The reference number column specifies the reference number that has used the mentioned dataset. The datasets column specifies the name of each dataset. The names of the datasets of the paper [19] have been used to name some of the datasets presented in this article. The application column also specifies which of the movement classification sections that each dataset belongs to, namely, GA, HAR, and GR. Movement classification sections generally overlap, the recognition of walking activities largely overlaps with GA, the recognition of body gestures can also be interpreted as the recognition of human activities, and the first overlap is less important than the second overlap and will not be discussed, but the second overlap is mentioned in the table. The column named sensors used fully specifies the sensors used to collect the respective dataset. The last columns specify the main reference as the dataset provider and the repository. The ActRecTut dataset is the result of an educational example for recognizing different hand gestures using inertial sensors attached to the upper and lower arms. The eight gestures in this article are opening a window, closing a window, watering a plant, turning the pages of a book, drinking from a bottle, cutting with a knife, chopping with a knife, stirring in a bowl, forehand, backhand, and smash [51]. Although this dataset is used to detect hand gestures, the authors somehow interpret its application as activity recognition. The Car Quality Control dataset is the result of the development and testing of real industrial activity tracking systems in the Škoda automobile factory and is somehow related to the fields of activity recognition and GR. Here, wearable sensors are attached to the hand too. The four activities of this article are inserting the lamp, mounting a supportive plastic bar using three screws and a cordless screwdriver, attaching the lamp body using two screws and a cordless screwdriver, and verifying the lamp's adjustment [52]. Providing a new method for continuous activity recognition based on ultrasonic hand tracking and motion sensors attached to the user's arms has led to the presentation of the Car Quality Inspection dataset [53]. Blanke et al. [54] propose a new type of feature based on the polynomial approximation of signals and report the results of their tests on the Car Quality Inspection, the Woodshop, and the Drink and Work datasets. The Car Quality Inspection dataset contains 20 activities performed during a car quality inspection. Example activities are checking gaps in the car's body or inspecting movable parts. The woodshop dataset contains data from eight different people's overall task of building two wooden book boxes. Building mentioned bookshelf needs a variety of activities, for example, sawing, drilling, or screw driving. In total, 22 activities are needed to get the job done. The Drink and Work dataset consists of several drinking events embedded in daily scenarios [54]. According to the UCI machine learning repository, the Mobile Health (MHEALTH) dataset used in [55] comprises body motion and vital signs recordings for ten volunteers. Activities in this dataset include standing still, sitting and relaxing, lying down, walking, climbing stairs, waist bends forward, the frontal elevation of arms, knee bending (crouching), cycling, jogging, and so on. The Daphnet dataset used in [56] is related to Parkinson's disease and analyzes the gait patterns of Parkinson's patients with fog symptoms. Users performed three kinds of tasks: straight-line walking, walking with numerous turns, going to different rooms while fetching coffee, opening doors, and so on. This information is fully contained in the UCI machine learning repository. Mahmud et al. [57] proposed a multistage long short-term memory (LSTM)-based deep neural network to integrate multimodal features from numerous sensors for activity recognition. For the training and evaluation of the proposed scheme, they have used a publicly available dataset from Physionet. This dataset contains wrist PPGs recorded during walking, running, and bike riding. Simultaneous motion estimates are collected using both accelerometers and gyroscopes to give multiple options for the elimination of motion interference from the PPG traces. A reference chest ECG is also used to allow a gold-standard comparison of heart rate during exercise. The description given for the HAR Using Smartphones dataset in the UCI machine learning repository is that this dataset is an activity recognition database built from the recordings of 30 subjects performing activities of daily living (ADLs) while carrying a waist-mounted smartphone with embedded inertial sensors. Each person performed six activities (walking, walking upstairs, walking downstairs, sitting, standing, and laying). This dataset is one of the most used datasets. Another dataset publicly available in the Physionet presented in the paper [58] contains measures of gait from 93 patients with idiopathic Parkinson's disease (66.3 years; 63% men) and 73 healthy controls (mean age: 66.3 years; 55% men). The PAMAP2 Physical Activity Monitoring dataset contains data on 18 different physical activities (such as walking, cycling, and playing soccer), performed by nine subjects wearing three inertial measurement units (IMUs) and a heart rate monitor. The dataset can be used for activity recognition and intensity estimation while developing and applying algorithms of data processing, segmentation, feature extraction, and classification. This information is provided in the UCI machine learning repository, and the dataset is used in the paper [59]. According to the UCI machine learning repository, the Daily and Sports Activities dataset provided by Bilkent University comprises motion sensor data of 19 daily and sports activities, each performed by eight subjects in their own style for 5 min. This dataset can be used for activity recognition [60]. Kawaguchi et al. [61] have started a project named "HASC Challenge" to collect a large-scale human activity corpus using accelerometers. The HASC dataset is the result of the research of the mentioned paper and is used for activity recognition and activities, including staying, walking, jogging, skipping, stair-up, and stair-down [62]. Chen et al. [63] have stated that the wearable motion capture device is used to take the kinematics data of the key nodes of the human body and fuse the data with the human skeleton data extracted from the video image by Openpose. In fact, in the mentioned paper, the Openpose dataset is used for GR, and the mentioned activities and gestures are squat, squat down and up, wave left hand, raise left hand, wave both hands, and raise both hands. The USC-HAD dataset is a Daily Activity dataset for activity recognition using wearable sensors; 12 different activities in this dataset are walking forward, walking left, walking right, walking upstairs, walking downstairs, running forward, jumping, sitting, standing, sleeping, the elevator up, and the elevator down. [64], [65]. The SHO dataset uses smartphone motion sensors for physical activity recognition. In the data collection experiments, they collected data for seven physical activities. These are walking, running, sitting, standing, jogging, biking, walking upstairs, and walking downstairs [66]. According to the UCI machine learning repository, the Opportunity dataset is a dataset designed to benchmark HAR algorithms. The dataset contains the readings of wearable motion sensors recording users' daily activities. It is useful for wearable activity recognition [67]. The Opportunity dataset comprises a set of complex naturalistic activities performed by four subjects in a daily living scenario performing morning activities. During the recordings, each subject performed a session five times with ADLs and one drill session. During each ADL session, subjects perform the activities without any restriction, and examples of activities are (preparing and drinking a coffee, preparing and eating a sandwich, and so on). During the drill sessions, each subject performed a predefined sorted set of 17 activities 20 times [67]. In the Skoda Mini Checkpoint dataset, one person performed activities in a car maintenance scenario using 20 accelerometers, placed on the left and right upper and lower arms. This dataset contains ten activity recordings, including writing on a notepad, opening the hood, closing the hood, checking gaps in the front door, opening the left front door, closing the left front door, closing both left doors, checking trunk gaps, opening and closing the trunk, and checking steering wheel [68]. The Actitracker dataset consists of triaxial accelerometer data samples. Subjects carried an Android phone in their front pants pocket and walked, jogged, ascended or descended stairs, sat, stood, and lay down for specific periods [69]. The Darmstadt Daily Routines dataset contains seven days of continuous data. The routines include dinner, commuting, lunch, and office work. Every routine contains various types of low-level activities; for example, dinner contains preparing food, eating dinner, and washing dishes [67], [70], [71]. The ubicomp08 dataset is recorded in the house of a 26-year-old man. Seven different activities were annotated, namely, leaving the house, toileting, showering, sleeping, preparing breakfast, preparing dinner, and preparing a beverage. Times during which no activity is annotated are referred to as Idle [72]. In paper [73], to evaluate their activity recognition model, they use two datasets called Bookshelf and Mirror. Bookshelf is a realistic dataset in a workshop scenario, in which subjects construct a wooden bookshelf. The dataset consists of a variety of activity events and types. The second dataset called Mirror is recorded and used in this article. Similar to the bookshelf, it contains a wide variety of activities, too. The first dataset is similar to the woodshop dataset described before [54], but they are used in different movement classification models. The Intel Research dataset is a dataset that contains various sensor data. The data have recordings of various human activities, such as sitting, walking, jogging, riding a bike, and driving a car [35], [74]. The MIT Place Lab dataset is recorded from a single subject wearing five accelerometers and a wireless heart rate monitor to perform a set of household activities. The activities include preparing a recipe, doing dishes, cleaning the kitchen, doing laundry, making a bed, and light cleaning around an apartment. In addition to the activities above, the subject also performs other everyday tasks such as searching for items and talking on the phone [65]. The UC Berkeley WARD dataset or simply the Wearable Action Recognition Database (WARD) dataset is developed by the University of California at Berkeley (UC Berkeley). WARD includes 20 human subjects (13 male and seven female) and a set of 13 activities, such as walking, standing, and jumping. The researchers have placed sensors at five body locations: two wrists, two ankles, and the waist. Each built multimodal sensor unit contains a three-axis accelerometer and a two-axis gyroscope [65]. The CMU-MMAC database was collected in the Carnegie Mellon University's Motion Capture Laboratory and contains multimodal measures of the human activity of subjects. The dataset focuses on cooking and food preparation. Wearable sensors that are used in data collection include a camera, an accelerometer, and a gyroscope. Five subjects performed cooking five different recipes: brownies, pizza, sandwiches, salad, and scrambled eggs, and related data were recorded [75]. Kwapisz et al. [76] presented a system that uses phone-based accelerometers to perform activity recognition and collected a dataset named wireless sensor data mining (WISDM) that contains labeled accelerometer data from 29 users as they performed daily activities, such as walking, jogging, climbing stairs, sitting, and standing. The paper [77] presents the UTD-MHAD dataset that consists of four different data modalities that include RGB videos, depth videos, skeleton positions, and inertial signals from a Kinect camera and a wearable inertial sensor for recording 27 human actions. 27 actions performed constitutes sports actions, such as bowling; hand gestures, such as drawing x; daily activities, such as knocking on the door; and training exercises, such as the squat. Wearable inertial sensors are placed on the right thigh and the right wrist. Stisen et al. [78] have recorded the HHAR dataset for detecting six different user activities: biking, sitting, standing, walking, stair up, and stair down. They have gathered data on nine users using smartphones and smartwatches. Smartphones were carried by the users around their waist, while smartwatches, were worn on each arm. The Sussex-Huawei Locomotion (SHL) dataset consists of multimodal transportation data, recorded by three individuals in eight different modes of transportation in real environments. Data were recorded using sensors of four smartphones located at the torso, backpack, hand, and pocket. The eight main activities in the dataset include standing or sitting, walking, running, biking, bus standing or sitting in a bus, driving and sitting in a car, and standing or sitting on a subway. The SHL dataset can be used in a wide variety of studies such as transportation recognition, activity recognition, mobility pattern mining, localization, tracking, and sensor fusion [79]. For the collection of the SARD dataset, the authors developed a data collection application for Android devices. This Android app currently collects data from the GPS, an accelerometer, a magnetometer, and a gyroscope at a rate of 50 Hz. They used four smartphones for data collection. Using these smartphones, they recorded data for six different physical activities, including walking, running, sitting, standing, and walking upstairs and downstairs. Four smartphones were located in four body positions (right jeans pocket, belt, right arm, and right wrist) [80]. Micucci et al. [81] propose a new dataset named UniMiB SHAR of acceleration samples collected using an Android smartphone designed for HAR and fall detection. The subjects placed the smartphone in their front trouser pockets: half of the time in the left one and the other half time in the right one. The dataset contains samples of nine types of ADLs, including running, sitting down, and so on, and contains samples of eight types of falls, including falling rightward, falling leftward, syncope, and so on in [82]; a mobile application (app) called ExtraSensory is developed, with versions for both iPhone and Android smartphones, and a companion application for the Pebble smartwatch that integrates with both. The ExtraSensory dataset contains data from 60 users, 34 of the subjects were iPhone users, and 26 subjects were Android users. The dataset contains data from various activities, such as walking, laying down, and bicycling. The phone was located in different places, such as in a bag, in hand, in a pocket, or on the table. Kyritsis et al. [83] propose a method for detecting food intake cycles during a meal using a wristband. They have presented a method that aims at detecting intake cycles. The FIC dataset contains acceleration data of eight subjects, and their proposed method detects five micro movements related to eating food. The WHARF dataset is presented as a freely available dataset of acceleration data, coming from a wrist-worn wearable device, targeting the recognition of 14 different human activities. Activities are mentioned in a table and include Brushing teeth, combing hair, getting up from the bed, lying down on the bed, and so on [84]. For the Smartphone-Based Recognition of Human Activities and Postural Transitions Dataset (SBRHA) collection, a group of 30 subjects was recruited. They were asked to perform six activities (walking, laying, sitting, climbing up the stairs, climbing down the stairs, and standing). The authors placed a smartphone on the waist and used it for the activity data recording using the built-in triaxial accelerometers and triaxial gyroscopes [85]. Liu et al. [86] present uWave, an efficient GR algorithm using a single triaxial accelerometer. They evaluate uWave with a gesture vocabulary identified by a VTT research for which they have recorded a library of 4480 gestures for eight gesture patterns from eight participants over multiple weeks. They have made the dataset open source. OU-ISIR and HAPT datasets are presented in the paper [87], which are datasets related to human activities, such as walking gathered by accelerometers and gyroscopes attached to the waist. The HAG dataset was collected from 50 subjects performing seven different activities in a controlled laboratory environment using an IMU sensor [88]. The HAG2 dataset is collected from 25 subjects using wearable IMU sensors for six different walking activities [89]. Raj et al. [90] have collected the robita-gait dataset of different gait using an accelerometer. For collecting the HAG3 dataset, 25 different subjects' data are collected for the identification of seven different walking activities using accelerometer readings [91]. The CASIA-B dataset contains the human walking pattern of 124 subjects collected using a camera, and CASIA-C contains 153 subjects and considers four variations of walking collected using the infrared camera that captures thermal images [92]. It can be easily shown that many preprepared datasets have set their agenda to identify walking activities because human walking styles, such as walking, running, and jumping, are an important field for activity recognition. Now that we have briefly introduced all the datasets, we want to perform a quantitative analysis of the information in the relevant table columns. Names ActRecTut, the Car Quality Inspection, the Woodshop, the Drink and Work, the Daily and Sports Activities dataset, HASC, Openpose, USC-HAD, SHO, Opportunity, the Skoda Mini Checkpoint dataset, Actitracker, Ubicomp08, Bookshelf, Mirror, Intel Research, the MIT Place Lab dataset, the UC Berkeley WARD Dataset, CMU-MMAC, HHAR, SHL, SARD, UniMiB SHAR, ExtraSensory, FIC, WHARF, SBRHA, HAG, HAG2, HAG3, robita-gait, CASIA-B, CASIA-C, and uWave are each used in only one paper. From the Physionet website, two separate datasets have been used once each. The Car Quality Control dataset, OU-ISIR, HAPT, and UTD-MHAD are used twice. The Daphnet dataset and the Darmstadt Daily Routines dataset are used in three papers. PAMP2 is used four times. WISDM is used five times. MHEALTH has been used six times. HAR Using the Smartphones dataset has been used seven times. 46 separate datasets have been introduced, 12 of which are related to the application of (HAR and GR), 30 of which are related to HAR, one is related to GR, and three of which are related to GA. The use of (HAR and GR) is usually related to the activities that are done by the subject's hand and can be interpreted as a gesture. This is clear from the description of each dataset. By checking the sensors used column, we get interesting results. The definition of sensor categories in Section II-A1 is valid here as well. Only in this section, there are some sensors that were not used in Section II-A1, and we will have a brief overview of them. The new sensors presented in this section are magnetic field sensors, linear acceleration sensors, real-time clocks, tilt switches, and IR/V light sensors. Magnetic field sensors and linear acceleration sensors are motion sensors that act in a way like magnetometers and accelerometers, respectively, and generally, are not considered separate sensors from them. The real-time clock is generally not included in the work scope of this article and is not addressed, but, due to the respect of the producers of the datasets, it is only present in the according table and is not present in the statistical analysis. Tilt switches or tilt sensors, sometimes called inclinometers, are used for measuring the angles or tilts of objects. Infrared light sensors are used to detect infrared light emitted by individuals or objects and are not capable of detecting visible light. The visible light sensor does not need a special definition. As in Section II-A1, we perform numerical analysis on the number of sensors used, then place the sensors in the mentioned categories, and analyze the results seriously. The accelerometer has been repeated 67 times and has been ranked first again. The second place, as before, belongs to the gyroscope with 43 uses. The magnetometer has been used 21 times and ranks third. The fourth place goes to the temperature sensors with eight uses. The ECG ranks fifth with seven uses. Camera, light, and heart rate are ranked sixth with five uses each. The orientation sensor and compass are ranked seventh with four uses, followed by tilt switches that are used three times. The FSR force sensor along with the UWB tag, pressure sensor, microphone, and GPS is ranked ninth. All these sensors have been used twice. The magnetic field sensor, ultrasonic transmitter (sensor), PPG, touch, linear acceleration sensor, proximity, audio, and humidity each with only one use is ranked tenth. The accelerometer and gyroscope sensors have kept their position, and to some extent, these results confirm the validity of the above results. Now, we specify the sensors that are placed in each category. Sensors placed in the motion sensor category are accelerometers, gyroscopes, compasses, magnetometers, magnetic field sensors, orientation sensors, linear acceleration sensors, and tilt switches. This category has been used 144 times. The sensors that make up the category of biological and chemical sensors are temperature, ECG, HR, PPG, and humidity. This category has been used 22 times. The sensors that make up the category of audio and visual sensors are the camera, microphone, and audio sensor. These sensors have been used a total of eight times. Optical and light sensors, including light sensors, IR/V light sensors, and HF-light sensors, are used five times. Position and tracking sensors' categories include UWB tags and GPS. This category is used four times. The category of pressure and force sensors has been used four times again. Motion detectors with an ultrasonic sensor as their representative along with proximity sensors are used only once. The other categories, which include touch sensors, were repeated one time, too. Bend sensors are not used in preprepared datasets. As you can see, the first and second places go to the categories of the motion sensor and bio and chemical sensors as in the previous section. Because the total number of sensors used in this section was 193 and was much less than the number of sensors in Section II-A1, we refused to provide the percentage share. Also, due to the smaller number of sensors, unlike Section II-A1, we provided a complete statistical analysis. The similarity of the results of these two sections confirms the validity of the presented results, and in a way, the analysis results of Section II-A1 are confirmed.

B. Preprocessing

Preprocessing actions convert raw data into a suitable and preferred format for data processing and further analysis, and improve the quality of the dataset. These preprocessing actions were identified from studied papers, and segmentation and feature extraction are considered separate steps in this article, but, due to respect for the authors who consider it as a member of preprocessing, they have been identified as preprocessing in this section. Preprocessing actions are fully presented in Table III. For a better understanding of this table, we first briefly define each preprocessing action defined in the

TABLE III
PREPROCESSING ACTIONS

Reference number	Preprocessing actions
59,67,125,135,248,253,285	Interpolation with the aim of replacing lost or missing values
31,56,88,95,96,97,99,101,104,106,114,118,121,122,124,140,143,145,156,158,159,167,169,171,172,174,176,192,193,196,197,200,204,213,220,222,223,225,231,239,240,241,244,246,248,249,251,253,260,263,266,267,269,274,276,278,280,285,288,289,291,292,294	Filtering mainly for noise removal
67,87,144,156,172,173,192,208,227,235,237,248,249,267,274,291,292,306,307,308	Data normalization
106,114,140,159,174	Drift removal
103,106,124,169,196,249,276	rectification
94,95,96	Calculating signal magnitude
97,98	Truncating, and trimming data
98,200,253,291,292	labeling
103,125,196,208,211,217,262	Smoothing
99,103,136,143,200,208,255,291	Segmentation
99,100	Creating an extra dimension or new dimensions
32,135,139,156,200,211,251,262,263,274,282,290	calibration
200,253	Feature normalization
101	Providing the same scale for raw data
31,32,118,136,222,223,225,244,269	amplification
31,144,172,267	Wavelet transform mainly for noise reduction
103,104,105	Removing the offset or DC component
160,161,292	Feature extraction

table, specify its use, and then proceed to provide statistical analysis. The first row presented in the relevant table belongs to interpolation and papers that used this preprocessing action. The main reason for using this preprocessing action is to fill in unknown or missing data values. Increasing the sampling rate of the sensor leads to the creation of data with unknown values. Of course, this is not the only reason for creating missing or unknown data. It may happen due to sensor signal loss, failure of sensor equipment, and many other reasons. For example, GPS data may be lost when entering a building. The next preprocessing action is filtering. Whenever the discussion of filtering in sensors comes up, it is unconsciously referred to as the discussion of noise in the sensor output. Sensor noise in signal processing is a general term for unwanted and unknown modifications that a signal may suffer during capture, storage, transmission, processing, or conversion [93]. This definition is very general, and in this section, we do not intend to discuss the noises in different wearable sensors and different filtering methods. The third preprocessing action mentioned in the table of this section is data normalization. Data normalization is actually a method that converts data to the same scale and maps them all to the same range. The main purpose of this preprocessing action is to reduce the redundancy of the data, and in fact, it makes the data consistent, that is, the data from different sensors with very different values in the records have the same range. If the sensor output changes independently for the same value as the input, drift has occurred. Physical changes in the long term cause drift, and it must be removed. The next preprocessing action is rectification. By rectifying the output signal of the wearable sensor, its positive or negative part is practically removed. The rectifier has two types: full wave and half wave. The next preprocessing action is calculating signal magnitude, which is not as well-known as the previous preprocessing actions. This preprocessing action is used in three papers [94], [95], and [96]. In the papers [94] and [95], just before extracting the features, the magnitude of the accelerometer signal was calculated, and then, the features were extracted from this value; in the paper [96], this action was done on the gyroscope in addition to the accelerometer. Truncating and trimming data preprocessing actions have been used in two papers [97] and [98]. It is somehow related to the concept of labeling, which is necessary for classifier training. In these papers, the data are limited to the beginning and end of the video clips associated with the labeling. Labeling is associated with the training phase of classifiers and creates labeled data for classifier training, which will be explored in the following. In some papers, as indicated in the table, labeling is considered a preprocessing action. Smoothing is a method to adapt to long-term changes in the output of sensors and, at the same time, smooth out short-term changes in the output. Smoothing makes it easier to follow important data patterns. Signal segmentation in the papers in the segmentation row in the table is considered a part of the preprocessing, but, as we have already announced, we will fully examine it as a separate step. The next preprocessing action is creating an extra dimension or new dimensions. This preprocessing action has been used in two papers [99] and [100]. In the paper [99], they create a new dimension with a special formula, such as signal magnitude, and extract the features from four dimensions. Li et al. [100] create four composite axes from the three main axes of each sensor and extract features again. Calibration is a widely used preprocessing action. Sensors must be calibrated to increase accuracy, that is, calibration must be done on the sensor so that the sensor works as accurately as possible. Feature normalization, such as data normalization, actually prepares features with different scales for use in machine learning models. Providing the same scale for raw data, as it is clear in the table, it has been used in only one paper [101]. Because the sensor data from different participants are different in terms of amplitude, they have provided a special formula that standardizes the raw data. To define signal amplification, we are satisfied with this general definition, signal amplification causes the output to be larger and its value to be proportional for subsequent applications, and, in a way, it will increase the signal strength. Wavelet transform (WT) mainly for noise reduction preprocessing action acts like filtering in terms of application, but, since it uses a special filter to reduce noise, we will deal with it separately. Research to find new methods of noise removal is still ongoing; a WT is a powerful tool for this field, and its combination with other noise removal methods will improve performance [102]. Most of the time information can be seen in the frequency domain much easier than in the time domain. It is very important to get the time–frequency characteristics of nonstationary signals. WT is particularly suitable for noise removal in these cases [102]. Removing the offset or distributed classifier (DC) component is the last preprocessing step because feature extraction will be explained fully later. The DC component or DC bias is the average amplitude of the waveform usually in the time domain, and the sensor offset means that the sensor output is higher or lower than the original value. In papers [103] and [104], they removed the DC component by subtracting the mean of signals from the raw data of the sensors. Hegde et al. [105] have presented a new signal preprocessing methodology to eliminate the offset of insole pressure sensors. Now, we provide a general quantitative analysis of preprocessing actions and the number of uses of each. A total of 163 times the mentioned preprocessing actions have been used. Filtering mainly for noise removal has been the most used preprocessing action with 63 repetitions and a share of about 39%. The second place goes to data normalization, which has a share of about 12% with 20 uses. Calibration with 12 times of use has a share of about 7%. Amplification is in the next rank with nine times of use and a share of about 6%. Segmentation has a share of about 5% with eight uses. Each of interpolation, rectification, and smoothing has a share of 4% with seven uses. Drift removal and labeling have a share of 3% with five uses. WT mainly for noise reduction, with four times of use, has a share of 2%. Calculating signal magnitude, removing offset or DC component, and feature extraction have been used three times each and have a share of about 2%. Truncating and trimming data, and creating an extra dimension or new dimensions along with feature normalization have a small share of 1% with two times of use each. For providing the same scale for raw data, we consider a 1% share by being it in only one paper. As it is known, a total of 18 preprocessing actions are known in the papers, a total of 163 times have been repeated in the corresponding table, and the share of using each and the number of times of use for each action were determined. Fig. 3 is provided for a better understanding of this numerical analysis and visually compares the number of times each preprocessing action is used.

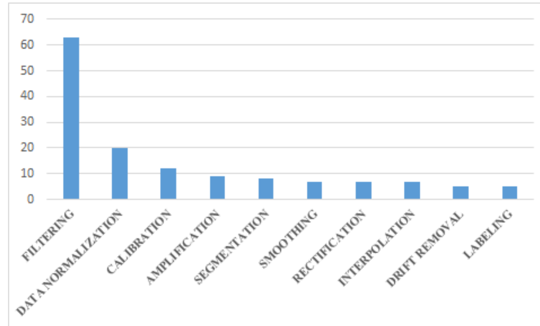


Fig. 3. Most commonly used preprocessing actions based on the number of repetitions in the papers. Vertical axis: the number of times the preprocessing actions are used. Horizontal axis: the names of the preprocessing actions.

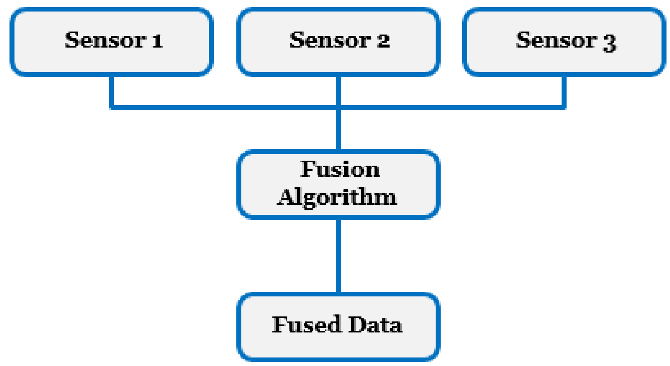


Fig. 4. Data-level fusion.

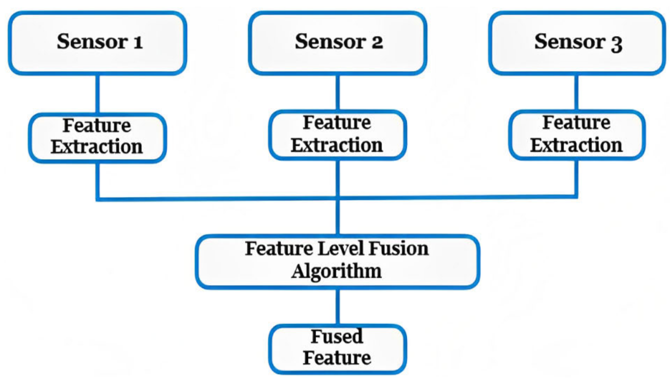


Fig. 5. Feature-level fusion.

C. Data Fusion

Data fusion is the process of combining data from different sensors to have a single data. After reading several papers, three levels of fusion are considered for the data: 1) data level; 2) feature level; and 3) classification level [106], [107], [108], [109], [110], [111], [112]. To define each of these levels, we use the figure drawing. However, we will briefly introduce some famous algorithms for these levels, or we will introduce a reference paper for more familiarity with related algorithms. Figs. 4–6 show the data-level fusion, the feature-level fusion, and the classifier-level fusion, respectively. According to the studied papers, estimation algorithms are mostly used for level 1. When we hear the word estimation, we remember the Kalman filter. Kalman filter and its nonlinear variants, i.e., extended Kalman filter (EKF) and unscented Kalman filter, are recursive filters for estimation. The first two will be discussed in detail. The weighted average fusion algorithm is one of

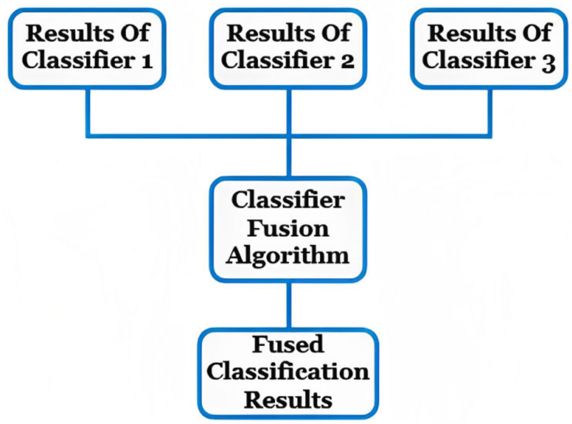


Fig. 6. Classifier-level fusion.

the easiest and most popular algorithms for data-level fusion, which takes the weighted mean of the redundant information from multiple sensors as the fused value. The next data-level fusion method is concatenation, which concatenates the raw data of sensors; for example, accelerometer, gyroscope, and EMG sensor data can be concatenated into a single input vector [106]. The least-squares method is a nonrecursive estimation method and is normally used only to merge redundant data [108]. This method works by minimizing the squared difference between the observed data and the expected values. Particle filter somehow acts like Kalman filter and estimates the posterior distribution of the state of the dynamical system conditioned on the data. Classification algorithms are the most widely used methods for level 2, and inferential algorithms are used in level 3. Now, we introduce some of the famous classification algorithms for use at the feature-level fusion. These algorithms include SVM, k-nearest neighbor (KNN), k means, LSTM, and ELM. The support vector machine (SVM) is a supervised learning method. A standard SVM is a binary linear classifier. The key idea is to generate an optimized discriminant, hyperplane, to classify the training data into two classes. The optimal hyperplane in SVM means that the classification has minimum errors and the maximum margin between the two classes [108]. Assigning a point to one of the $+1$ or -1 classes is done using a linear classifier function. To classify more than two classes with SVM, we need to change the structure, which is known as one versus all. The KNN algorithm is a very widely used supervised classification algorithm. The algorithm calculates the distance between the unknown data and all data samples usually using the Euclidean distance function, but it also can use distance functions, such as Minkowski, correlation, and Chebyshev. Then, select the K closest samples to the unknown sample, and according to what class the neighbors of this data belong to, it determines the class of the unknown data by voting. Unlike KNN, k means is an unsupervised method. In this algorithm, K is the number of clusters defined in advance by the user, and when the algorithm comes across unknown or unlabeled data, it selects the final data class from K clusters through an iterative method. In fact, in this algorithm, the final answer is determined through the minimization or maximization of an objective function. The LSTM network is a special type of recurrent neural network (RNN). RNN is a type of feedforward neural network that has internal memory. This network is a neural network that has a loop in its structure through which the output of the previous step is entered into the network along with the new input. This feature makes it able to work with sequential data. RNNs may struggle with long-term dependencies. LSTMs, which have built-in cell states and gates, i.e., the forget gate, the input gate, and the output gate to control the flow of information, solve the long-term memory problem of the RNN network and capture dependence at different time intervals. An extreme learning machine (ELM) is a special type of single-layer feedforward neural network. Unlike in traditional feedforward neural networks where training the network involves finding all connection weights and biases, in ELM, connections between input and hidden neurons are randomly generated and fixed, which means that they do not need to be trained. Thus, training an ELM becomes finding connections between hidden and output neurons only, which is simply a linear least-squares problem whose solution can be directly generated by the generalized inverse of the hidden layer output matrix [26]. By applying Mercer's condition to traditional ELM, a kernel ELM (KELM) is obtained [26]. Traditional ELM has less generalizability than KELM. To address the issue of imbalanced classwise data distribution, a weighted ELM (WELM) can be used [26]. ELM may cause the overfitting problem and also does not perform well in the presence of outlier data; that is why regularized ELM is introduced to solve such problems. The variants of ELM, including KELM, WELM, and regularized ELM, are used directly to handle the multiactivity classification problem, without involving the one-versus-one or one-versus-all method [26]. Interested parties should refer to the paper [112] to learn about some of the famous classification-level fusion algorithms. Anyway, we will also introduce some of the most famous and widely used classification-level fusion algorithms. Majority voting is one of the most famous classifier-level fusion methods, which is a member of the family of voting methods, However, this algorithm has been examined separately both in statistical analysis and performance introduction [112]. In this method, there are several classifiers in the system, each providing a single class label; the algorithm sums the predictions for each label and selects the label with the majority vote. The Bayesian fusion methods can be applied to the classification-level fusion under the condition that the outputs of the classifier are expressed in posterior probabilities [112]. Bayesian fusion methods include algorithms that work based on the Bayesian theory; these algorithms can be used both in feature-level fusion and classification-level fusion. Boosting algorithms are algorithms that combine several weak classifiers and create a stronger classifier. These algorithms have led to the production of new classification algorithms. Adaboost and Gradient boosting are such algorithms that are usually created by combining decision trees. In general, joint boosting is very similar to boosting in terms of performance and structure [51] and is not considered separately in the statistical analysis. These algorithms are a bit more complicated than boosting, and as a result, they are slow algorithms, but they are more robust against uncertainties. Stacking algorithms, which are one of the most widely used algorithms at this level, are algorithms in which the classification results of several classifiers are provided to a metaclassifier that determines the final classification result. Fuzzy integral algorithms, fuzzy template algorithms, Dempster–Shafer methods, products of experts, and neural networks have similar performances to Bayesian fusion methods. These algorithms operate on classifiers that produce so-called soft outputs. The outputs are the real values in the range [0, 1]. These values are referred to as fuzzy measures, which cover all known measures of evidence. Measures of evidence are used to describe different dimensions of information uncertainty. These algorithms try to reduce the level of uncertainty maximizing suitable measures of evidence [112]. The intersection of Neighborhoods and Union of Neighborhoods are based on a class set reduction, and their objective is to reduce the set of considered classes to as small a number as possible but ensure that the correct class is still represented in the reduced set. These algorithms try to find the tradeoff between minimizing the class set and maximizing the probability of inclusion of the true class [112]. The highest rank method, the Borda count method, and logistic regression aim at a class set reordering to obtain the true class ranked as close to the top as possible. These algorithms try to improve the overall rank of the true class [112]. It is not bad to know the strengths and weaknesses of these algorithms. An advantage of the highest rank method is that it utilizes the strength of every single classifier, which means that, as long as there is at least one classifier that performs well, the true class should always be near the top of the final ranking. The weakness is that combined ranking may have many ties, which have to be resolved by additional criteria. The Borda count method is easy to implement. The weak point of this technique is that it treats all classifiers equally and does not take into account individual classifiers' capabilities. This disadvantage can be reduced to a certain degree by applying weights and calculation of the Borda count as a weighted sum of a number of classes. The weights can be different for every classifier, which, in turn, requires additional training [112]. The Borda count method does not recognize the quality of individual classifiers' outputs. An improvement can be achieved by assigning the weights to each classifier reflecting their importance in a multiple-decision system and performing so-called logistic regression [112]. All these proposed algorithms, i.e., Intersection of Neighborhoods, Union of Neighborhoods, the highest rank method, the Borda count method, and logistic regression, may be applied to the same problem so that the set of classes is first reduced and then sorted [112]. A bagging algorithm creates a metaclassifier that runs each of the constituent classifiers on random subsets of the target dataset and then aggregates their predictions to form a final decision. Dynamic classifier selection, classifier structuring and grouping, and the hierarchical mixture of experts operate on the classifiers rather than their outputs, trying to improve the classification rate by pushing classifiers into an optimized structure [112]. The hierarchical mixture of experts does not seem to be applicable to high-dimensional data because high-dimensional data can lead to increased variance and numerical instability [112]. Voting methods, of which majority voting is one of the main algorithms, are similar to the behavior-knowledge space (BKS) method in terms of performance. Classifiers producing crisp, single-class labels provide the least amount of useful information for the fusion process. The fusion process with these classifiers can be upgraded by voting methods [112]. Voting strategies can be applied to multiple classifier systems assuming that each classifier gives a single class label as an output. There are several approaches to the combination of such uncertain information units to obtain the best final decision. However, they all lead to the generalized voting definition [112]. The BKS method can efficiently aggregate the decisions of individual classifiers. This method provides a K-dimensional knowledge space by collecting the records of the decisions of all K classifiers for each learned sample, then combines decisions generated from individual classifiers, and enters a BKS method unit of the mentioned space. A unit of BKS is an intersection of decisions of every single classifier and makes a final decision by a rule that estimates the balance between the current classifiers' decisions and the recorded behavior information the knowledge in the BKS unit [112]. Now that we are familiar with the performance of famous and widely used fusion algorithms of every level, we intend to specify more precisely the algorithms that can be used at each level. Papers that have used these levels directly with mentioned names or mentioned the fusion algorithm precisely are listed in Table IV. Also, the papers that have used data fusion without presenting the level are not present in this table and are not counted among the final statistics. In this table, d, f, and c stand for the data level, the feature level, and the classifier level, respectively. Now, we present a detailed numerical analysis of the algorithms used at each level. First, we start with the data level. Eight papers directly refer to this level of data fusion and present 13 algorithms. A total of seven papers have mentioned the Kalman filter algorithm for data fusion at this level. Six papers mentioned the Kalman filter algorithm, and one paper mentioned both the Kalman filter and the EKF algorithms. Therefore, the Kalman filter with seven times of use has a share of about 54%. The weighted average method along with concatenation each has a share equal to 15% by being used only twice. The least-squares method and the particle filter have the least number of uses and have a share of about 8% with one use. Feature-level fusion has been proposed in 14 papers. In total, these 14 papers have proposed 26 algorithms for data fusion at this level. SVM is one of the leading algorithms at this level with three times of use and a share of about 12%. According to the table, three different variants of ELM have been used in only one paper for feature fusion. These variants are KELM, WELM, and regularized ELM. Therefore, this fusion algorithm is one of the most widely used fusion algorithms at the feature level, with three uses and a 12% share. The k means, KNN, and LSTM derivatives (bi-LSTM and stack of LSTM layers) have been used twice and have a share of 8%. All subsequent feature fusion algorithms have been used only once and have a share of 4%; these algorithms include concatenation, cluster analysis, Kohonen feature map, learning vector quantization, artificial neural network (ANN), decision tree, GMM, PCA, CCA, combining features into a single matrix, conditional random field (CRF), CNN, a score-based sensor fusion scheme, and the fuzzy logic. Fusion at the classifier level has been proposed in 27 papers. A total of 72 algorithms have been proposed, which are divided into 35 distinct algorithms. In the meantime, majority voting is the most widely used classifier-level fusion algorithm with ten uses and a share of 14%. After that, Bayesian approaches with seven uses have a 10% share. The terms that specify these algorithms in the relevant table are Bayesian inference, Bayesian inference, such as naïve Bayes, Bayesian fusion methods, Bayesian fusion, naïve Bayes combiners (NBCs), Bayesian framework, and Bayesian inference. The third place goes to boosting with six times of use in papers and a share of about 8%. The closest follower of boosting is the fusion method named stacking. This method is used only once less than the previous method and has a share of about 7%. Fuzzy methods have also been used four times (6% share); in total, three papers have used these methods, and according to the table, one paper has mentioned two different methods. The terms that are referred to as fuzzy methods are fuzzy, fuzzy logic, fuzzy integrals, and fuzzy templates. The Dempster–Shafer method is ranked next with three uses and has a small percentage of 4%. The Borda count method also has the same conditions. Neural networks, highest rank, logistic regression-based methods, bagging, hierarchical weighted decision (HWD), and class-based weighted fusion all have a share of about 3% with two times of use. Now, we specify more precisely the terms that are included in some of these methods. Two variants of HWD have been used in only one paper. The terms presented in this method in general numerical analysis are HWD and a novel HWD algorithm, called DC. Class-based weighted fusion has the same conditions as the previous one, and the terms presented in this method in general numerical analysis are posterior-adapted class-based weighted fusion and class-based weighted fusion. All next algorithms used at this level are used only once and have a share of 1%. These algorithms include average output, genetic algorithms (GAs), evolution algorithms, topic models, equal weight fusion, recall combiners, body multipositional decision selection, plurality voting, an average of probabilities, dynamic classifier selection, classifier structuring, grouping, a hierarchical mixture of experts, voting methods, BKS method, Intersection of Neighborhoods, Union of Neighborhoods, the product of experts, summation, the logarithm opinion pool (LOGP) technique, hierarchical decision (HD), model-based fusion, a decision tree, and multistream hidden Markov models (HMMs).

D. Signal Segmentation

Segmentation is used frequently in papers and identifies important information in the preprocessed dataset. To define signal segmentation precisely, we use the definition of [113]. Azami et al. [113] describe signal segmentation as follows: "signal segmentation is the act of splitting a signal into smaller parts that each has the same statistical characteristics, such as amplitude and frequency." In [51], it is stated that segmentation can be done using the following approaches: 1) sliding window; 2) energy-based segmentation; and 3) additional sensors and contextual sources. Of course, other approaches in addition to these three methods are presented in the papers. In this section, we present the segmentation algorithms used in papers. Before providing a comprehensive numerical analysis of the number of each algorithm or approach used, we first provide a brief description of the performance of each member of the column named segmentation algorithm used in Table V. The first and perhaps the most widely used approach in this field is the sliding window. In the sliding window approach, a window is moved over the time series data to "extract" a portion of the data that can be used in subsequent processing steps [51]. Energy-based segmentation relies on the fact that different activities are performed with different intensities. The intensity difference is directly related to the different energy levels of the sensor signals. The signal energy (E) is calculated through the signal energy formula. By thresholding on E, data segments belonging to the same activity can be found [51]. Additional sensors and contextual sources that we simply refer to as additional sensors are the third approach discussed for segmentation. Sensor data recorded with one modality can be segmented through information derived from other modalities [51]. For example, using GPS traces, acceleration data recorded using mobile phone accelerometers can be segmented [51]. The head-based segmentation scheme, which is proposed in two papers, was proposed by Bulling et al. [114]. They developed a segmentation approach that requires only a single-axis accelerometer on the head. Their segmentation is based on two hypotheses. First, the reading happens only when the subject's head is down. Second, the up and down movements of the head can be detected using the mentioned accelerometer. They detect these head movements by thresholding the x-component of the denoised, mean-subtracted head acceleration signal [114]. Blanke et al. [73] use a segmentation technique that replaces the standard sliding window approach. This segmentation technique is based on the human body model. Assuming low-motion moments at the beginning and end of interactions, segments of interest are created using such points [73]. Symbolic aggregate approximation (SAX)- and GA-based approaches are proposed in the paper [115]. The former approximates a given time series by piecewise constants encoded in a discrete alphabet, and the latter uses evolutionary search to find a suitable segmentation. The segmentation approach proposed in the paper [116] is obtained by thresholding the acceleration variance and pairwise combined segments. In the paper [117], a rectangular window function with a window length of 4 s is used for segmentation. The windowing itself is done in steps of 0.5 s. Khairuddin et al. [118] have segmented the EMG signals into two distinct sections: preintention and intention. Their purpose was to identify the intention of the movement. The intention signal is recorded based on the definition of a muscle burst that transpires between 40 and 100 ms prior to any muscle activities. In the paper [119], the autocorrelation function (ACF) is used for the segmentation of accelerometer data by devising a concept of tuning parameters that are based on minimum standard deviation. It does not seem necessary to provide numerical analysis in this section because the sliding window is at the

TABLE IV
FUSION STRATEGY AND METHODS USED

Reference number	Fusion strategy	Algorithms
20	data level fusion	Kalman Filter
26	feature level fusion	kernel ELM, weighted ELM, regularized ELM
27	feature level fusion	SVM
36	classifier level fusion	Bayesian Inference
40	feature level fusion, classifier level fusion	f=Conditional Random Field (CRF) based supervised activity classifier, c=Body Multi-positional Decision Selection
46	classifier level fusion	majority decision
51	classifier level fusion	ensemble classifiers or boosting, summation, majority voting, Borda count, Bayesian fusion
56	classifier level fusion	majority voting
57	feature level fusion	stack of LSTM layers
59	classifier level fusion	posterior-adapted class-based weighted fusion, model-based fusion, class-based weighted fusion
71	classifier level fusion	Joint boosting
99	classifier level fusion	an ensemble model of classifiers (NN) based weighted majority voting
106	data level fusion, feature level fusion, classifier level fusion	d=Concatenation, f=Concatenation, c=Average output
108	data level fusion, feature level fusion, classifier level fusion	d= Weighted-Average Method, The Least-Squares Method, Kalman Filter, Extended Kalman Filter, f= (SVM), Cluster Analysis, K means, Kohonen Feature Map, Learning Vector Quantization (LVQ), c= Neural Networks, Genetic Algorithms, Evolution Algorithms, Fuzzy, Bayesian Inference, The Dempster–Shafer Method
109	classifier level fusion	Borda count, highest rank, and a method using logistic regression
110	classifier level fusion	(majority/plurality voting, average of probabilities, boosting, bagging, stacking)
111	data level fusion, feature level fusion, classifier level fusion	d= weighted averages, Kalman filter and its nonlinear derivatives, particle filter (PF), f= ANN, SVM, Decision tree, KNN, GMM, K means, c= Bayesian inference such as naïve bayes, Dempster – Shafer theory, fuzzy logic, and topic models
112	classifier level fusion	Dynamic Classifier Selection, Classifier Structuring and Grouping, Hierarchical mixture of experts, Voting Methods, Behavior-Knowledge Space Method, Intersection of Neighborhood's, Union of Neighborhood's, The Highest Rank Method, The Borda Count Method, Logistic Regression, Bayesian Fusion Methods, Fuzzy Integrals, Dempster-Shaffer Combination, Fuzzy Templates, Product of Experts, Artificial Neural Networks
123	classifier level fusion	based on stacking framework
125	classifier level fusion	Joint boosting
126	classifier level fusion	Joint boosting
137	classifier level fusion	the logarithm opinion pool (LOGP) technique
143	classifier level fusion	Hierarchical Decision (HD), Majority Voting (MV), Hierarchical Weighted Decision (HWD), novel HWD algorithm, called "Distributed Classifier" (DC)
145	classifier level fusion, feature level fusion	c=equal weight fusion, majority voting (MV), recall combiners (RC), and Naïve Bayes (NBC)combiners, f=bi-LSTM
159	feature level fusion	PCA, CCA
160	classifier level fusion	majority voting, stacking
161	classifier level fusion	majority voting, stacking
162	classifier level fusion	majority voting, stacking
190	classifier level fusion	Bayesian Framework
193	feature level fusion	combining features into a single matrix
200	data level fusion	Kalman filter
201	data level fusion	concatenation
203	classifier level fusion	Bagging, boosting
205	classifier level fusion	A decision tree and multi stream hidden Markov models
208	feature level fusion	a convolutional neural network-based feature fusion strategy
211	feature level fusion	a score-based sensor fusion scheme
253	data level fusion	Kalman filter
257	classifier level fusion	Majority voting
263	data level fusion	Kalman Filter
265	feature level fusion	KNN
266	feature level fusion	fuzzy logic

top by a large margin compared to other approaches. Despite this general analysis, a more detailed numerical analysis is not without grace. Papers have announced their used segmentation approach a total of 52 times. The sliding window approach with 34 uses and a 65% share is the most used segmentation approach. The second place is dedicated to energy-based approaches with seven uses and a 13% share. Additional sensors with two times of use and a share of 4% along with the head-based segmentation scheme are ranked third. Other algorithms and approaches are ranked next by being used in only one paper and about 2% share each.

E. Feature Extraction

In this section, we present feature extraction methods, the predominant type of features mentioned in papers, and the domain of features of papers. Of course, it is also possible for data to be used raw, and some newer machine learning methods, namely, deep learning methods, automatically extract features and do not require handcrafted features. First, we describe the feature extraction methods presented in Table VI. The spectral analysis feature extraction method for extracting the features of the frequency domain is described in the paper [27]. This method can also be

TABLE V
SEGMENTATION ALGORITHMS

Reference number	Segmentation algorithm
40,46,48,56,59,62,70,96,98,99,100,109,110,114,124,145,151,167,169,195,202,213,221,222,224,241,252,261,279,291,294,303,305,307	Sliding window
115,136,158,205,206,211,212	Energy based approaches like (swab, etc.)
205,211	Additional sensors
109,114	Head based segmentation scheme
73	segmentation technique based on human body model
115	sax
115	genetic algorithm (GA) based approach
116	segmentation by thresholding the acceleration variance and pairwise combined segments
117	rectangular window function
118	Muscle burst-based segmentation
119	Auto correlation function

TABLE VI
FEATURE EXTRACTION METHOD

Reference number	Feature extraction method
27,120	Spectral analysis
31,54,56,74,96,100,120,122,126,138,173,205,214,221,268,293,294,295	Fourier transform and its derivatives
57	principal component analysis (PCA), linear discriminant analysis (LDA), independent component analysis (ICA), and factor analysis (FA)
31,97,124,169,205,241,245,266,270	Wavelet transform and its derivatives
121	histogram
58	peak detection and pulse width estimation techniques
260,277	Discrete cosine transforms
122	EMD
92	GEI
92	HOG
92	Radon transform and Zernike moments

used to extract time–frequency-domain features [120]. Fourier transform and its derivatives such as the fast Fourier transform (FFT) or short-time Fourier transform (STFT) are mainly used to transform the signal from the time domain to the frequency domain and are widely used methods in extracting features with frequency information. This transform is useful in extracting features with frequency and time–frequency domains. In the paper [57], methods including principal component analysis (PCA), linear discriminant analysis (LDA), independent component analysis (ICA), and factor analysis (FA) are known as traditional feature extraction methods. In these methods, the feature domain is not provided clearly. WT and its continuous (CWT) and discrete (DWT) derivatives are used to extract features in frequency and time–frequency domains. Usually, WT coefficients provide suitable features for classification. In paper [121], the histogram method is used to extract features from American Sign Language signs. Perumal and Sankar [58] extracted various gait features using peak detection and pulsewidth estimation techniques. Discrete cosine transform is a special case of Fourier transform that uses only real numbers, unlike the aforementioned transformation that uses complex numbers. In paper [122], empirical-mode decomposition (EMD) is used for extracting time–frequency features. The paper [92] included feature extraction techniques, namely, gait energy image (GEI), histogram of gradients (HOG), and Zernike moment with radon transform for object identification. Regarding the type of feature, it should be explained that the signal-based statistical feature refers to the statistical properties obtained from the sensor data, for example, the variance or the mean of the acceleration signals. Statistical features available in the papers are mean, standard deviation, variance, minimum, maximum, median, percentiles, mean absolute deviation, mode, skewness, rms, interquartile range, zero crossings, and so on. Regarding structural features, it should be said that these features are also based on polynomials of signals. Transient features are trends (increasing and decreasing), the magnitude of change, and so on [123]. Medical features, especially found in medical applications, are defined by physicians such as freezing index properties and exercise intensity [99], [124]. Body model-based features are features based on motion primitives defined in papers [54], [125], [126]. Gait features are also features that have a spatial, temporal, or spatiotemporal domain, such as stance time, step velocity, and step width, refer to Table VII for more information [127]. We briefly explain some of the items in the table related to the domain of features. The time and frequency domains do not need a special definition, but, for the time–frequency domain, we must state that we will check this domain by mentioning an example; if we calculate the Fourier or wavelet coefficients for a signal in time windows and average these coefficients at each window, we will have a feature with a time–frequency domain. The spectral features themselves are actually frequency domain features, but, in some papers, they are considered separate domains. Spatial features are caused by changes in space due to body motion, while temporal features represent time and its related factors during motion. Therefore, the second case can be considered as a kind of time feature. Spatial features are mainly related to gait features. Spatiotemporal features will have spatial and temporal information on wearable sensor signals. Now, we present the complete statistical and numerical analysis of all three tables related to this section. In total, the feature extraction methods are presented 41 times. Fourier transforms and its derivatives are ranked first with a share of 44% and 18 times of use. The second place goes to WT and its derivatives with nine uses and a share of 22%. Spectral analysis with two uses has a share of 5%. Discrete cosine transform has a similar condition to the mentioned method. All subsequent feature extraction methods have been used in only one paper and have a share of 2%. A total of 110 times, papers have specified the type of feature that they use. Signal-based statistical feature with 90 times of use has the largest share equal to 82%. Gait features have a share of 7% with eight uses. Medical features have a share of 6% with seven uses. Body model-based features have a share of 3% with three uses, and other types with a one-time use have a share of 1%. According to Table VIII, a total number of 100 times, papers have presented the domain of their features. The time domain with 51 times of use has a share of 51%. The frequency domain has a share of 32% with 32 times of use. The spatial domain has a 7% share with seven uses. The time–frequency domain with four times of use has a share of 4%. The other two domains have a share of 3% with three uses. At the end of the statistical analysis, we intend to do a comparative analysis between some of the items in the table related to feature extraction methods. First, we want to make a comparison between the spectral analysis method and the Fourier transform. The main feature domain of both of these methods is frequency, but the spectrum is the appearance and shape of a signal in the frequency domain, and the Fourier transforms generally transform a signal in the time domain into a function in the frequency domain. In general, it can be stated that all feature extraction methods that take the signal to the frequency domain can be a basis for the spectral analysis method. That is, by using these methods, the signal is transferred to the frequency domain, and then, spectrum analysis is done. Therefore, all three proposed feature extraction methods, i.e., Fourier transform, WT, and discrete cosine transform, can be used for this issue. We try to have a comparative analysis between these methods. The comparison of WT and Fourier transform has also been discussed in Section V, but, in this section, we are also trying to make a general comparison of these methods. The Fourier transform decomposes a signal into simple sines and cosines. Unlike the Fourier transform that is limited to a scaled single sinusoidal function, the WT generates a two-parameter family of wavelet functions by scaling and shifting the function [31]. It can be stated that the WT displays the signal in both the time and frequency domains, while the Fourier transforms displays the signal only in the frequency domain. Discrete cosine transforms express a signal in terms of the sum of cosine functions. The discrete cosine transform is very similar to the discrete Fourier transform, and the obvious difference between the discrete cosine transform and discrete Fourier transform is that the former uses only cosine functions, while the latter uses both cosine and sine. Therefore, the result of the discrete cosine transformation has only real values. A discrete cosine transform is equivalent to a discrete Fourier transform of twice the length. EMD, which is another feature extraction method proposed in this article, is a well-known method for data analysis that breaks a signal into intrinsic mode functions (IMFs) that describe the behavior of the signal [122]. They consist of a single frequency or a narrowband of frequencies. This method breaks the time signal into a series of basic functions just like the Fourier transforms and the WTs, but, unlike the two announced methods, this method extracts the basic functions from the data itself. PCA is an unsupervised linear transformation that can be used for feature extraction and feature reduction. We are trying to provide a general definition of how PCA works, which can be used for both feature extraction and feature reduction. This algorithm obtains the relationships between data using the covariance matrix. Then, using special relations from the covariance matrix, eigenvalues and eigenvectors are obtained. Eigenvectors are used to transform the data into principal components, and finally, the important principal components are selected by examining the eigenvalues. Unlike PCA, which is an unsupervised feature extraction method, LDA is a supervised feature extraction method that is also used as a machine learning classification algorithm. Feature extraction or feature reduction is performed by this algorithm in such a way that the algorithm calculates intraclass and interclass variances of data or features and tries to extract or reduce features by minimizing intraclass variance and maximizing interclass variance. ICA is an unsupervised feature extraction method and the machine learning algorithm that decomposes signals into independent subcomponents of non-Gaussian nature. This algorithm can be used for feature reduction, too. FA is also an unsupervised machine learning algorithm that is used for feature extraction and feature reduction; this algorithm removes the correlation between a huge set of data or variables and extracts the basic factors that represent the dependents. The factors that are created show the variance caused by similarity and correlation. Incremental FA (IFA) is the FA that calculates covariance with an incremental approach; incremental approaches are especially used in feature reduction and feature extraction methods to reduce time complexity and save storage space. It is not bad to have a comparison between the performances of the above algorithms. All these algorithms look for linear combinations of variables that best describe the data. PCA is defined as an orthogonal linear transformation that aims to create new components that capture the maximum input variance. LDA creates new components that separate classes. The goal of ICA is to recover the original features that are mixed in a linear combination in the input dataset. FA tries to describe a dataset via a linear combination of variables called factors. It was tried to check and compare the performance of the most widely used and famous feature extraction algorithms. For more familiarity with other feature extraction methods in the table, refer to the relevant papers.

F. Feature Selection

Feature selection has many applications in various fields, such as machine learning, classification, pattern recognition, data mining, and clustering, for reducing the size of the feature space [128], [129]. Feature selection algorithms and their type will be specified in this section. There are three different methods for feature selection: filter methods, wrapper methods, and embedded methods [129], [130]. Some papers have also discussed hybrid methods for feature selection [129], [131], [132]. Jović et al. [129] have described all these methods. Filter methods select features based on a performance measure regardless of the employed data modeling algorithm. Only after the best features are found, the modeling algorithms can use them [129]. Filter methods are mostly based on similarities and statistical measurements. In this article, the wrapper method is also defined as follows: wrappers consider feature subsets by the quality of the performance on a modeling algorithm. Embedded methods perform feature selection during the modeling algorithm's execution. These methods are, thus, embedded in the algorithm either as its normal or extended functionality [129]. Hybrid methods were proposed to combine the best properties of filters and wrappers. First, a filter method is used in order to reduce the feature space dimension space. Then, a wrapper is employed to find the best candidate subset [129], [132]. Before dealing with the statistical analysis, we are going to introduce the items in the feature selection table. The minimal-Redundancy Maximal-Relevance (mRMR) method measures the relevance and redundancy of the feature candidates with the target class based on mutual information and selects a promising feature subset that has maximal relevance and minimal redundancy [27]. Generally, it can be said that mRMR, joint mutual information (JMI), conditional mutual information maximum (CMIM), and double-input symmetrical relevance (DISR) methods are based on "relevance" and "redundancy," and they can be considered mutual information-based feature selection methods. The JMI method just calculates the JMI between a target class and each of the features, and selects the feature with the highest performance. DISR has a similar structure to JMI. These two methods differ only in the objective function. CMIM selects features by maximizing mutual information with a target class, given the preselected features. The information gain-based feature selection method calculates the information gain (entropy) for each feature. Features that contribute more information will be selected, and those with lesser information will be removed. The correlation-based feature selection method selects the most useful features. This feature selection method is fast and simple [59]. According to our studied papers, this method selects features that are highly correlated in a certain class but not correlated with each other. Relief is a feature selection algorithm or method that calculates a score for each feature, then uses this score for ranking, and selects high-scoring features to continue. Many updates have been made to fix the limitations of the ReliefF algorithm [97]. These limitations include inadequate performance in the presence of missing data, unreliable performance in the presence of noise, and so on. One of the most famous of these updates is the ReliefF algorithm, which removes some of the limitations of the original algorithm, such as poor performance in the presence of missing data, and can be used in multiclass classification problems, unlike the original algorithm, which was designed for binary classification problems. The t-test, f-test, paired t-test, Wilcoxon sum rank test, and analysis of variance (ANOVA) methods are statistical methods for feature selection. All of them select the best features by thresholding the p-value, that is, they compare this value for each feature and select the best feature. It is better to do a comparative analysis of how these methods work. The t-test is a statistical test used to compare the means of two groups. The t-test can be used as a statistical feature selection method that assigns a p-value to the features based on their discriminability and then selects the appropriate features based on the value of the p-values. Paired t-test is a special type of t-test that stands in front of an unpaired t-test; this test shows the mean difference between two dependent groups, and the second one shows the mean difference between two independent groups. The f-test compares the variance of the two groups, while the t-test compares the mean of the two groups. The Wilcoxon sum rank test is a nonparametric statistical analysis method that selects the most relevant features [31]. The method can be considered as the nonparametric version of the t-test. This method calculates the p-value and removes features that have a p-value less than a certain threshold [31]. ANOVA is a statistical test that can be used to analyze the difference between the means of two or more than two datasets. It can be said that ANOVA is a generalization of the t-test. To select the features in this method, a variable called f-value is calculated for each feature from the variance of the data, and then, it is converted into a p-value, which determines the importance of a feature, and features are selected again by applying a threshold on this value. The p-value threshold is usually set to 0.05 [31], [58]. All the algorithms and methods mentioned above were of the filter type; now, we introduce the wrapper-type methods. Backward elimination (BE), also named backward selection or sequential backward selection (SBS), is a wrapper-type feature selection method that starts with all the features, then eliminates the weaker features by scoring, and selects the new feature set. This method is a type of sequential feature selection method. The sequential feature selection method has two types (forward feature selection and backward feature selection or elimination) and greedily selects features. In the paper [101], the type of this feature selection method is specified as the sequential forward selection method, so we also explain the forward selection method. Forward selection works exactly the opposite of the BE method, that is, there are no features in the model, and then one by one, features are added to the model. In this method, the features that improve the performance of the model in the best way are added one by one until the addition of features does not improve the performance of the model. Inoue et al. [62] reduced these 27 feature variables to 13 by applying stepwise feature selection using logistic regression. Logistic regression is a classification algorithm that can perform feature selection by using regulatory rules and determining penalty variables. In the paper [91], the important features for gait activity recognition are selected using the biogeography-based optimization (BBO) technique. BBO is an evolutionary method. This algorithm is derived from the theory of biogeography and is inspired by the analysis of the geographical distribution of species. The greedy heuristic feature selection method looks at the feature selection problem as an optimization problem and finds local optimal solutions for the problem. In the paper [118], the best features of the classification process are attained by means of an extremely randomized tree (ERT) technique. The ERT is a tree-based ensemble learning technique that combines the results of multiple decorrelated decision trees collected. The entropy-based information gain is essentially used as the decision criteria for the significant features [118]. In the paper [122], to search for the near-optimal subset of features, which maximizes classifier performance, a floating forward–backward feature selection algorithm was employed. The performance of each feature subset was assessed using cross-fold validation. Sequentially, by selecting the best feature from an unselected pool of features, the algorithm adds the feature to the existing selected set of features, provided that the addition of this feature increases the classification accuracy. After the selection of each feature, the removal of a feature from the selected set of features was also considered. The selection procedure stopped

TABLE VII
DOMINANT FEATURE TYPE

Reference number	feature type
26,27,35,40,46,54,55,56,59,60,62,70,74,94,95,96,97,98,99,100,101,103,104,105,106,110,114,118,121,122,123,124,125,126,136,137,139,140,141,142,145,151,156,158,160,161,162,167,169,170,171,173,189,190,192,193,195,197,199,200,202,203,205,206,207,211,213,217,221,222,225,234,241,242,243,252,256,257,263,264,265,266,268,271,279,282,285,287,292,294	Signal-based statistical feature
123	Signal-based structural feature
123	Signal-based transient feature
56,99,124,169,236,241,287	Medical
54,125,126	Body model-based features
58,124,169,170,241,257,260,274	Gait

TABLE VIII
FEATURE DOMAIN

Reference number	Feature domain
26,27,35,46,54,55,56,58,70,74,96,99,103,104,109,110,118,120,122,124,126,138,139,141,142,145,151,169,170,190,193,199,202,203,205,207,211,214,217,221,222,234,241,242,243,257,260,287,293,294,295	Time
26,27,31,35,54,56,70,74,96,99,103,110,122,124,126,145,151,169,199,202,203,214,221,234,236,241,260,268,287,293,294,295	Frequency
122,124,169,241	Time-frequency
122,138,268	Spectral
58,120,124,169,170,241,257	Spatial
58,120,170	Spatiotemporal

TABLE IX
FEATURE SELECTION (METHODS AND ALGORITHMS)

Reference number	Type of method	Name of the method used or introduced
26	Filter	mRMR
27	Filter	mRMR
31	Filter	Wilcoxon Sum Rank
56	Filter	mRMR
56	wrapper	BE
58	Filter	Anova test
59	Filter	Correlation-based feature selection method
62	Wrapper	stepwise feature selection using logistic regression
91	wrapper	BBO evolutionary algorithm
95	Wrapper	greedy heuristic
97	Filter	ReliefF algorithm, improved version of original relief algorithm
101	Wrapper	Sequential Forward Selection (SFS)
103	Wrapper	Did not mention
110	Filter	information gain-based feature selection
114	Filter	
118	Wrapper	
122	Wrapper	floating forward– backward feature-selection search algorithm
139	Filter	The Correlation-based Feature Selection
140	Filter	mRMR, F-TEST
145	Wrapper	Sequential Backward Selection (SBS)
158	Filter	information gain-based feature selection
171	Wrapper	Greedy heuristic
189	Filter	mRMR, JMI, CMIM, and DISR
207	Wrapper	Did not mention
213	Filter	mRMR
252	Wrapper	RFECV
267	Wrapper	Brute Force Search
287	Filter	Paired t-test
293	Wrapper	Boruta

when no further classification performance improvement was observed through the addition or removal of a feature from the unselected pool [122]. Recursive feature elimination (RFE) is a feature selection method that starts feature selection with all features and removes weak features until a certain number of important features remain. Ranking of features is done using model coefficients or feature importance attributes. RFE with cross-validation, which is abbreviated as RFECV, as a type of RFE, performs the same feature ranking process using the cross-validation score of the model. The brute-force feature selection algorithm or method, which is also known as exhaustive search feature selection, examines all candidate feature subsets and, finally, selects the best subset in terms of performance criteria. If the number of features is large, this method will have a very high computation time. The last wrapper-type feature selection algorithm that is introduced is the Boruta algorithm, which is based on the random forest classifier and finds the importance of a feature using shadow features. Shadow features are random copies of all features. This method compares the importance of the features with their shadow features using a criterion and selects the more important features. In total, 34 feature selection methods are described. 20 methods are filter methods. The mrMR algorithm is the most widely used algorithm with seven times of use and having a share of 35%. Information gain-based feature selection, correlation-based feature selection, and relief-based algorithms have been used twice and have a share of 10%. Other feature selection algorithms in Table IX are used only once and have a share of 5%. In total, 14 methods are wrapper methods. Two papers have not presented a specific name for the algorithm used for feature selection; the greedy heuristic approach along with the backward selection method has been used two times and has a 14% share; and all the other algorithms used for feature selection with this method are used only once and have a share of 7%.

G. Feature Reduction

Feature reduction is one of the most famous machine learning glossaries and terms. Feature reduction is also known as dimension reduction, and according to the deepAI, machine learning dictionary is the process of reducing the number of features without losing important information. To differentiate between feature selection and feature reduction, we need to know that in feature selection, and we simply choose from the features and do not change them, while, in dimension reduction, some kinds of features with smaller dimensions are produced. In this section, feature reduction or dimensionality reduction methods are examined. Since we have explained the main feature reduction algorithms (such as PCA, LDA, and IFA) in the feature extraction section and considering that the feature reduction algorithms can be used in the feature extraction step, the performance of the remaining algorithms will be fully investigated in Section V. Statistical analysis of the table in this section is not necessary because the PCA

TABLE X
FEATURE REDUCTION ALGORITHMS

Reference number	The algorithm proposed in the paper
104,156,159,192,245,253,257,273,282,292,296,302, 304	PCA
156	CPCA
156,292	LDA
156,292	KDA
156	NWFE
156	PCA+LDA
156	NWFE+PCA
156	NWFE+LDA
60	1D-LBP
233,234	Discriminant analysis
250	locality preserving projections
159	CCA
267	MRMI-SIG
292	KPCA
302	IFA

algorithm is the most widely used. This algorithm has been used in 13 papers. Since 30 times algorithms are introduced for dimensionality reduction, the algorithm has a share of 43%. The discriminant analysis (DA) algorithm consists of rows with names LDA, kernel DA (KDA), and DA in Table X. This algorithm is used in six papers and has a share of about 20%. All the other algorithms have been used once and have a share of 3%.

H. Classification

In the classification section, we categorize our datasets into two or more classes, patterns, groups, and templates. We assign a special label to each dataset. Classification as we have specified in this article is the last step of the project. It has also been mentioned in other papers as the last step [51], [118], [133]. A classification algorithm, in general, is a function that weighs the input features so that the output separates one class into positive values and the other into negative values [134]. SVM, LDA, ANNs, KNN, logistic regression, decision trees, and naïve Bayes can be easily used to classify the two classes. However, most of the time, it might be necessary to classify more than two classes; for example, we need to distinguish between walking, running, sitting, and standing activities as movement classes, which is needed at this time to make changes to the structure of binary classifiers, such as SVM, while multiclass classifiers, such as gradient boosting, do not require structural modification. The classification algorithms used in the papers along with their types are specified in Table XI. An overview of the table shows that SVM and neural networks are the most widely used classifiers, but, since neural networks are so diverse, we have introduced each member of this family separately in the table. Note that SVM is a neural network only when we face a binary classification problem, and since binary classification problems are less common in movement classification, we consider SVM as a separate classifier. We have shown the most commonly used machine learning algorithms based on the number of repetitions in the papers in the bar chart. The classification algorithms in the papers are divided into two categories: machine learning algorithms and classical classification algorithms. We chose the classical term because it is hard to choose a name for the algorithms that go against machine learning, and that choice is because these algorithms existed somehow before scientists became familiar with the concepts of machine learning. We first turn to classical classification algorithms because, as you can see in the table, they are less commonly used and, of course, weaker than their competitors, the machine learning algorithms. The most widely used of these algorithms are threshold-based algorithms and correlation-based algorithms. Threshold-based algorithms are typically used in conjunction with flowcharts. Binary classification can be completed by identifying the threshold (usually experimentally) and applying it to the discriminative feature. If you have more than two classes, you need different thresholds for different classification classes that must be properly embedded in the flowchart. Correlation-based classification investigates the correlation between features or raw data in order to create a classification model. To learn more about these algorithms, see the papers in the rows of the table called discrete WT, threshold-based algorithms, linear classifiers, and so on. Other algorithms are included in the table just for the sake of familiarity. In this article, we are not interested in dealing with classical classification algorithms, and these algorithms will not be discussed further. The rows, which are light blue, contain machine learning algorithms that are very versatile. These algorithms have different categories. We have tried to introduce a general category in this article that includes all the different categories of these algorithms. The category that we are considering contains supervised, unsupervised, combined, rule-based, probabilistic, and reinforcement algorithms. As you may know, supervised algorithms require labeled data to be first trained and then tested, and unsupervised methods do not need labels to identify the pattern. Therefore, these two can be named the most familiar types to determine and identify classification algorithms. Some papers, as you can see, combine classification algorithms and invent new ones. FFSVC, SRC-SVD, and FMM-cart fall into this category. This can lead to the production of semisupervised methods for cases where we have a few labeled or trained data. Probabilistic algorithms are used when we have uncertainty in our model or data; perhaps, the main uncertainty that we all know is noise. According to the table data, it is easy to determine that the most famous rule-based classification algorithm is fuzzy logic. This logic, which is a kind of extension of Aristotelian logic, is the best way to deal with human logic. Fuzzy logic is the best option when we are faced with the ambiguities of human logic in classification, and we want to translate these ambiguities in the best way for the machine. Of course, as you can see in the table, fuzzy logic is not the only rule-based algorithm, and there are other types, including rough sets. Of course, other categories can also be found in various papers, and unfortunately, in our studied papers, reinforcement learning algorithms, such as genetics, have not been used. However, it is possible to use such machine learning algorithms for the classification. As you can see, supervised

TABLE XI
CLASSIFICATION ALGORITHMS

Reference number	type	Classification algorithms
46,48,59,62,91,94,99,100,118,123,136,138,139,151,158,159,160,161,162,199,203,205,206,221,254,257,258	Supervised	Decision tree
26,27,36,56,58,59,60,62,64,70,91,92,94,96,97,99,101,103,104,105,110,114,118,135,136,140,142,145,158,159,160,161,162,167,189,191,192,194,199,200,201,202,203,204,208,213,237,241,242,243,249,252,253,254,256,257,258,260,264,273,282,292,293,294,296,302,304	Supervised	SVM
59,67,89,198,201,308	Supervised	Deep neural network
59,62,91,94,95,96,100,136,157,159,171,199,200,201,207,254,279	Supervised	Random forest
35,59,96,214,292	Supervised	ADABOOST
100,118,123,159,170,254	Supervised	Logistic regression
26,60,89,302,304	Supervised	ELM
26,46,47,60,91,94,95,97,98,99,110,116,118,139,157,159,169,171,192,197,202,203,207,208,217,221,237,252,257,258,265,292,293,302,304	Supervised	KNN
56,58,89,92,95,99,121,141,142,171,190,253,254,264,267,279	Supervised	Neural network or ANN
35,56,70,94,96,99,100,110,116,138,139,151,158,190,192,199,202,207,208,211,217,225,227,237,245,246,257,260,268,279,293	Probabilistic	Bayes derivatives (naïve bayes, bayes net, …)
35,36,42,46,70,72,109,136,138,143,174,191,195,202,205,206,212,214,215,216,229,235,261,262	Probabilistic	HMM
106, 120	Supervised	Fully connected net
106	Supervised	Deep belief network
31,99,105,110,124,160,161,162,193,254,279,293,302	Supervised	Multi-layer perceptron
137	Supervised	SVDD
137	Supervised	CRC
196,278,301	Classical	Rule based algorithm
57,63,88,145,201,208,209,272,291,308	Supervised	LSTM
99	Supervised	Linear regression
99	Supervised	Bagged trees
40,72,73	Probabilistic	CRF
136,217,218,253	Unsupervised	K means
36,122,206,250,295	Probabilistic	GMM
55,64,67,88,120,202,208,209,237,272,291,303,306,308,309	Supervised	CNN
202	Combined algorithms	SRC_SVD
202	Combined algorithms	FMM-CART
202	Supervised	Knowledge distilling
202	Probabilistic	mfap
104	Combined algorithms	FFSVC
151,158,172	Supervised	Nearest neighbour
210	Supervised	RNN
144,172,191,211	Supervised	Dynamic time warping (DTW)
269, 270	Classical	discrete wavelet transforms
70	Unsupervised	Topic model
151	Supervised	Decision table
88	Combined	LSTM-CNN
88	Supervised	GRU
88	Combined	GRU-CNN
91	Supervised	Gradient boosting
91	Supervised	Extreme randomized tree
91	Combined	PSO-SVM
88,303,305,306,307	Combined	CNN-GRU
114,115,174	Supervised	String matching
229	Probabilistic	Relational Markov network
48,58,100,109,118,203,222,233,234,257,263,284,287	Supervised	Discriminant analysis
117,238,289	Classical	Threshold-based algorithm
239	Classical	Linear classifiers
239,240	Supervised	Dynamic neural network
141,142,291	Supervised	BPNN
244	Rule based	Rough sets
244	Supervised	Adaptive logic network (ALN)
245	Unsupervised	Non-parametric adaptive method based on data clustering
220,247,251,259,266,276,280,288	Rule based	fuzzy

TABLE XI
(Continued.) CLASSIFICATION ALGORITHMS

95,171	Supervised	m. logistic
95,279	Supervised	Rotation forest
255	Supervised	classification scheme based on TUCKER2 tensor decomposition
263	Supervised	Classification tree
122,248,281	Supervised	Distance-based methods
105	Supervised	Multinomial Logistic Discrimination (MLD)
274	Classical	Comparison of cases associated with the disease in healthy and unhealthy individuals
144,173,281,285	Classical	correlation
285	Classical	similarity score based on distance
176,286	Classical	Cycle matching method or cycle length similarity
56	Supervised	Feed forward neural network
284	Supervised	inductive learning algorithm
285,286	Classical	histogram
285	Classical	High order moments
292	Classical	FAST FOURIER TRANSFORM
92,293	Supervised	XGB
110	Supervised	Tree method
110	Supervised	K star
303	Combined	ICGNET
303	Combined	ISPLINCEPTION
87,88,303,306,307,308,309	Combined	CNN-LSTM
307	Combined	CNN-biLSTM

algorithms are the most widely used algorithms in this field. We use bar charts to identify the most commonly used types of machine learning algorithms. For a better understanding of bar charts, a full numerical analysis will also be provided. However, before that, we are trying to introduce some commonly used machine learning algorithms, some algorithms have been implicitly introduced in Section II-C, and we will not introduce them in this section again. Decision trees are one of the most widely used nonparametric supervised machine learning algorithms. The nonparametric means data analysis is done without different assumptions or specific parameters. This algorithm has a branched tree-like structure. The decision tree consists of different nodes. The main node is the root node, which is considered the starting point of the algorithm, and the leaf nodes are the endpoints of the tree branch and can represent the endpoint of the set of decisions; the leaf with the most records can be introduced as a class. A random forest is a metaclassifier consisting of several decision tree classifiers. This classifier usually has a better classification accuracy than a decision tree and prevents overfitting. AdaBoost is a metaclassifier that can combine several weak classifiers, such as decision trees, and improve performance. AdaBoost is an abbreviation for adaptive boosting. Logistic regression is an example of a binary supervised machine learning algorithm used for classification. It can be used to calculate or predict the probability of an event with two states (0 and 1). In general, this algorithm is used for binary classification problems, but, by changing the structure and creating the multinomial logistic regression algorithm, it can also be used in multiclass problems [123]. The HMM is a statistical Markov model that models the system as a Markov process with hidden states. The HMM is a generative probabilistic classifier. HMMs have been successfully used in modeling different types of time-series data, such as speech recognition and gesture tracking [35]. Neural networks, also known as ANNs, form a large class of machine learning classifiers and have different types that are specified separately in the table of classification algorithms. We try to introduce famous types of algorithms in the relevant table. Neural networks simulate the way the human brain classifies related concepts. A neural network consists of several neurons in a layered structure. The neural network forms a mathematical function that takes the input data, transfers it to the output, learns the pattern, and performs the classification. The feedforward neural network is a neural network that does not form a cycle or loop in the connections between the constituent units. This neural network is the first and simplest type of neural network. In this neural network, information is transferred in one direction from input to output. The multilayer perceptron neural network is also a special type of these neural networks, which, in its simplest form, consists of three layers: the input layer, the hidden layer, and the output layer. Information is transferred from the input to the output, and the output layer is responsible for the classification process. The backpropagation (BP) neural network (BPNN) is the feedforward neural network trained by the backpropagation method, which is a mathematical method to increase classification accuracy. A fully connected net (FC net) is one of the most commonly used neural networks. In FC net, every neuron in layer I have a connection with every neuron in layer $I + 1$, while the nonfully connected networks only have partial connections [106]. Deep neural networks refer broadly to neural networks that exploit many layers of nonlinear information processing for feature extraction and classification, organized hierarchically, with each layer processing the outputs of the previous layer [67]. A deep belief net (DBN) is a deep neural network model that is made by stacking several restricted Boltzmann machine (RBM) layers. The output of the RBM at the previous layer is set to be the input of the RBM at the current layer, and there will be a soft-max layer at the top RBM layer. The purpose of the soft-max layer is to transform the model scores for each class into the normalized probability distribution [106].

A convolutional neural network (CNN) is a feedforward deep neural network that uses convolution operation in some layers. CNN typically consists of a combination of three different layers: a convolutional layer, a pooling layer, and a fully connected layer. In the convolutional layer, the convolution operation is applied to learn local features automatically. A pooling layer is added to reduce the training time and avoid overfitting by reducing the feature representation. The output of the pooling layer provides high-level distortion-invariant features. Both convolutional and pooling layers could be applied multiple times depending on the CNN structure [55]. The automatically extracted features by these layers are used to train a fully connected neural network layer. The output of this fully connected layer is used to compute the probability distribution over the learned activity classes inside a soft-max layer [55]. One of the main benefits of using CNN is that it does not require any prior knowledge about the data [55]. Gated recurrent units (GRUs) follow a very similar approach to LSTM units. GRU has an updated gate and a reset gate that are responsible for the flow of information vectors. These gates combinedly decide what part of the tensor needs to be remembered in the next step and which may be updated [88]. Dynamic neural networks are actually opposite to static neural networks and are created with structural changes in routine neural networks, for example, creating feedback from output to input in the structure of a static neural network can lead to the creation of a dynamic neural network. So far, we have tried to introduce numerous neural network algorithms that are used for classification. Now, we introduce some other famous or widely used algorithms in this field. To get acquainted with other less-used algorithms in the table, refer to the papers provided for them. As we have previously announced, DA divides the data into two or more classes by increasing the interclass variance and decreasing the intraclass variance. There are two types of DA classifiers, namely, LDA and quadratic DA (QDA) classifiers. In LDA classification, the decision boundary is linear, while the decision boundary in QDA is nonlinear. The second one is more flexible than the first one. Gaussian mixture models (GMMs) are probabilistic machine learning classification models that assume that a dataset can be considered as a mixture of several Gaussian probability distributions and perform classification based on these criteria. CRF is a class of statistical modeling methods used for structured learning and prediction. CRF can support more complex and useful feature sets by modeling the posterior probabilities [40]. We already explained the KNN classification algorithm; if k in that algorithm is considered equal to one, the nearest neighbor algorithm is born. In this algorithm, the output is simply labeled to the nearest neighbor. Topic models stem from the text processing community. They regard a document—e.g., a scientific paper—as a collection of words, discarding all positional information. This is called a "bag-of-words" representation. As a single word captures a substantial amount of information on their own, this simplification has been shown to produce good results in applications such as text classification [70]. Perhaps, the presence of this classification algorithm among movement classification algorithms is surprising. However, Huynh et al. [70] have introduced a novel approach for modeling and discovering daily routines from on-body sensor data based on this machine learning algorithm. Inspired by machine learning methods from the text processing community, they have converted a stream of sensor data into a series of documents consisting of sets of discrete activity labels. These sets are then mined for common topics, i.e., activity patterns, using latent Dirichlet allocation. In an evaluation using seven days of real-world activity data, they have shown that the discovered activity patterns correspond to the high-level behavior of the user and are highly correlated with daily routines. String search (or string matching) algorithms are for finding places where one or more strings are found in a larger string or text. They are used to find the strings of a text or string. The use of this algorithm in the field of movement classification is also a bit surprising, so, to disambiguate, we present some examples of how to use it. In the paper [114], eye movements are recorded using an EOG system. The string matching algorithm is used for explicitly modeling the characteristic horizontal saccades during reading. In the paper [115], string matching is used to spot occurrences of gestures in a continuous stream of data. Now, we will check the statistics of the algorithms in the table. The total number of papers presented in Table XI is 402, including both machine learning and classical algorithms. The SVM algorithm with 67 uses has a share of 17%. The KNN algorithm with 35 uses has a share of 9%. Bayes derivatives include all the classification algorithms that use the Bayes probability law for classification, such as naïve Bayes and Bayes net. These algorithms also have a share of about 8% with 31 uses. The decision tree with 27 uses has a share of 7%. HMM, with 24 times of use, has a share of 6%. Random forest with 17 uses and a 4% share is in pursuit of HMM. Neural networks or ANNs have a share of 4% with 16 uses. CNN also has a share approximately equal to the previous algorithm with 15 uses. The DA algorithm is in the next rank with 13 uses and a share of about 3%. This algorithm has different types, such as linear and quadratic, which are used in papers for classification. The multilayer perceptron is also used in 13 papers and has a share of 3%. LSTM is used ten times in papers and has a 2% share. Fuzzy algorithms are used eight times in papers and have a share of 2%. Other machine learning algorithms have a percentage share of about 2% or less and are used less than eight times, so their presence in the related bar chart has been omitted. As can be seen, supervised algorithms, such as SVM and KNN, are at the top of use, which is not far-fetched and is predictable, because these algorithms have proven their usefulness over the years. In total, there are 402 proposed classification algorithms, 21 are classical algorithms, and their presentation in Fig. 7(b) is omitted; 280 algorithms are supervised and have a share of about 70%. The number of probabilistic algorithms is 65, and their share is 16%. Combined algorithms have been used in 21 papers and have a share of about 5%. Rule-based machine learning algorithms are used nine times and have a share of 2%. Six algorithms are unsupervised and have a share of 1%. With these numbers in mind, the reader can have a better understanding of bar charts and can choose freely from the most used classification algorithms and classification

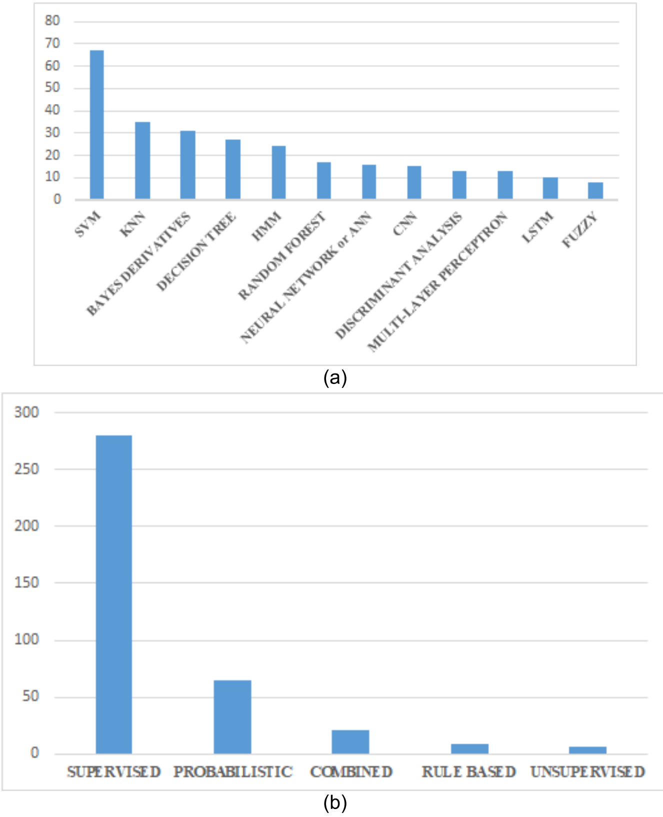


Fig. 7. (a) Most commonly used machine learning classification algorithms based on the number of repetitions in the papers. Vertical axis: the number of times the algorithms are used. Horizontal axis: the names of the algorithms. (b) Most commonly used machine learning classification algorithm types based on the number of repetitions in the papers. Vertical axis: the number of times the classification algorithm types are used. Horizontal axis: the names of the classification algorithm types.

algorithm types. Unfortunately, reinforcement algorithms are not used for classification in this field.

I. Software or Language Used and Their Field of Application

This article does not cover the hardware part as it can be extensive, vary, and also depends on the taste of the author or researcher, and it does not follow a specific pattern. In this section, the software or language used in the papers and the application area of this software or language will be specified. The sensors section is perhaps the most common piece of hardware. However, this part is very important, and it can be a great help to researchers and authors to complete the project by introducing the software or language used and their application area. Of course, not all researchers mention the software or the language that they use in the paper, and this also depends on their taste in writing the paper. Sometimes, all data processing is done with just one software or language [47], [135], [136], and in some papers, several software, language packages, and even device software are used in combination to complete the task [26], [103], [110], [115], [137], [138], [139], [140], [141], [142], [143], [144]. The papers mainly describe the dominant area of using the software or language. The software or the language is present in all the mentioned steps of the project from data collection [137], [145], preprocessing [32], [97], [110], [135], [143], [144], signal segmentation [115], feature extraction [110], feature selection [110], [139], and feature reduction [104] to classification. Classification software or classification language has been reviewed in many papers as you can see in the table. Of course, there are miscellaneous applications, such as creating a graphical user interface or creating a musical environment [141], [142], [146]. The purpose of this section is to get acquainted with the most widely used toolboxes, software packages, and languages and their area of application. As can be deduced from Table XII, the most widely used software in this field is MATLAB, the second place goes to Weka, and classification is the most widely used area by these two popular software programs. However, this information is very general and we intend to fully specify the software and programming languages used in this field and also specify their use in each step of movement classification. A total of 61 papers have introduced their software or programming language, and movement classification steps are presented 85 times. MATLAB software is used in 36 papers. Weka software is used in 14 papers. Scikit learn python library is used five times. C and C++ languages have been used five times each. LabVIEW is a graphical programming language that is used three times same as the TensorFlow-Keras python library. SPSS is a statistical software that has been used twice. Objective C programming language, rapid miner data science software, and MAX as a visual programming language for music and multimedia have been used only once. Now, we introduce the steps of movement classification that are implemented by these software or programming languages. In the application area column of the corresponding table, only steps related to movement classification are counted in numerical analysis. Evaluation is also not presented in statistical analysis because it can be considered a part of the classification step anyway. The steps of movement classification are presented in the relevant table in total 85 times. The classification step has been implemented 51 times by these software or programming languages and has a share of about 60%. Preprocessing has been done 12 times by these software or programming languages, so this step has a 14% share. Feature extraction has been done ten times and its share is about 12%. The segmentation step has been implemented seven times and has a share of 8%. Feature selection and feature reduction were implemented two times each and have a share of 2%. Data fusion is presented only once in the table without mentioning the level and has a share of 1%. The software and languages used for preprocessing are MATLAB, which was used ten times, C, which was used two times, and C++ and LabVIEW, which were used once each. Weka is used two times for feature selection. MATLAB was used nine times for feature extraction, and C and C++ were used once each in the same paper for feature extraction. Feature reduction is done two times with MATLAB software. It can be inferred from the table that the software and languages used in the classification step along with their usage rate are MATLAB software 27 times, Weka 14 times, scikit-learn python library four times, TensorFlow-Keras python library three times, and

C and C++ two times each in the same paper and rapid miner once. MATLAB software has been used once for data fusion.

J. Evaluation

As it was mentioned earlier, the classifier allows us to identify unknown data, tag it, and specify its motion class. This occurs when the quality of the trained classifier is evaluated using the evaluation step [147]. In some papers, evaluation is divided into two parts, training and testing, and in some other papers, this step is divided into three parts: training, validation, and testing. Of course, it should be noted that unsupervised classification does not require a specific training step and directly infers activities from sensor data. Ground truth, a concept related to the training phase that leads to the production of labeled data, is not discussed in all papers, and we ignore it and only get acquainted with this concept. Generally, papers that deal with hyperparameter tuning or optimization parameter tuning need validation [106], [124], and papers that do not need this part or use default hyperparameters will only run the training and testing part. Because validation is not an essential part, its methods will not be covered much, but validation methods are like evaluation methods. Now, we will define each part of the evaluation. The definition of training datasets and test datasets is very comprehensive. We use the training dataset to fit the model and, in a general sense, to train and create the desired model and for understanding the relationship between the dataset and its corresponding class. This dataset contains a large part of the entire existing dataset and usually determines the weights of the nodes. Test datasets are also unknown and unlabeled datasets that determine how well our model performs the labeling operation and examines the quality of the created model. To get acquainted with validation, we need to know what hyperparameter optimization is and why the hyperparameter needs to be tuned and then define validation. The paper [148] has stated that hyperparameter optimization is a process to find suitable hyperparameters for predictive models. It typically incurs highly demanding computational costs due to the need for the time-consuming model training process to determine the effectiveness of each set of candidate hyperparameter values. There is no guarantee that hyperparameter optimization leads to improved performance. However, this can be achieved by thinking of measures. Hyperparameters from the classifier in the toolbox of various softwares have a default value that the model performance can be maximized by tuning the hyperparameter. Hyperparameter tuning is very common in SVM and neural networks, but hyperparameter tuning of classifiers such as decision tree, random forest, KNN, naïve Bayes, linear discriminate analysis, and AdaBoost has also been discussed in the papers [148], [149]. As the last recommendation of this section, we want to announce the data split rate for all three sections of training, testing, and validation. They usually allocate 70%–75% of the data for training and 20%–25% for testing. If the hyperparameters need to be tuned, 70% of the data are generally intended for training, 20% for validation, and 10% for testing [150]. Evaluation methods, metrics, methods of obtaining metrics, and methods of announcing the results are discussed in Tables XIII–XVI, respectively.

In Table XIII, evaluation methods are presented. First, we summarize the proposed methods as specific methods and then define each specific method. Evaluation methods under the titles of tenfold cross-validation, k-fold cross-validation, twofold cross-validation, fivefold cross-validation (5-foldCV), threefold cross-validation, sevenfold cross-validation, fourfold cross-validation, sixfold cross-validation, 20-fold cross-validation, and random split k-fold cross-validation are methods with the same structure and are considered as k-fold cross-validation. Evaluation methods with the titles leave-one-subject-out cross-validation, leave-one-out cross-validation, leave-one-participant-out cross-validation, leave-one-out test cross-validation, leave-one-day-out cross-validation, leave-one-person-out cross-validation, leave-one-user-out cross-validation, leave-one-instance-out cross-validation, and leave-one-out cross-comparison also have a similar structure and are considered leave-one-out cross-validation. Biased cross-validation is considered a special method. Titles such as subject-based cross-validation, cross-validation, and individual-based cross-validation are also considered cross-validation methods. Hold-out cross-validation is considered a special method. User-specific training also has the same condition as the previous method. Titles such as repeated leave-one-out random subsampling cross-validation and repeated random subsampling cross-validation are considered as repeated random subsampling cross-validation methods. The titles subjectwise leave-one-out, grouped stratified k-fold cross-validation, and stratified k-fold cross-validation will also be defined separately. First, we define the concept of cross-validation. Cross-validation is a method that determines how generalizable the classification results will be to an independent and unknown dataset. The most widely used method of this concept is k-fold cross-validation, which is used 52 times in the total of 106 evaluation methods proposed in the table, has a share of 49%, and is the most used method. In this evaluation method, the dataset is divided into k groups of equal size. A subset is used to test the classification model, and $k-1$ subsets are used to train the classification model; this process is repeated k times. The next most used method is leave-one-out cross-validation, which is used a total of 37 times in the studied papers and has a share of 35%. In this method, the dataset is divided into several groups; all groups except one are used for training and only one is used for testing; and this is done so much that all groups are selected as the test group once. The third place goes to cross-validation, which has a share of 7% with seven repetitions. Repeated random subsampling cross-validation with three repetitions and a 3% share is ranked fourth. This method is also known as the Monte Carlo method. This method works in such a way that the dataset is randomly divided into training and testing, the model is evaluated as many times as desired by the user, and the overall result is averaged. When this method is combined with the leave-one-out method, the repeated leave-one-out random subsampling cross-validation method is created, in which one set is randomly selected for testing and the rest for training, the evaluation is performed, and the result is averaged. The next rank is grouped stratified k-fold cross-validation

TABLE XII
SOFTWARE USED ALONG WITH THEIR APPLICATION AREA

Reference number	SOFTWARE or LANGUAGE	Application area
26	MATLAB, C, C++	Classification
32	LABVIEW	collecting data from the acquisition system to display the data mode, & preprocessing
47	C++	Not precisely defined
55	Scikit-learn Python	Classification
56	MATLAB	Not precisely defined
89	Tensor flow- keras	Classification
91	Scikit-learn Python	Classification
97	MATLAB	Data preprocessing including labeling, segmentation, classification, & evaluation
99	PYTHON	Not precisely defined
103	MATLAB	Preprocessing, segmentation, feature extraction, & classification
104	MATLAB	feature reduction, classification, & graphical user interface
110	WEKA, MATLAB	MATLAB for preprocessing and feature extraction, & WEKA for feature selection and classification
115	MATLAB, C, C++	All three are used for segmentation. MATLAB is also used for classification
118	Scikit-learn Python	Classification
123	WEKA	Classification
135	MATLAB	Preprocessing, data fusion, & classification
136	MATLAB	Preprocessing, segmentation, feature extraction, & classification
137	MATLAB, C++	MATLAB for classification, & data Capture in C++
138	C, C++, objective C	Preprocessing, segmentation, feature extraction & classification algorithms are written in C & C++. Objective C for graphical user interface.
139	WEKA, MATLAB	MATLAB & WEKA for classification and evaluation, & WEKA for feature selection
140	C, MATLAB	Not precisely defined
141	MATLAB, LABVIEW	LABVIEW program for graphical user interface design, & MATLAB for classification
142	MATLAB, LABVIEW	LABVIEW program for graphical user interface design, & MATLAB for classification
143	MATLAB, SPSS	MATLAB for data preprocessing and feature extraction, & SPSS for Statistical Analysis
144	MATLAB, C	Preprocessing was done by a software developed in MATLAB & C
145	MATLAB	Collecting sensor data using MATLAB script
146	MAX	Creating a musical environment
151	WEKA	Classification
158	WEKA	Classification
160	WEKA	Classification
161	WEKA	Classification
162	WEKA	Classification
174	MATLAB	Preprocessing, & classification
189	MATLAB	classification
193	MATLAB	MATLAB acquisition graphical user interface for collection & preprocessing of the data
199	WEKA	Classification, & evaluation
200	MATLAB	Classification
203	MATLAB	Classification
204	MATLAB	Data acquisition, & Classification
217	WEKA	Classification
221	WEKA	Classification
233	MATLAB, SPSS	MATLAB's usage not precisely defined, & SPSS for Statistical Analysis
243	MATLAB	Classification
247	MATLAB	Classification
249	MATLAB	preprocessing, feature extraction, & classification
250	MATLAB	Feature extraction, feature reduction, & classification
251	MATLAB	Classification
252	WEKA	Classification
254	MATLAB, RAPID MINER	MATLAB for segmentation, & RAPID MINER for classification
257	MATLAB	Classification
261	MATLAB	Classification
262	MATLAB	Classification
270	MATLAB	Feature extraction, & classification
277	MATLAB	Data acquisition, feature extraction, & classification
279	WEKA	Classification, & evaluation
282	WEKA	Classification, & evaluation
287	MATLAB	Segmentation, feature extraction, & classification
304	Scikit-learn Python	Classification
305	Tensor flow- keras	Classification
307	Tensor flow- keras	Classification
308	Python GUI	Data acquisition

with two uses and a share of about 2%. However, first, we explain the stratified k-fold cross-validation, which has a 1% share with one use. This method is a variant of k-fold that provides stratified folds, which means that each fold has the same percentage of samples with a given label. Grouped stratified k-fold cross-validation benefits from the advantages

TABLE XIII
EVALUATION METHODS

Reference number	Evaluation methods
26,27,59,106,151,156,191,200,229,249,261,287	Leave one subject out cross validation
56,73,89,95,96,99,123,156,158,159,160,161,162,189,200,207,212,222,237,241,273,279,302	10-fold cross validation
73,95,105,109,117,125,143,216,225,250,258,262,276,292	Leave one out cross validation
26,48,104,292,294	K-fold cross validation
46	Biased cross validation
101,103,145	Leave one participant out cross validation
156,227,281	2-fold cross validation
74,88,103,118,139,156,203,205,227,254,256,257	5-fold cross validation
55	Subject based cross validation
62,211,255,261,263	Cross validation
63	individual based cross validation
124,206	3-fold cross validation
104	Leave one out test cross validation
104	Hold out cross validation
70	leave one day out cross validation
71	7-fold cross validation
151	USER SPECIFIC training
114,213	Leave one person out cross validation
120,214	4-fold cross validation
73	Repeated leave one out random subsampling cross validation
54,126	Leave one user out cross validation
227	Subject wise leave one out cross validation
124,169	Grouped stratified-K-Fold cross validation
252	Random split k-fold cross validation
252	Stratified k-fold cross validation
171	leave one instance out cross validation
31,97	repeated random subsampling
122,295	6-fold cross validation
282	20-fold cross validation
285	leave one out cross comparison

of grouped k-fold and stratified k-fold at the same time; the first case is a special variant of k-fold that ensures that the same group is not present in both test and training sets, and the stratified variant ensures that each fold has the same percentage of samples with a given label, and generally, folds are stratified. Hold-out cross-validation with one repetition and a share of about 1% is the next method. In this method, which is done only once, the dataset is divided into two unequal parts: the larger part is used as usual for training, and the smaller part is used for testing. User-specific training is used only once and has a share of 1%. In this method, classifiers were trained on each subject's activity sequence data and tested on that subject's obstacle course data [151]. For each iteration of biased cross-validation, a different subset of the available recorded datasets has been chosen for training, and the remaining sets have been used for testing. However, an additional constraint has been applied for each iteration: out of three available datasets per bike repair subject, always choose two for training and the remaining one for testing [46]. This method is similar to the previous method in terms of numerical analysis. The subjectwise leave-one-out method, which statistically has the same conditions as the previous two methods, is placed in a separate category only because of the term subjectwise. Subjectwise is a strategy that is opposed to recordwise. In this case, the training and testing folds carry the information and data of other subjects, while, in the second strategy, the data of the same subject may be included in the training and testing. It is not bad to have a performance comparison between some of the most famous evaluation methods at the end. Before starting, we introduce the concepts of bias and variance in classification. High bias in classification algorithms causes underfitting, and high variance causes overfitting. A compromise can be made between these two cases, which we will not deal with. We start performance comparison with the most widely used method, k-fold cross-validation, and introduce its advantages and disadvantages. This method has less computational time and is, therefore, suitable for use in large datasets. Due to the large amount of data used for training, it has little bias compared to other methods. However, in general, it will not be suitable for use in imbalanced datasets. Leave-one-out cross-validation is a very simple method, but it takes a lot of computing time and should be used for small datasets or when time is not as important to us as other classification parameters. The system may lead to higher bias under this method. Hold-out cross-validation is also one of the simplest evaluation methods, and it takes little computing time, but, because a large amount of data is missing in the training model, it has a high probability of overfitting. This method is also not suitable for imbalanced datasets. Repeated random subsampling cross-validation has the advantage that the ratio of dataset divisions to training and testing does not depend on the number of repetitions or folds. One of the disadvantages of this method is that, due to the random nature of selection, some samples or data may not be selected for training and testing at all, and this method is not suitable for imbalanced data, too.

In Table XIV, we introduce the major types of metrics; the first type of metric is the threshold. These metrics are defined in the paper [152] as follows: metrics based on a threshold and a qualitative understanding of error. These metrics mainly are derived from the confusion matrix, and among the metrics that fall into this category are accuracy, error rate, sensitivity, specificity, precision, recall, f score, geometric mean, macroaveraged accuracy, kappa statistic, and more [152], [153], [154]. Metrics based on a probabilistic understanding of error, i.e., measuring the deviation from the true probability, are the second category of the metrics, and we know them as probabilistic metrics [152]. These metrics include mean absolute error (MAE), mean square error (mse), Brier score, log loss, cross-entropy, rms, MAPE, calibration

TABLE XIV
METRICS

Reference number	Formula or definition	Type of metric	metric
35,54,55,57,59,70,71,73,74,87,88,89,91,92,103,109,114,116,118,123,125,126,138,140,157,158,160,189,193,199,200,201,207,208,214,217,222,241,247,250,251,252,262,294,302,303,304,305,306,307,308,309	$(tp)/(tp+fp)$	threshold	precision
35,54,55,57,59,70,71,73,87,88,89,91,92,103,109,114,118,123,125,126,138,140,158,174,189,199,200,201,207,208,213,214,217,222,241,250,252,294,302,303,304,305,306,307,308,309	$(tp)/(tp+fn)$	threshold	recall
55,57,59,67,87,88,89,91,92,98,103,118,123,140,157,158,189,199,200,201,207,208,241,252,291,292,302,303,304,305,306,307,308,309	$(2*precision*recall)/(precision+recall)$	threshold	F score
26,27,42,60,63,64,103,106,121,122,136,137,139,140,142,151,158,191,194,195,198,199,200,202,203,204,205,211,215,216,246,277	$(correct/total)or(tp+tn/tp+tn+fp+fn)$	threshold	Recognition accuracy
58,95,96,99,100,101,105,110,115,118,122,137,138,139,141,142,145,151,158,159,160,171,189,193,195,198,203,205,207,210,211,233,242,243,244,255,256,257,264,265,267,272,273,276,280,282,284,289,291,302	$(correct/total)or(tp+tn/tp+tn+fp+fn)$	threshold	Classification accuracy
31,35,40,55,57,87,88,89,91,92,104,120,156,157,158,159,161,162,170,190,200,201,212,220,222,224,227,252,263,269,288,303,304,305,306,307,308,309	$(correct/total)or(tp+tn/tp+tn+fp+fn)$	threshold	accuracy
42,46,62,63,137,174,197,199,200,206,221,225,227,248,271,277,281,282,285	$(correct/total)or(tp+tn/tp+tn+fp+fn)$	threshold	Recognition rate
58,192	$(correct/total)or(tp+tn/tp+tn+fp+fn)$	threshold	Classification accuracy rate
167,193,213,282,295	$(fp/tn+fp)$	threshold	False positive rate
99,141,142	$(incorrect/total)or(fp+fn/tp+tn+fp+fn)$	threshold	Error, %of ERROR
31,56,95,97,117,122,141,142,143,145,156,219,224,236,237,238,239,242,243,260,261,266,269,287,292,293	$(tp/tp+fn)$	threshold	sensitivity
31,56,95,97,117,122,	$(tn/tn+fp)$	threshold	specificity

TABLE XIV
(Continued.) METRICS

141,142,143,156,200, 219,224,237,239,240, 242,243,249,260,261, 262,266,269,287,292, 293			
31,58,92,95,124,169, 170,201,222,254,276, 287,295	The area under the corresponding curve	Ranking	AUC
206,209	$(incorrect/total)\,or\,(fp+fn/tp+tn+fp+fn)$	threshold	classification error rate
207,254	$k=(fo-fe)/(N-fe)$ where fo is the number of observed agreements between raters, fe is the number of agreements expected by chance, and N is the total number of observations.	threshold	kappa statistics
210,269,293	$(incorrect/total)\,or\,(fp+fn/tp+tn+fp+fn)$	threshold	classification error
54,73,74,117,126,144 ,168,171,172,173,176 ,248,277,279,285,286	describes the point where the FRR and FAR are equal	threshold	EER
72,116	$(correct/total)\,or\,(tp+tn/tp+tn+fp+fn)$	threshold	Class accuracy
58,151,160,282	$(correct/total)\,or\,(tp+tn/tp+tn+fp+fn)$	threshold	Accuracy rate
143	$(tn/tn+fp)$	threshold	True negative rate
215,229	$(incorrect/total)\,or\,(fp+fn/tp+tn+fp+fn)$	threshold	error rate
220	$(correct/total)\,or\,(tp+tn/tp+tn+fp+fn)$	threshold	recognition ratio
225,246	$(incorrect/total)\,or\,(fp+fn/tp+tn+fp+fn)$	threshold	recognition error
233,234	$(incorrect/total)\,or\,(fp+fn/tp+tn+fp+fn)$	threshold	errors of classification
144,171,248	$falserecognitions/totalattempts$	threshold	false rejection rate
252	$(tp*tn-fp*fn)/\left(\sqrt{((tp+fp)+(tp+fn)+(tn+fp)+(tn+fn))}\right)$	threshold	Matthews Correlation Coefficient (MCC)
170	$\frac{1}{N}\sum_{t=1}^{N}((fp)t-(op)t)^\wedge 2$ Fp=forecast probability Op=acutal output, 0 if it does not happen 1 for when it happens N= total number of forecasting samples	probabilistic	Brier score, Adjusted B
170	$LR+=(TPR/(1-TNR))\,or\,LR-=((1-TPR)/TNR)$	threshold	LR stat
255	$(correct/total)\,or\,(tp+tn/tp+tn+fp+fn)$	threshold	Recognition accuracy rate
171	$(falseacceptance/totalattempts)$	threshold	false match rate(fmr)
122,155	$\frac{1}{N}\sum_{t=1}^{N}\lvert et\rvert=\frac{1}{N}\sum_{t=1}^{N}\lvert(predicted)t-(actual)t\rvert$	probabilistic	mean absolute error
122,155,289	$\sqrt{\frac{1}{N}\sum_{t=1}^{N}(et)^\wedge 2=\frac{1}{N}\sum_{t=1}^{N}((predicted)t-(actual)t)^\wedge 2}$	probabilistic	RMSE
155,276	$\frac{1}{N}\sum_{t=1}^{N}(et)^\wedge 2=\frac{1}{N}\sum_{t=1}^{N}((predicted)t-(actual)t)^\wedge 2$	probabilistic	MSE
172,279	$(correct/total)\,or\,(tp+tn/tp+tn+fp+fn)$	threshold	identification accuracy

TABLE XIV
(Continued.) METRICS

155,289	$\frac{1}{N}\sum_{t=1}^{N}\left\|1-((acutal)t/(predicted)t)\right\|$	probabilistic	MAPE
295	$fn/(fn+tp)$	threshold	False negative rate, false rejection rate (FRR)
167	The micro average uses all classes to calculate the average	threshold	Micro average
87,88,89,91,92,302,304,309	the number of occurrences of each particular class in the true responses	threshold	Support value
167	The macro average, for example, calculates the accuracy for each class separately and then averages the results	threshold	Macro average
170	Expansion from the area under the Roc curve	ranking	c-index, adjusted c
173	$(trueacceptance/totalattempts)$	threshold	True acceptance rate (Genuine Acceptance Ratio)
157	$(tn/tn+fp)$	threshold	selectivity
143,167,282	$(tp/tp+fn)$	threshold	True positive rate
144,248	$(falseacceptance/totalattempts)$	threshold	False acceptance rate(FAR)

(CAL), and more [152], [154], [155]. Metrics based on how well the model ranks the examples are ranking metrics [152]. The area under the receiver operating characteristic (ROC) and precision/recall curves are the most important rank metrics. These are the three main types of metrics for evaluating classifiers, but other metrics exist, which do not fall into these categories. Therefore, in this article, the main evaluation metrics will be presented along with their formula, and those who are interested can refer to the relevant paper to get acquainted with other metrics. You need to know this as a recommendation from us that accuracy in this field is introduced either based on the formula [correct/total] [94], [109] or based on [(tp+tn)/(tp+tn+fp+fn)] [156], [157], [158], [159], [160], [161], [162]; this, metric alone, especially with imbalanced data, is not a good measure of classification performance. Therefore, the authors of this article strongly recommend that other metrics are be used to report classification results too. In Table XIV, we introduce the well-known evaluation metrics, and the following equations provide formulas for the more well-known metrics. Most of the formulas and definitions are derived from [163], [164], and [165]. See [51] for more information on time-based evaluation metrics, such as insertion, overfill, and underfill, and event-based evaluation metrics. These types of metrics are less common in contrast to the other metrics in this field and, thus, are not mentioned in this section. However, in a few of the reference papers of our paper, those metrics have been used along with the metrics in Table XIV. The most widely used metrics are threshold metrics that have been repeated a total of 398 times. As you can see, some of the names presented in the table have the same formulas; however, we presented each possible name for a formula separately in the table. Accuracy is at the top of usage with 151 times of use and a share of 38%. Precision ranks second with 52 uses and a 13% share. Recall ranks third with 46 uses and a 12% share. The f score ranks next with 34 uses and a 9% share. Specificity ranked fifth with 27 uses and a share of 7%. Sensitivity takes sixth place with 26 times of use and about 7% share. EER is in seventh place with 16 uses and a 4% share. The error metric also has a share of 4% with 14 uses. The support value with eight uses has a share of 2%. Other proposed metrics have a share of about 1% or less and are presented in the table only for information. About the probabilistic metrics, we must also announce that they have been used ten times. Root mse (RMSE) is used three times and has a share of 30%. MAE, mse, and mean absolute percentage error (MAPE) metrics are used twice each and have a share of 20%. Brier score and adjusted B are used in one paper and have a share of about 10%. Ranking metrics have been mentioned 14 times in total, 13 of which are assigned to AUC, and the share of this metric is 93%. The c-index and adjusted c also have a share of 7% by being used in only one paper. In Table XV, we will present the methods for obtaining these metrics or, in fact, the graphical evaluation methods. At this level, we want to introduce you to each of these items in the table. Han et al. [166] have described confusion matrices comprehensively. As they have described, “a confusion matrix is a useful tool for analyzing how well your classifier can recognize tuples of different classes. True positive (TP) and true negative (TN) tell us when the classifier is getting things right, while false positive (FP) and false negative (FN) tell us when the classifier is getting things wrong.” Han et al. [166] have described ROC as a visual tool for comparing two classifiers and have clarified that ROC shows the tradeoff between the TP rate (TPR) and the FP rate (FPR). In order to respect the authors of scientific papers, other definitions of ROC are reviewed in this article. Shaafi et al. [167] described ROC that it shows the variation of TPR concerning false alarm rate. Another definition is that the ROC shows the variation of correct acceptance concerning false acceptance [168]. Zinnen et al. [125] consider the ROC as recall changes in terms of precision. Tahafchi and Judy [169] define ROC as the ratio of the true accept rate to the false accept rate. In [95], the ROC curve is expressed as sensitivity changes in terms of FPR. Papers [58] and [170] describe the ROC curve as sensitivity changes in terms of specificity. In the paper [171], in the figure that describes the ROC diagram, the vertical axis is considered to be 1-fmr, and the horizontal axis is considered to be false rejection rate (FRR). In the papers [144] and [172], the ROC curve is plotted in terms of FRR and false acceptance rate (FAR). In [173], the ROC curve is plotted for the genuine acceptance ratio (GAR) and the FAR. The error division diagram (EDD) shows the ratio of the entire dataset, including error classes and other related items. The event analysis diagrams (EADs) show counts of predefined events as a proportion of the total ground-truth event count. The use of these diagrams is not as extensive as other items in the table. For more information, you can refer to papers [114] and [174]. The precision–recall curves are known as a suitable complement to ROC curves, which are less commonly used, and these curves display precision values on the vertical axis and recall values on the horizontal axis for different thresholds. Saito and Rehmsmeier [175] show the advantages of this curve over the ROC curve for imbalanced datasets for binary classifiers. The specificity/sensitivity curve is introduced in the paper [117] and shows the values of specificity in the vertical axis and the sensitivity axis in the horizontal axis, and this curve shows the distribution of sensitivity and specificity for detection accuracy in the paper. To evaluate the performance of the classifier, a curve can be used, which is the FAR diagram versus the FRR. This curve is called decision error tradeoff (DET). The DET curve shows the performance of a biometric system under different decision thresholds [176]. 102 methods of obtaining metrics are mentioned in the relevant table, where the confusion matrix is at the top with 67 repetitions and a share of 66%. ROC is in second place with 20 repetitions and a 20% share. Precision/recall curves with seven repetitions have a share of 7%. Other methods have a share of 3% or less. In the last table of this section, you can see the methods of announcing the results and comparing the results of the metrics, for example, for several types of classifiers, comparing different values of hyperparameters. Announcing and comparing the results by the table have been most used due to their ease of use. However, if we want to go into the statistical analysis of this table in a little more detail, we must state that 171 times the methods of announcing and comparing the results have been presented in the papers, and the table with 116 repetitions has a share of 68%. The bar chart with 41 repetitions has a share of 24%, the box plot with six times of repetition has a share of 4%, the scatter plot with five repetitions has a share of 3%, and the other two methods have a share of 1% or less. These methods include the cumulative matching score (CMS) curve and the cumulative match curve (CMC), which can be used as metrics or for announcing and comparing recognition results. Now, we are familiar with all the steps, and we can easily do projects related to activity recognition, GR, and GA based on this information. The authors hope that this article would be of great help to engineers, students, and researchers interested in doing a project in the field of movement classification.

III. Overview of Findings

First, we must state that data collection for human motion analysis is usually done with three methods: 1) wearable sensors; 2) specialized systems, such as Vicon (Vicon Motion Systems Ltd., Oxford, U.K.) or Optotrak (Northern Digital Inc., Waterloo, ON, Canada); and 3) Kinect systems [6]. The second and third cases usually create image and video data and have limitations [6], [51]. Specialized systems, such as Vicon or Optotrak, have high accuracy when operating in controlled environments [6]. These systems can provide a large amount of redundant data. Also, these systems are very expensive compared to the other two. Ambulatory systems, such as those using a Kinect (Microsoft Corporation, Redmond, A, USA) to capture human motion, are set in relatively uncontrolled environments and have a restricted field of view. These systems have a restricted margin of maneuverability and are intended for indoor use mainly. In contrast, wearable sensors have the advantage of being portable and suitable for outdoor environments [6]. It is not bad to announce the other reasons for favoring wearable sensors. In addition to being portable and cheap, it can be said that these sensors are more ubiquitous, and it is easy to use them. The use of these sensors does not require special knowledge. It is easy to teach the user how to use the wearable system with a little training. By equipping the wearable system with a memory, it is possible to analyze the wearer's behavior at any time. Considering the variety of wearable sensors and the possibility of measuring different parameters by these sensors, it can be said that these sensors provide more diverse information compared to other methods. Wearable systems are easier to update, can adapt to changes in society, and can advance with fashion. For these reasons, we have focused on human motion analysis by wearable sensors, and we have tried to review movement classification by wearable sensors. Of course, there are challenges in this area that will be discussed in the following. The purpose of this section is only to present a summary of the findings, and for the readers to be familiar with the main results of the statistical analysis by reading this section and for a more general understanding, refer to the previous sections. In this article, a wide variety of papers have been studied, each of which is related to three areas of movement classification, namely, activity recognition, GR, and GA; for the first time, all three of these areas have been addressed simultaneously, and other review papers in this area have only addressed one area [10], [14], [19], [20], [50], [51]; and in the common concepts associated with the steps, our research is much broader. In identifying the movement classification chain, the number of algorithms proposed for each part of this chain in this article is very large and very diverse. For example, only Table XVII introduces many different types of sensors available in this field, and the set of sensors introduced in this article is very diverse, which is the leader compared to the existing review papers in this field, because different sensor categories are presented, and the state-of-the-art papers in human motion recognition fields or IoT-based wearables area may only mention some of these categories or sensors [3], [19], [20], [51]. A large number of datasets related to movement classification are presented in this article. We identified 18 preprocessing actions, which is a significant number. After identifying the levels of data fusion, we introduced many different algorithms for each level. Different algorithms have been identified for signal segmentation. Topics related to the feature extraction step have been discussed extensively. Also, a comparative analysis has been done on the performance of some feature extraction methods. It has been tried to introduce all kinds of algorithms related to feature selection. Each of the algorithms is defined separately, and the way they work is specified. Various feature reduction algorithms have been proposed. The functioning of the algorithms has been investigated, and a comparative analysis has been done on the performance of some of them. In the classification step, a very large number of algorithms have been introduced if we ignore the classic classification algorithms; a wide range of machine learning algorithms have been identified along with their types; and in this field, papers can be found that only deep learning algorithms have been discussed [19], [50], but these algorithms are only a part of machine learning algorithms. The functioning of widely used or famous classification algorithms has been fully investigated and analyzed. In evaluation, a wide range of metrics has been proposed along with their formula and type. There are many different evaluation methods, and a standard definition is provided for each one. Also, the most famous and most used of them have been examined in terms of performance. The methods of obtaining the metrics and the methods of announcing and comparing the results have been fully investigated in this article. In fact, the authors have tried to provide the readers with a suitable guide to continue and conduct research in the field of movement classification. The reason for this effort is that there are major reasons that force the world community to refer to wearable sensors, and also, the need increases with human motion recognition. Due to the aging of the global society, and the loneliness and inability of patients to attend medical centers, by using wearable sensors to detect human movement, patients can be saved from visiting these centers, and costs can be reduced. Since the doctor can notice changes in the patient's movement pattern and may call the patient to inquire about their condition. Alternatively, a special alarm may be activated to notify the patient. Even considering that the world society is facing the problem of obesity, it is easy to know the weight, degree of obesity, or the discomfort of their organs by analyzing the walking of people or the speed of movement. In general, identifying human activity through walking, examining the pattern of human walking make people aware of the current state of health, and based on these, decisions can be made about the future state of health. Of course, this issue is not exclusive to human walking. Many movements of the human body can be used to evaluate a person's body condition and health. Thus, GR, GA, and activity recognition by wearable sensors have many applications in the fields of medicine, education, entertainment, sports, and games. With these explanations, the human movement recognition chain or the movement classification chain has been identified in general, and each part of this chain has been discussed in detail. The numerical analysis has been fully presented in relevant sections, and now, we only qualitatively repeat the results. After categorizing the human body motion analysis with wearable sensors in a general and global format, and briefly explaining why we have to combine this technology with IoT, the common steps for doing a project in each section of movement classification were examined. All the steps are presented together with the corresponding algorithms in this article. The purpose of this article was to clarify all the common steps of the project in the movement classification of the global chart so that the readers of this article can easily do the project in the field of movement classification. Common steps of project implementation in the studied papers are data collection, data fusion, preprocessing, segmentation, feature extraction, feature selection, feature reduction, classification, and evaluation. Some of these steps may not have been used in all papers. Using the corresponding tables, the most commonly used topics and algorithms can be easily identified. Bar charts are also used for a better understanding of some steps of the operational plan. As can be seen from Table XVII and the bar chart, the most widely used sensors in this field are accelerometers, gyroscopes, EMGs, force sensors, and pressure sensors, respectively. In the following, we have specified the category of sensors, and we have specified the most used sensor categories through a bar chart. There are also preprepared datasets produced by universities, companies, and so on, which are a great help in creating papers in this field, and with these datasets, there is no need for a data collection step. For the preprepared datasets, we specified both the most used sensors and the most used category, and the results were almost similar to Section II-A1. A side result of this section is to specify the importance of human movement in the field of activity recognition because most of the datasets related to HAR contain data on walking activities with different styles. Then, in this article, various sensor fusion strategies were discussed, and the corresponding algorithms were identified. We then introduced the signal preprocessing, specified the preprocessing actions in the table, and then identified the most commonly used actions using a bar chart. The most common preprocessing actions are filtering, data normalization, sensor calibration, amplification, segmentation, smoothing, rectification, interpolation, labeling, and drift removal, respectively. In Section IV, along with the definition of signal segmentation, we present various segmentation algorithms. Next, feature extraction methods, dominant feature type, and feature domain are specified. Fourier transforms and WT are the first and second most widely used feature extraction methods, respectively. The signal-based statistical feature is the dominant feature type, and the other feature types are used lesser than this feature type. The time domain and the frequency domain have the highest number of uses as feature domains. The issue of feature selection in the next step is examined. Feature selection methods have been introduced, and from the reviewed papers, we find that filter and wrapper methods have been two of the authors' favorite methods for feature selection in this field. Dimensional reduction algorithms were also examined, and PCA is a dominant algorithm in this field. For classification, after defining and clarifying the purpose of using it in papers, we recognized it as the last step of the project and completely identified the algorithms used in the papers in Table XI, and then, we exhibited the most commonly used algorithms and their types by bar charts. SVMs are the most widely used classifier, followed by KNN and Bayes derivatives (naïve Bayes, Bayes net, and so on). Decision trees, HMM, and random forest are also preferred by authors for classification, and they hold the next ranks. Supervised algorithms are at the top. Probabilistic algorithms have taken second place. Combined algorithms are ranked third, and rule-based and unsupervised machine learning algorithms are ranked next. In the section, software or language used and their field of application, we introduced the software or languages that can be utilized in movement classification steps. For each step or, in some way, each piece of the human motion detection chain, we specified the software or language used. Most of the papers in this field mentioned the software used for classification; for classification, the first rank of the most widely used software goes to MATLAB, and another software that ranks second is named Weka. Both well-known brands are widely used to implement classifiers, while they can be active in other steps of the project. Side results can also be obtained about other software or languages used in papers related to the field of movement classification, which has been omitted from the presentation due to their low importance. In the last part of this article, we have announced the methods of evaluating the performance of the model, the relevant metrics and types of them, graphical metrics or methods of obtaining the metrics, and methods of presenting and comparing the results. In general, evaluation cannot be identified as a separate step, and it should be considered as a part of the classification, but, because there are different concepts and parts related to it, we dealt with it separately. Different evaluation methods are presented in this article, and in general, k-fold cross-validation is considered a popular evaluation method. The most widely used metric types for evaluating the performance of classification are threshold-based metrics. It is recommended to use several metrics to evaluate the performance of the model to have a more accurate understanding of the performance of the model. In Table XV, we have presented the methods for obtaining these metrics or, in fact, the graphical evaluation methods. The confusion matrix with a relatively large difference is the most widely used method to obtain the metric. ROC and precision/recall curves are ranked next in terms of usage rate, respectively. In the last table, the methods for announcing the results and comparing the results of the metrics are presented. Announcing and comparing the results by tables are the most common methods. Bar charts also are used but not as many as the previous method. Box plots are placed in the next rank in terms of usage. These results are presented quantitatively in more detail in the relevant sections, be careful that the numbers presented are approximated and rounded, and this approximation may cause the sum of the share percentage to not be 100%, but there will be no change in the overall results.

IV. Related Concepts on the Internet of Things in This Field

The IoT has many different applications and is not limited to motion recognition. The IoT provides insights into many applications in various sectors of a variety of industries and businesses. It brings efficiency and safety, and can revolutionize the way many businesses and industries operate [2]. In this section, we intend to briefly present the structure of the IoT, its implementation methods, and concepts related to machine learning or artificial intelligence in general. The IoT can have various components, the most important of which are sensors/devices, gateways and connections, cloud and database, analytics, and user interface. The first component is related to collecting and sending information by objects. Sensors can be temperature, accelerometer, compass, proximity, humidity, pressure, light, or any other sensor. As mentioned, these sensors can be used alone or fused with other sensors. All the sensors mentioned in the data collection section can be used in the first component, and when talking about the device, you can easily remember things such as smartphones. The second component is related to how the data reaches the cloud and is related to data flow management. There are various methods for connecting sensors to the cloud, such as Wi-Fi, Bluetooth, and ZigBee, and the choice of each of them depends on the application of the IoT. The third component provides a location to store and access IoT data. In the analytics section, the data of the sensors and the device are examined, and various decisions are considered according to the conditions of the data. The user interface section informs the end user of the results of the analysis and, actually, the decisions made or the conditions and also gives the user the ability to perform some operations related to the conditions. IoT can be implemented using many IoT connectivity schemes that connect an IoT device to other devices through the Internet. The Internet connection can be either wired or wireless [2]. Wired and wireless communications have their advantages and disadvantages, and should be chosen depending on the application. Understanding the benefits and drawbacks of wired and wireless connectivity schemes enables us to make an informed decision regarding IoT implementation [2]. Wired connections are reliable, fast, and secure. They are more reliable than wireless connections since they are less prone to packet loss as a result of path loss or interference from other electronic devices. However, they suffer from the higher cost of implementation and lack of mobility support. Scalability is also another problem with wired networks. The wired IoT network is only practical if IoT devices not only are close to each other to reduce the cabling cost but also at least one of them is located close enough to a wired Internet access point. For many IoT applications, wired connectivity is not very practical, and wireless IoT implementations are the common solutions [2]. For a wireless connection, there is a need for an IoT gateway, especially for short-range communications [2], [3]. IoT gateway connects sensors, devices, and so on to the internet at the network's edge and can perform computing locally [2]. Regardless of whether the implementation is wireless or wired; there are four types of data communication in IoT: device-to-device, device-to-cloud, device-to-gateway, and back-end data shape. Only wireless protocols related to each model are presented because of practicality; for wired protocols, refer to [2]. In device-to-device communication, two or more devices are connected directly to each other. Bluetooth protocol is one of the most widely used protocols in this type of communication. In connecting the device to the cloud, a device is directly connected to the Internet cloud. Some of the widely used protocols in this connection are Wi-Fi and low-power wide-area networks (LPWANs). In the

TABLE XV
METHOD OF OBTAINING THE METRIC

Reference number	Method of obtaining the metric (metric tool)
26,27,40,46,48,56,57,59,67,72,87,88,96,98,99,100,101,103,105,109,110,118,120,122,137,138,140,145,151,156,157,160,189,191,193,194,200,202,203,205,207,208,212,214,216,227,241,246,252,264,272,273,277,279,282,291,292,293,294,302,303,304,305,306,307,308,309	Confusion matrix
210	Error confusion matrix (ECM)
31,58,95,124,125,144,167,168,169,170,171,172,173,174,196,201,222,241,276,295	ROC
54,73,74,109,140,217,222	Precision / recall curves
114,174	EDD
114	EAD
117	specificity/ sensitivity curve
176,285,286	DET

TABLE XVI
METHODS FOR ANNOUNCING AND COMPARING RESULTS

Reference number	Methods
26,27,31,35,42,46,54,55,58,59,60,64,67,70,71,72,87,88,89,91,92,94,95,96,97,104,109,110,115,117,118,122,123,124,126,136,137,138,139,140,141,142,143,144,151,158,159,160,161,162,168,169,170,189,190,191,192,194,195,197,199,205,209,210,211,213,214,217,220,225,234,237,239,240,241,242,244,245,250,252,254,256,257,258,260,261,262,263,264,265,266,267,268,269,275,276,279,281,282,285,287,288,289,291,292,293,294,295,302,303,304,305,306,307,308,309	Table
27,59,62,87,88,89,91,92,103,104,118,136,139,157,158,189,205,206,207,208,211,213,227,229,244,246,254,263,267,268,271,273,276,284,289,295,302,304,306,308,309	Bar chart
56,106,145,243,272,288	Boxplot
97,250,253,264,283	Scatter plot
144	CMS curve
281,285	CMC curve

model of connecting the device to the gateway, there is an application on the desired gateway that acts as a communication interface. One of the protocols used in this model is Wi-Fi. The back-end data-sharing model is the extension of the device to the cloud connection. In this connection, the user can use cloud data along with data from other devices and sources. Artificial intelligence continuously improves performance and decision-making capabilities and enhances the true potential of IoT. Artificial intelligence or specifically machine learning is an integral part of motion recognition. In general, the wearable IoT has many applications in motion recognition. We have tried to examine some of the algorithms and techniques available in these applications briefly. Motion recognition by IoT-based wearable sensors has applications in health, gaming, sports, safety, and so on. The health wearable IoT device is mainly used for remote patient monitoring, treatment, and, in some cases, rehabilitation purposes. The sensors such as blood pressure, temperature, accelerometer, and heart rate monitor collect health-related data, and the user/patient's health information will be sent to the Internet for further analysis. In many applications, wearable devices are connected to smartphones to analyze the collected data and then transmit it to a cloud computing-based framework, such as Microsoft Azure or Amazon Web Services (AWS) in order to store, process, and analyze the data [3]. Detection and prevention of falls are other applications of wearable sensors based on the IoT. To be able to detect falls, usually, inertial sensors such as a gyroscope or accelerometer are used. The fall detection system must be fast enough to detect fall fast to be beneficial. However, in order to detect fall events accurately and minimize FPs, the fall detection system must differentiate between a fall and other daily activities [3]. Machine learning algorithms, such as SVM, along with other motion recognition steps, such as feature extraction and feature selection, can be used to detect falls from raw sensor data, and this is one of the important issues of artificial intelligence related to the IoT. Other applications have similar conditions, but we tried to examine the most used applications. In general, the benefits of IoT by adding human-like awareness and decision-making using machine learning algorithms can lead to increase efficiency and improve motion recognition.

V. SCOPE FOR FUTURE RESEARCH

Usually, to be able to provide a scope for future research in any scientific subject, we must fully understand the challenges and opportunities in that field. There are many public challenges in the field of human body motion recognition by wearable devices. Challenges such as power consumption or battery life, ergonomic designs, user safety from wireless transmission radiation, miniaturization, memory capacity, privacy, security issues, training the end user to use these devices and trust them, equipment flexibility, cheap and affordable price, user comfort, wearability issues, and reliability are commonly raised when commercializing products. The discussion of creating standards in user interfaces and related application updates is also somehow included in this category. Dealing

TABLE XVII
SENSORS

Reference number	Sensor category	Sensor type
20,26,27,35,36,40,46,47,48,55,57,60,62,64,70,71,74,94,95,96,97,98,99,100,103,104,105,106,109,110,114,116,117,119,120,122,123,124,126,136,137,139,141,142,144,145,146,151,156,157,158,159,160,161,162,167,169,170,171,172,173,176,189,190,192,193,194,195,196,197,198,199,200,201,202,203,205,206,207,208,209,211,214,216,217,218,219,220,221,228,231,233,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255,260,261,264,265,266,267,268,270,271,274,276,277,279,281,282,283,285,286,287,289,290,291,292,294,302,308	Motion sensor	Accelerometer
20,40,46,48,55,57,60,64,94,96,98,100,106,116,119,120,124,137,141,142,143,145,146,156,157,158,159,160,161,162,167,169,189,192,193,194,195,200,202,203,207,208,217,221,228,237,238,241,242,243,245,246,247,248,251,253,254,261,266,267,269,274,275,276,277,278,279,287,289,294, 308	Motion sensor	Gyroscope
26,27	Bio sensors & chemical sensors	Ventilation sensor
20,60,116,145,189,193,221,237,308	Motion sensor	Magnetometer
32,35,40,47,97,189,214,223,246,253,262,272,273,274	Pressure & force sensors	Pressure Sensor
32,36,40,47,70,72,189,214,223	Bio sensors & chemical sensors	Temperature
135,191,198,227	Pressure & force sensors	Strain
46	motion detectors	Ultrasonic Sensor
20,36,103,104,106,118,124,135,136,158,169,193,204,205,206,211,225,233,234,239,240,241,249,276	Bio sensors & chemical sensors	EMG
20,101,105,143,146,217,226,228,244,259,260,274,275,278,284,288,289	Pressure & force sensors	Force
57	Bio sensors & chemical sensors	PPG
36,57,223	Bio sensors & chemical sensors	ECG/ EKG
99	Bio sensors & chemical sensors	Heart Rate
48	Other categories	Lidar
32,35,40,72,214	Bio sensors & chemical sensors	Humidity
36,40,201,229,230	Positioning & tracking sensors	GPS
40	Positioning & tracking sensors	Bluetooth Beacon
47,109,138,168,199,214	Audio & visual sensors	Microphone
201	Motion sensor	Orientation
35,47,70,74,139,201,214	Optical & light sensors	Light
146,208,228,268,274	Bend sensors	Bend/Flex Sensor
210	Optical & light sensors	Linear Optical Gesture Sensor
114,140,174,213	Bio sensors & chemical sensors	EOG
35,74,214	Motion sensor	Compass
215	Audio & visual sensors	Camera
217	Positioning & tracking sensors	UWB Tag
146,222	Proximity sensors	Capacitive Sensor for measuring height, distance.
35	Audio & visual sensors	Audio
47	Motion detectors	Passive Infrared
36,231,232	proximity sensors	Infrared
47	Bio sensors & chemical sensors	Carbon Monoxide
47	Other categories	Touch
36	Audio & visual sensors	Audio input and Output devices
36	Bio sensors & chemical sensors	Galvanic Skin Response Sensor
224	Bio sensors & chemical sensors	Oximetry
146,228	Pressure and force sensors	PVDF
228,274	Proximity sensors	Electric Field Sensor
20,42	Positioning & tracking sensors	Tracker
20,280,284	Motion sensor	Goniometer
19	Bio sensors & chemical sensors	MMG
31	Bio sensors & chemical sensors	EEG

with such commercial or public challenges should be the responsibility of economists, marketers, researchers, managers of famous companies in this field, and even governments. By addressing each of these commercial or public challenges, researchers and students can help solve a societal problem by providing a solution to the challenge. Our goal is not to deal with these types of challenges, and we have another intention of providing the scope of the future section, but we will provide examples that address some of these challenges briefly. You can find solutions for other challenges in different papers. Since information security and privacy are one of the most important challenges in this field, governments must consider strict laws for stealing information from wearable devices and implement security policies. As another example in the field of power consumption or battery life, Bluetooth low energy has been proposed instead of Bluetooth in mobile phones and has not been very successful, but, with the progress in the semiconductor industry, integrated circuits with lower power consumption can be produced. Also, energy harvesting technology provides additional means to extend battery life [177]. Our goal in this section is to address the technical challenges of identifying the human body's movement. In general, data collection from different people is very time-consuming and challenging. Especially, since the ground-truth data need to be collected for the training of the supervised classifiers, there are various issues related to the sensors that should be addressed, such as accelerometer bias [178], magnetometer dysfunction in the presence of the intrusive magnetic field, and the loss of GPS signals. The presence of noise in the output of sensors is challenging because the presence of noise generally affects the recognition performance. In the paper [97], it is stated that removing the noise from the corresponding wearable sensors has improved the classification performance of walking on the stairs. It is stated in the paper [144] that the presence of noise in the output of the accelerometer generally causes problems in identifying the phases of gait. In general, it can be concluded from the studied papers that noise disrupts the motion recognition process and weakens performance by reducing recognition accuracy. The motion recognition process should be robust to noise. Generally, a higher signal-to-noise ratio will provide better results. The noise of the sensors and the data noise, in general, deteriorate the classification performance because the machine learning algorithm or any other classification algorithm can identify the noise as a pattern, so misleading generalizations begin and eventually cause the false identification of patterns. Classification accuracy reduction is only one of the problems that noise will cause, complicating classification, overfitting, increasing training time, or maybe the whole system's execution time, and so on are other problems caused by the noise. In this section, we want to talk a little more about the concept of noise in data preprocessing because the filtering action, which is generally mixed with the concept of noise, is the most widely used preprocessing action. Eliminating noise, in general, is challenging, so we try to define the challenges related to the concept of noise in this field to some extent. Noise is present in all the wearable sensors presented in this article. For example, there is noise in the output of EEG, accelerometer, gyroscope, EMG, EOG, and other sensors [31], [88], [96], [104], [114], [144]. In the paper [88], it is stated that, in general, the reading in IMU is noisy due to environmental noises, self-occlusions, reduced accuracy due to fast movements, and so on in data collection. In the paper [104], it is stated that noise should be removed from the EMG sensor, and a common problem in sEMG is motion artifact that produces low-frequency noise. This type of noise is caused by the movements of the muscles under the skin, and the movement of the electrode relative to the skin is another reason. There are various sources of noise in wearable sensors. Therefore, identifying noise sources in the output of different sensors is a challenging task, and it is very important to deal with it. In the studied papers, the presence of intrinsic noises of sensors and motion artifacts, respectively, has been challenging for researchers. Intrinsic noises are the noises that exist in the output model of the sensors. For example, the output of three sensors, an accelerometer, a gyroscope, and a magnetometer, is affected by bias, scale factor, and white noise. As stated, motion artifacts generally change the performance of the sensors and occur when the user's movements affect the placement of the sensors or other factors related to sensors. The next source, which is perhaps less mentioned than these two, is the environmental factors, examples of which were mentioned a little earlier for GPS and magnetometer. Although these three cases are the most famous causes of noise in wearable sensors, the main challenge for researchers is to fully understand the causes of noise in the sensors, and they choose motion recognition. The next challenge is choosing the right filter, which is somehow related to the recognition of the sensor noise. According to the studied papers, the filters that are used to remove the noise are a notch filter, a linear Kalman filter, an EKF, an infinite impulse response filter, a finite impulse response filter, a high-pass filter, a bandpass filter, a low-pass filter, a median filter, and so on. The need to produce preprepared datasets specific to GR is strongly felt. In the field of GR, it is necessary to collect the sign language datasets, publish the datasets, and make them available to researchers for further research. Activity recognition through human walking has led to the production of many datasets, which shows the importance of human walking for activity recognition and GA; this is because human walking is a basic activity of daily living, and human walking or gait is defined as a particular way or manner of moving on foot [179] and has many applications in health monitoring, sports, rehabilitation, video surveillance, and so on. If we try to provide examples, we must announce that, according to the studied papers, human walking activities recognition has many medical applications in the fields of poststroke rehabilitation, detecting gait abnormality, Parkinson's disease rehabilitation, fog detection, analyzing neuropathy disorders, pathological gait assessment, walking stability detection in older people, postural stability analysis, postinjury rehabilitation, and so on, so there is a need that the data collection are to be more application-specific, which means that researchers collect data related to specific diseases, sports, and so on. Data fusion for different sensors from different categories is very challenging. Regardless of the specific model, the challenges in this field should be identified. Challenges related to data fusion mainly include data association and management, sensor uncertainty, dynamic system modeling, and system validation [108], [111]. They arise from the inherent uncertainties in the sensory information, which are caused by not only device imprecision but also noise sources within the system and the sensor itself [108]. One of the examples of the uncertainty of sensor data can be missing data. Target environments and natural behavioral conditions can be responsible for these challenges, too, especially in system validation challenges [111]. The strategies of data fusion should be capable of dealing with these uncertainties and result in a consistent perception efficiently [108]. A proper data fusion mechanism or strategy is expected to reduce overall sensory and even nonsensory uncertainties and, thus, serve to increase the accuracy of system performance and find the optimal structure for the structure of the recognition system [106], [108]. Perhaps, one of the most important factors in the optimality of the structure and choosing the best strategy is the recognition delay, which affects real-time performance [106]. The other challenge is choosing between data fusion algorithms that can be used at the same level, and many factors are effective in this choice, but we will try to answer this challenge by mentioning an example. Although we have addressed this issue to some extent in the explanation of the previous challenge, in this example, we are going to somehow evaluate the performance of the Kalman filter and the EKF. Of course, these two algorithms have not been used in the studied papers on a common system according to Table IV, but an indirect comparison of their performance will be useful. In general, to compare the performance of estimation algorithms for data-level fusion on a shared system, usually, the accuracy of their performance on the system should be considered, and after that, issues such as computational load and ease of implementation can be considered. These cases can be generalized to the selection of fusion algorithms at all three levels. However, in any case, prechecking a series of issues related to algorithms will be effective in choosing them. The Kalman filter is a recursive type estimator and is utilized in many engineering applications. Traditional Kalman filters need an accurate linear model of both the system dynamics and the observation process to be optimal in a least-mean-squared-error sense [108]. The main advantages of the Kalman filter are its computational efficiency and ease of implementation. The main limitations of this filter are its restriction to linear and Gaussian assumptions and low accuracy [108]. EKFs linearize the system model using Taylor series expansions around a stable operating point [108] and overcome the limitation of the linear Kalman filter. The main advantages of the EKF are computational efficiency, intuitiveness, ease of use, and stability in practical estimations. The main limitations of these filters are being limited to Gaussian noise and the need for the derivability of the system model and the measurement model [108]. Therefore, when it is necessary to choose an algorithm for data fusion in each of the three levels, it is necessary to know the advantages and disadvantages of that algorithm or even its structure. According to the statistical analysis in this article, the Kalman filter and its nonlinear derivatives are an important part of data-level fusion algorithms; although various algorithms are announced in Table IV for data-level fusion, the Kalman filter algorithm is by far the most widely used. To the list of algorithms in the mentioned table, you can also add algorithms such as the complementary filter, which is a data-level fusion method that consists of a low-pass filter and a high-pass filter, and is generally widely used in attitude estimation. Therefore, we should also talk a little more about how to use this filter in the field of movement classification. In general, it can be said that the Kalman filter is often used to fuse accelerometer and gyroscope information to provide better estimates, an example of which is the use of the KF to detect postural sway during quiet standing (standing in one spot without performing any other activity or leaning on anything) [111]. For biomechanical modeling, the Kalman filter can be used to estimate the states [111]. Therefore, fusing accelerometer, gyroscope, and magnetometer data to obtain related directions and angles can provide comprehensive information for movement classification. Therefore, by placing the mentioned sensors in different places of the body and using the Kalman filters to obtain orientation-related concepts, such as quaternions and Euler angles, different activities can be recognized. Since the authors of the papers studied in Table IV have not fully mentioned the linear or nonlinear model used for data fusion by the Kalman filter in their papers, we present two models for use in data fusion of the mentioned sensors, which, in general, is used in the movement classification. The authors of these papers have only mentioned the general names of the algorithms, and we have shown these names in the relevant table to respect them. In the paper [180], a quaternion-based EKF is developed for determining the orientation of a rigid body from the outputs of a sensor, which is configured as the integration of a triaxis gyroscope and an aiding system mechanized using a triaxis accelerometer and a triaxis magnetometer. The suggested applications are for studies in the field of human movement. In the proposed EKF, the quaternion associated with the body rotation is included in the state vector together with the bias of the aiding system sensors. Moreover, in addition to the in-line procedure of sensor bias compensation, the measurement noise covariance matrix is adapted to guard against the effects that body motion and temporary magnetic disturbance may have on the reliability of measurements of gravity and the Earth's magnetic field, respectively [180]. Another version of the quaternion-based Kalman filter can also be found in the paper [181]. The paper [182] presents a successful design of a wearable device to monitor walking patterns. It offers a low-cost wearable fitness monitoring device utilizing a six-axis IMU embedding a three-axis gyroscope and a three-axis accelerometer. The Kalman filter has been employed to provide reliable angle measurements that, in turn, are used to estimate the stride length. In this article, a linear Kalman filter has been used to measure foot angles; the system states in the linear Kalman model were the angle of the accelerometer and the bias value of the gyroscope; and the measurement model consisted of the angle of the accelerometer. In the process of signal segmentation for choosing the length of the window, factors should be considered so that both feature extraction is done well and the system does not suffer from delays. However, it is better to examine the performance of different segmentation algorithms to identify the effective factors in choosing a better segmentation algorithm. Comparing the performance of the classifier under different segmentation algorithms is one of the factors in choosing the proper segmentation algorithm [109], [114]. In the paper [109], an algorithm for segmentation has been devised, its performance has been compared with sliding windows by different approaches, and it has a better performance than sliding window segmentation in terms of precision and recall. The paper [114] has investigated the performance of two segmentation algorithms, i.e., sliding window and head-based segmentation using the SVM classifier performance. By examining the classification results, it has been stated that, by using a head-based segmentation scheme, precision and recall percentages are increased. It has also been announced that this algorithm is computationally lightweight. However, the lack of adaptation to different head movements while reading (depending on the fact that the head should be down while reading) is one of the limitations of this method, especially in short reading sequences. In the paper [115], one of the important factors in choosing a signal segmentation algorithm is the computational load of the algorithm. For performance comparison reasons, three algorithms, which are commonly used for segmentation, have been implemented and applied to the same dataset: 1) SAX; 2) SWAB, and 3) a GA-based approach. A table indicates the CPU time required by each of the algorithms. Their availability for real-time execution is compared. According to the above literature review, computational load and overall recognition performance are the main reasons for choosing the superior segmentation algorithm. One of the most important technical challenges can be the feature extraction step. Such a step imposes the need for feature selection/feature reduction steps. This is because this step is very time-consuming. As mentioned earlier, we should go for deep learning classification algorithms because these algorithms eliminate the need to extract the features. Perhaps, another reason that increases the need to remove this step is that the extracted features may only perform well in a specific application and are somehow application-specific. The next challenge is to choose the algorithm or method of feature extraction. Although, in the feature extraction section, feature extraction methods have been tried to be fully explained, it is not bad to compare the feature extraction methods because choosing between these methods is also a challenging matter. This issue is addressed by citing an example. Handojoseno et al. [31] investigated the EEG features determined by both Fourier and wavelet analysis in the confirmation and prediction of FOG. In this study, they attempted to find discriminating features by investigating the performance of Fourier-based features and their counterpart in the wavelet domain. This article somehow compares Fourier and wavelet feature extraction methods and has announced the reasons for the superiority of WT. Over the past few decades, wavelet analysis has been developed as an alternative and improvement to Fourier analysis. Its main advantage in analyzing physiological systems is its capability to detect and analyze nonstationarity in signals, and its aspects such as trends, breakdown points, and discontinuity since wavelets are localized in both the time and frequency domains [31]. Even they have declared that the continuous WT has a better frequency (scale) representation compared to the discrete WT. The sensitivity, specificity, accuracy, and the area under the ROC curve of the classification system were calculated by the authors to measure the performance of the features and feature extraction methods. By announcing the classification results, they have compared these methods. In this article, the computational time has also been discussed as a comparative measure of the performance of two feature extraction methods, and this criterion has been examined in two methods and declared that the continuous time WT has limitations for practical use. Thus, general classification performance, computational cost and time, and suitability for the nature of the data are the main criteria in choosing feature extraction methods. It can be said that feature selection is also very challenging, and the main challenge of feature selection is choosing the optimal feature subset, which is very difficult and tiring. To avoid this complicated search operation, three types of feature selection methods were generally introduced. To choose the type of feature selection method, many factors should be considered, which will be discussed in general. However, first, it should be noted that, if we already have an algorithm for feature selection in mind, regardless of its method type, we must first know whether that algorithm is useful for classification or not because some feature selection algorithms are useful exclusively for regression or clustering and are not useful for classification [129]. However, algorithms such as relief or mRmR are used in both classification and regression topics, or the information gain algorithm is only used in the field of classification. This issue should be considered especially in the selection of filter methods. Although the feature selection algorithm specific to regression or clustering may also be used in the subject of classification, caution must be observed. This caution might act as a catalyst to speed up the work. For comparing the performance of the methods, several datasets should be employed, aiming at reviewing the performance of three methods in the presence of a crescent number of irrelevant features, noise in the data, redundancy, and interaction between attributes, as well as a small ratio between the number of samples and the number of features [130]. Finally, announce which algorithm's classification accuracy or, in general, which type of algorithm's performance accuracy is better. Because there is no silver bullet method [129], it is possible to state the advantages and disadvantages of all three methods in general. The advantages of filter methods are independence from the classifier, lower computational cost than wrappers, being fast, and good generalization ability. The main disadvantage of this method is having no interaction with the classifier. An embedded method interacts with the classifier, has a lower computational cost than wrappers, too, and captures feature dependencies, but its feature selection is classifier-dependent. Wrapper methods are like embedded methods in terms of interactions with the classifier and capturing feature dependencies, but they have high computational costs, they have overfitting risk in classification, and their feature selection is classifier-dependent, too [130]. Future research should focus on optimizing the efficiency and accuracy of the feature subset search strategy by combining earlier the best filter and wrapper methods to produce hybrid methods. Most research tends to focus on a few datasets on which their methodology works. Larger comparative studies should be pursued in order to have more reliable results [129]. In the feature reduction step, it is still necessary to apply different feature reduction algorithms on the preprepared dataset and compare their performance in classification. Of course, according to the table related to the feature reduction step, this issue has been presented in a few papers. These papers have expressed the classification results for different feature reduction algorithms in terms of accuracy, sensitivity, specificity, recall, precision, and so on, and compared the performance of these algorithms. From reviewing all of these papers, it can be understood that the feature reduction algorithm should increase the recognition accuracy and reduce the computational complexity [156], [159]. Thus, the best feature reduction algorithm is an algorithm that reduces the computational load while increasing the classification accuracy. Also, the algorithm must be compatible with the type and complexity of the data; these factors are especially useful in choosing the linearity or nonlinearity of the algorithm. These are the main factors for comparing the performance of feature reduction algorithms. It is recommended to familiarize yourself with the dos and don'ts of your movement classification problem and then choose the best feature selection algorithm by examining the advantages and disadvantages of the algorithms. It is stated in the paper [156] that the most feature reduction methods used to combine with the machine learning classifiers are the unsupervised feature reduction methods (e.g., PCA) and DA feature reduction methods (e.g., LDA). However, there are some limitations in using the mentioned feature reduction methods to deal with classification problems; for example, the eigenvectors extracted by PCA are not robust to variations in the durations of subjects' activities, and only most C – 1 (number of classes minus one) features can be produced by DA feature reduction methods. That is, the DA feature reduction has a poor performance on high-dimensional classification problems. However, PCA is a suitable algorithm from the point of view of computational load, and LDA is generally considered an easy algorithm. To get to know more about the feature reduction algorithms, we try to analyze the rest of the feature reduction algorithms available in the feature reduction section. The main feature reduction algorithms are explained in the feature extraction section. CPCA stands for Common PCA [156]. CPCA is a generalization of ordinary PCA. The latter works only on one group or dataset, but CPCA applies to several datasets or groups. The nonparametric weighted feature extraction (NWFE) is a feature extraction or feature reduction method used to assign every sample with different weights and to define nonparametric between-class and within-class scatter matrices for finding a linear transformation that can maximize the nonparametric between-class scatter and minimize the nonparametric within-class scatter [156]. As we said before, the main disadvantage of the DA feature extraction is that only most C – 1 (number of classes minus one) features can be extracted. In order to solve the abovementioned problem, NWFE is developed for obtaining more than (C – 1) features to deal with high-dimensional classification problems [156]. Kernel PCA (KPCA) and KDA are nonlinear counterparts of PCA and LDA, respectively. These algorithms are extensions of mentioned algorithms based on kernel techniques. In the paper [156], the combined feature extraction methods are used, which are PCA + LDA, NWFE + PCA, and NWFE + LDA. The authors have compared the recognition performances between the six feature reduction methods, such as PCA, LDA, NWFE, PCA + LDA, NWFE + PCA, and NWFE + LDA, once the optimal dimensions of each of the feature reduction schemes were estimated. Algorithms have also been examined from the point of view of computational time. In the paper [60], 1-D local binary patterns (1-D-LBPs) were employed in order to exact relevant features. 1-D-LBP was based on LBPs. In 1-D LBP, all values in the 1-D signal are compared with their neighbors and the histograms of the results of the comparisons or the statistical features of extracted histograms. Locality-preserving projections (LPPs) are linear projective maps that arise by solving a variational problem that optimally preserves the neighborhood structure of the dataset. LPP should be seen as an alternative to PCA [183]. Since LPP is derived by preserving local information, it is less sensitive to outliers than PCA [183]. Canonical correlation analysis (CCA) summarizes the data correlation into fewer statistics while preserving the main aspects of the relationships. The motivation for CCA is very similar to PCA; however, in the latter, the next new variable represents the maximum variance in the individual datasets. On the other hand, in CCA, the new variable is identical for both sets of data such that the correlation between the two resulting new variables is maximized [159]. MRMI-SIG is an optimal data class separator that can be used as a linear feature reduction algorithm. The method uses a nonparametric estimation of Renyi's entropy for feature reduction by maximizing an approximation of the mutual information between the class labels and the reduced features [184]. In the classification step, there are many challenges, such as the null class problem (which is the presence of various activities that do not belong to the set of desired activities) [185], class imbalance (it happens when there is an unequal distribution of classes in the training data), interclass similarity, intraclass variability, overfitting, underfitting, and computational complexity that must be addressed seriously. Interclass similarity is a challenge caused by classes that are fundamentally different, but that shows very similar characteristics in the sensor data [51]. Intraclass variability occurs when the same activity may be performed differently by different individuals [51]. One of the main challenges in the discussion of classification is analyzing and comparing the performance of classification algorithms. Classical algorithms for classification are generally simpler than machine learning algorithms but perform weaker, and according to statistical results, they are not comparable to machine learning algorithms in terms of usage. Specific disadvantages can also be found for these algorithms. We present some of the disadvantages of the most used classical algorithms. When threshold-based algorithms are used for multiclass problems, it will be very difficult to find threshold values. For correlation-based algorithms, it should also be stated that, in general, the correlation-based search cannot provide information about why the relationship is found. Thus, to investigate this challenge, it is better to first compare different types of machine learning algorithms, and finally, we declare a general rule for comparing the overall performance of all classification algorithms. We try to express the main advantages and disadvantages of each machine learning algorithm type. As we stated, the most widely used type of classification algorithm is the supervised algorithm. In general, it is useful to know the advantages and disadvantages of these algorithms. One of the advantages of these algorithms is that we can choose the labels carefully, and as a result, we can easily determine the number of classes. Considering that we know the data well along with their labels, we can say that these algorithms are usually more accurate, especially compared to unsupervised algorithms. These algorithms also have disadvantages. The main disadvantage of these algorithms is ground-truth annotation [54]. Ground-truth annotation is an expensive and tedious task, as the annotator has to do the annotation in real time or skim through the raw sensor data and manually label all instances. Although motion data recorded from motion sensors, such as an accelerometer or gyroscope, are often more difficult to interpret than data from other sensors, such as cameras [54], in daily life settings, ground-truth annotation can be much more difficult [54]. In addition, more computation time is needed for training. Unsupervised methods generally do not have the problem of labeling and can be used when labeled data are scarce or not available. However, these algorithms are less accurate than the previous algorithms. In these algorithms, it is not possible to accurately comment on the relationship between input and output. Also, the number of classes is not known in advance, which creates some kind of confusion. Probabilistic machine learning algorithms can be supervised or unsupervised. The naïve Bayes algorithm is a supervised probabilistic algorithm, and GMM is an unsupervised probabilistic algorithm. The main advantage of these algorithms is that they express uncertainty, while other algorithms are unable to do so. Their main disadvantage is that, due to their probabilistic nature, they require many assumptions that may not always be true. The main advantage of rule-based classification algorithms is that they are easily interpreted due to being close to human logic. However, providing a list of related rules is very difficult and requires experience and skill. Reinforcement learning algorithms do not require labeled data. This is one of their advantages over supervised algorithms [54]. These algorithms are used to solve more complex classification problems, for example, finding the best structure for a neural network. However, reinforcement learning algorithms often have high computational complexity. Combined algorithms are kind of the future of machine learning algorithms because, by adding the capabilities of one classifier to another classifier, many of the flaws and disadvantages of other types of classification algorithms can be avoided. For example, the combination of different algorithms leads to the production of semisupervised algorithms that solve the problem of supervised algorithms and perform classification well with a few labeled data [54]. However, the difficulty here is that we must know the structure of the algorithms that we want to combine, choose compatible algorithms, and know what defect of each algorithm, which we want to solve by combining algorithms. This work requires expertise and time, and may require trial and error. So far, we have announced some factors that must be considered for choosing a classification algorithm and even comparing the performance of different classification algorithms. Now, we are trying to declare a law that makes it possible to analyze the performance of classification in general. For having a good classification performance, a classification algorithm must be robust to the effective factors in classification; these factors are numerous and can be mentioned, such as class imbalance [153], noise in the data, and different distributions of train and test data [26]; robustness increases the generalization ability of the classifier; and a robust algorithm can achieve higher accuracy. Now, we will discuss the challenges related to the concepts of the evaluation section. First, we will examine what criteria are important for choosing the evaluation method. In the paper [103], 5-foldCV and leave-one-participant-out cross-validation (LOPOCV) have been considered as the two evaluation methods widely used for recognition. The authors have announced that they focused on the LOPOCV evaluation results because, usually, it is more difficult to obtain good recognition results when the subject's signals are not involved in the training set. The recognition accuracy of these two algorithms has been considered, and they have announced that, according to the results, LOPOCV is a more suitable evaluation method for their considered application. The paper [120] also mentions the problem of overfitting the classifier, which should be considered when choosing the evaluation method; a good evaluation method can help us avoid this problem. Tahafchi and Judy [124] have stated that all the algorithms used in the motion recognition steps, including cross-validation algorithms, must be accurate and reliable. They also stated that one of the criteria for choosing cross-validation algorithms is that they should work well with imbalanced data. They have stated that stratified-K-fold cross-validation is helpful for imbalanced datasets. In the paper [151], classifiers were trained and tested using two protocols (user-specific training protocol and leave-one-subject-out validation). Recognition accuracy was significantly higher for all algorithms under the leave-one-subject-out validation process. Because of larger training sets, this protocol may have resulted in more generalized and robust activity classifiers. The markedly smaller training sets used for the user-specific training protocol may have limited the accuracy of classifiers. Another issue that is important is the computational load; for example, if we have two algorithms that have almost the same performance in terms of accuracy, recall, precision, recall, and so on, then the algorithm with less computational load should be selected. Now, we are trying to select the criteria for choosing the best evaluation method from the overview of the papers presented above and the papers in Table XIII. The selection criteria of evaluation methods are low computational load, robust and generalizable recognition performance, dealing with overfitting of the classifier, suitable performance on imbalanced data, and so on. The next challenge is to choose the desired metric to evaluate the performance of the classifier. As we mentioned, the evaluation metrics are categorized into three different types: threshold, probabilistic, and ranking metrics. Graphical metrics, such as confusion matrix and ROC, have also been examined in the evaluation section, but these metrics can be considered as methods for obtaining the metrics. Overall comparison of metrics makes it easy to choose between these items. It is not bad to first have a practical comparison between three types of evaluation metrics and introduce useful metrics in each of the applications. All these types of metrics are scalar and present the performance using a single score value [153]. These types of metrics are mostly used in three different evaluation applications [153]. First, the evaluation metrics are used for evaluating the generalization ability of the trained classifier. Second, the evaluation metrics are used to select the best classifier among different types of classifiers. Third, the evaluation metrics are employed to discriminate and select the best solution among all generated solutions during training [153]. In the first and second applications, all three types of metrics can be used [153]. However, only a few types of metrics can be employed in the third one [153]. The third application is less common in motion recognition. For more familiarity with the metrics that can be used in the third application, refer to the paper [153]. The paper [153] has offered factors for the construction of new metrics for use in the third application. These factors, more or less, can be used even to compare and select metrics in all three applications. Before dealing with these factors, we intend to analyze the performance of some most frequently used metrics. Before starting, we should point out that most of the evaluation metrics are made for binary classification, and through modifications, they can be extended to multiclass mode as well. Through accuracy, the classification quality is evaluated based on the percentage of correct predictions over total instances. The error rate is the complement metric of accuracy and evaluates the classification by its percentage of incorrect predictions [153]. Sensitivity measures the fraction of positive patterns that are correctly classified [153]. Specificity does the same for negative patterns. Precision measures the correctly predicted positive patterns rate to total predicted patterns in a positive class [153]. Recall measures the rate of correctly predicted positive patterns. The f score is the harmonic mean between recall and precision values [153]. The declared cases were the most famous threshold metrics. We also introduce some probabilistic methods. The mse is a measure of the difference between the predicted solutions and actual solutions. The smaller the mse value, the better the classification results [153]. RMSE is the square root of the mse. The area under the ROC curve, known as the AUC, is also one of the most famous examples of the ranking type of evaluation metrics. Unlike the other two types of metrics, this value shows the overall ranking performance of a classifier [153]. Now that we are familiar with a few evaluation metrics of the classification performance, we cite the advantages and disadvantages of some metrics, and then, we will provide the general reasons that can be used to choose the desired metric. According to the paper [153], accuracy and error rate are easy to compute; applicable for multiclass and multilabel problems; and easy to understand by humans. This article also states that accuracy has many weaknesses. For example, accuracy is not a good metric when dealing with imbalanced class distribution and is biased toward majority class data. Another disadvantage of accuracy is that this metric produces less distinctive and less discriminable values. The mse is also not suitable for working with imbalanced class data. The AUC is proven to be better than the accuracy metric for evaluating the classifier performance, but it has a very high computational cost [153]. Factors can now be introduced for the comparison of existing metrics. It is better to choose a metric that can be used in multiclass problems and is not limited to binary classification. It is better to choose a metric that has a lower computational load. A good metric should not be biased toward the majority class and must work well on imbalanced data. Of course, the factors raised are largely public factors. It is not bad to look at the matter a little more technically. Another challenge is that learning methods that perform well on one metric may not perform well on other metrics; for example, SVM classifiers optimize accuracy, while neural networks optimize probabilistic metrics, such as RMSE and cross-entropy [154]. Therefore, after choosing one or more classifiers for our work according to the stated criteria, in addition to considering the above factors, it is better to get familiar with what metric these classifiers optimize best and choose that metric to evaluate the classifier. Of course, it is not bad to provide a general answer to the question that, in general, if we do not know the correct evaluation metric, which metric should we use by default? The paper [154] generally stated that RMSE might serve as a good general-purpose metric to use when a more specific optimization criterion is not known. Now with these factors in mind, one can choose a suitable metric for the classification problem. Now, we want to discuss the data communication model, power efficiency, and propagation delay since these concepts are very relevant for movement classification using wearable IoT. Familiarity with these concepts can also solve many existing challenges. Before dealing with the topics, we will talk about body sensor networks (BSNs) that are an inseparable part of wearable IoT. BSN is a set of sensors connected to the body that together forms a network and collects the necessary information. The BSN used in this field is usually wireless and can be considered a type of wireless sensor network (WSN). BSN is an important component of the IoT [186]. First, we want to specify the data communication model in BSN. The general architecture of a BSN consists of sensor nodes that are placed in the body to collect data and perform preliminary processing. The data are gathered by a sink node and then transmitted to a base station to share over the Internet [186]. This method of data communication with a slight modification is also presented in the paper [187]. However, the entire structure has the same skeleton. Sensors are the key components of BSN, as they connect the physical world with electronic systems. They are mainly used to collect information about the human body. Sensor nodes, which have a sensor as their main part, are responsible for processing information by format conversion, logical computing, data storage, and transmitting. One sensor node generally comprises a sensor module, a processor module, a wireless communication module, and a power supply module. The sensor module is responsible for collecting the status of measurements and converting data to electrical signals. The processor module is responsible for controlling the sensor nodes. The wireless communication module, consisting of the network layer, the MAC layer, and the wireless transceiver in the physical layer, is responsible for communication among sensors and computers. The power supply module is responsible for providing energy for the entire sensor node [186]. Nowadays, BSN research still faces many key technical challenges, such as energy consumption and service quality [186], [187]. Energy consumption and power efficiency are among the most important challenges in these networks, which, of course, was briefly mentioned at the beginning of this section, but, now, we intend to look at the issue a little more generally. BSNs can be battery-powered. They can also be powered by kinetic energy and heat [186]. Our energy resources are limited, so we try to explain the methods of reducing energy consumption and, thus, improving power efficiency. First, we start energy consumption reduction concerning BSN sensors. Low-power design is one of the main challenges for sensors in this network. Familiarity with the classification of sensors can be useful in reducing their energy consumption. According to the types of measured signals, sensors in BSNs can be divided into two categories. The first category that collects signals continuously includes accelerometers, gyroscopes, ECG sensors, EEG sensors, EMG sensors, and so on. The second category, including temperature sensors, humidity sensors, and so on, collects discrete-time signals. Usually, the first category's power consumption is more than the second one. Therefore, it is better to choose the second type of sensor among sensors that may measure similar signals for movement classification (of course, if energy consumption is the priority). Another possible way to reduce energy consumption is using the sleeping mode [186], [187]. The most commonly used sensors in BSNs can be divided into the following three categories according to the types of data transmission media: wireless sensors, which employ wireless communication technologies, such as Bluetooth or Zigbee, and radio frequency identification devices (RFIDs), to communicate with other sensors or devices. Wired sensors, employing wired communication technologies, can replace wireless sensors if wearability is not seriously affected. The transmission mode is more stable than the wireless one. However, their installation and deployment are complicated. The third category is human-body communication (HBC) sensors that use the human body as the transmission medium [186]. The latter can have lower power consumption and sensor node size than the first two, but it has been introduced in recent years and needs more time to settle [186]. In the design of sensor nodes, issues related to reducing power consumption can also be considered. In the sensor node design process, energy control and reduction of sensor nodes can be considered to meet the demands of low power consumption [186]. Energy control has been one of the hot topics in the field of BSN sensors for the implementation of long-term monitoring functions. The low-power architecture design, the low-power processor design, the low-power transceiver design, and the energy acquisition design are preliminary research topics in energy control at present [186]. The goal of reducing sensor nodes is inertial sensors for activity recognition. It not only improves the wearability of the mentioned systems but also lowers the cost, saves energy, and so on. Principal methods to solve the problem are node placement optimization and the improvement of activity recognition algorithms [186]. In the paper [186], it is also stated that data fusion techniques can reduce data redundancy and, thus, reduce the load and energy consumption of BSN with the advantage of extending the network lifetime. In the BSN communication section, the factors that can be addressed to improve energy efficiency and reduce power consumption include proper network topology, energy-efficient MAC and routing protocols, and so on [186], [187]. At the beginning of the discussions, we announced that service quality is one of the most challenging topics in BSN. In BSNs, this concept is known as Quality of Service (QoS). QoS generally can be considered as a description of overall network performance. QoS can be characterized by packet loss possibility, available bandwidth, end-to-end delay, jitter, and so on. Examining other factors related to QoS, such as jitter, available bandwidth, and packet loss, is not on the agenda of this article, and addressing them is not related to the main topic of our paper and will take us away from the main goal, so they will not be addressed. The specialized investigation of these factors is in the field of telecommunication engineering. Therefore, we are going to briefly discuss the end-to-end delay. In general, the end-to-end delay in the BSN is divided into four types: propagation delay, transmission delay, queuing delay, and processing delay. The time that it takes for the data to be transmitted from the source to the destination is called propagation delay. The time that it takes for the data to be completely transmitted is called transmission delay. The time that data must wait in the buffer until the busy destination can check it is called queuing delay. The time that it takes a processor to process data is called the processing delay. These four delays make up the end-to-end delay. It is better to know the causes of each of these delays. General factors that cause propagation delay include the characteristics of the medium and environmental characteristics, such as humidity, pressure, temperature, signal disturbances, and so on. However, we will try to provide some more specialized examples. Propagation delay in electronic circuits or logic gates is one of the most obvious examples of this delay in BSN [186]. Addressing the problem of propagation delay in electrical circuits involved in BSN, such as logic circuits and SRAM in microcontrollers, is one of the main concerns of electrical circuit designers [186]. In paper [188], the end-to-end delay was considered to include four types of delays: transmission delay, queuing delay, processing delay, and channel capture delay. In this article, channel capture delay is the same as propagation delay. This phenomenon occurs when a device from a shared medium takes possession of the media for a significant period. In the mentioned paper, the authors have presented a relay-based routing protocol for in vivo BSNs. The proposed protocol is provided with linear programming-based mathematical models for network lifetime maximization and end-to-end delay minimization [188]. Therefore, we must know the various sources of propagation delay in BSN and find a suitable solution for each of them. Now, we also announce the general factors causing other delays. Factors such as transmission speed and bandwidth are effective in causing transmission delay. Factors such as bandwidth, data volume, and the type of queuing method are also effective in causing queuing delays. The features of the processing device, the volume of data, and the complexity of the processing algorithms are also factors that cause the processing delay. Now that we are familiar with the technical challenges in movement classification, we must mention that recognizing complex activities (such as cooking and doing the dishes) is also a technical challenge that is beyond the scope of this article. Finding a solution for this challenge will also be a very suitable topic for future papers.

VI. Conclusion

In this article, we announced that wearable IoTs will be widely used in the future. Human motion recognition by wearable sensors is investigated in this article. Since classification is an integral part of human body motion recognition, it can be claimed that movement classification is closely related to human motion recognition. Movement classification includes three subsections: GA, GR, and HAR. The goal is to first introduce the reader to the important steps of human body movement classification by wearable sensors and then determine the algorithms and methods used for each step using tables. To better understand the results of the tables, approximate numbers and percentages have been used. In some cases, bar charts have been used to visualize numerical results. By reading this article, the readers will be fully acquainted with the concepts in movement classification, know the steps of conducting research along with commonly used algorithms, wearable sensors, IoT concepts, and future directions, and can carry out the project in the human motion recognition area.

Appendix

See Table XVII.

References

[1] E. O. Thorp, *Beat the Dealer: A Winning Strategy for the Game of Twenty-One*. New York, NY, USA: Vintage Books, 1966.

[2] F. J. Dian and R. Vahidnia, "IoT use cases," in *IoT Use Cases and Technologies*, Vancouver, BC, Canada: BCIT, 2020, pp. 1–23.

[3] F. John Dian, R. Vahidnia, and A. Rahmati, "Wearables and the Internet of Things (IoT), applications, opportunities, and challenges: A survey," *IEEE Access*, vol. 8, pp. 69200–69211, 2020, doi: 10.1109/ACCESS.2020.2986329.

[4] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, "Reviews on various inertial measurement unit (IMU) sensor applications," *Int. J. Signal Process. Syst.*, vol. 10, pp. 256–262, Jan. 2013, doi: 10.12720/ijsps.1.2.256-262.

[5] S. T. Pheasant, "A review of: "*Human walking*". By V. T. INMAN, H. J. RALSTON and F. TODD. (Baltimore, London: Williams & Wilkins, 1981.) [Pp.154.]," *Ergonomics*, vol. 24, no. 12, pp. 969–976, Dec. 1981, doi: 10.1080/00140138108924919.

[6] I. H. López-Nava and A. Muñoz-Meléndez, "Wearable inertial sensors for human motion analysis: A review," *IEEE Sensors J.*, vol. 16, no. 22, pp. 7821–7834, Nov. 2016, doi: 10.1109/JSEN.2016.2609392.

[7] K. L. Moore, A. F. Dalley, and A. M. Agur, *Clinically Oriented Anatomy*. Philadelphia, PA, USA: Wolters Kluwer Health, 2011.

[8] N. Alshurafa et al., "Designing a robust activity recognition framework for health and exergaming using wearable sensors," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 5, pp. 1636–1646, Sep. 2014, doi: 10.1109/JBHI.2013.2287504.

[9] A. Jordao, A. C. Nazare Jr., J. Sena, and W. R. Schwartz, "Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art," 2018, *arXiv:1806.05226*.

[10] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. Syst., Man Cybern., C, Appl. Rev.*, vol. 37, no. 3, pp. 311–324, May 2007, doi: 10.1109/TSMCC.2007.893280.

[11] A. N. Belbachir et al., "CARE: Dynamic stereo vision sensor system for fall detection," Presented at the IEEE Int. Symp. Circuits Syst. (ISCAS), Seoul, South Korea, May 2012.

[12] P. Pierleoni et al., "A wearable fall detector for elderly people based on AHRS and barometric sensor," *IEEE Sensors J.*, vol. 16, no. 17, pp. 6733–6744, Sep. 2016, doi: 10.1109/JSEN.2016.2585667.

[13] M. P. Lawton and M. E. Brody, "Assessment of older people self maintaining and instrumental activities of daily living," *Gerontologist*, vol. 9, no. 3, pp. 179–186, 1969.

[14] C. Jobanputra, J. Bavishi, and N. Doshi, "Human activity recognition: A survey," *Proc. Comput. Sci.*, vol. 155, pp. 698–703, Jan. 2019, doi: 10.1016/j.procs.2019.08.100.

[15] A. Das and M. B. Kjærgaard, "Activity recognition using multi-class classification inside an educational building," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Austin, TX, USA, Mar. 2020, pp. 1–6.

[16] A. Mann and M.-F. Chesselet, "Techniques for motor assessment in rodents," in *Movement Disorders*, 2nd ed. Los Angeles, CA, USA: AP, 2015, ch. 8, pp. 139–157.

[17] T. Allevard, E. Benoit, and L. Foulloy, "Hand posture recognition with the fuzzy glove," Presented at the Mod. Inf. Process., Theory Appl., 2006.

[18] B. H. Dobkin, "Wearable motion sensors to continuously measure real-world physical activities," *Current Opinion Neurol.*, vol. 26, no. 6, pp. 602–608, Dec. 2013, doi: 10.1097/WCO.0000000000000026.

[19] S. Zhang et al., "Deep learning in human activity recognition with wearable sensors: A review on advances," *Sensors*, vol. 22, no. 4, p. 1476, Feb. 2022, doi: 10.3390/s22041476.

[20] W. Tao, T. Liu, R. Zheng, and H. Feng, "Gait analysis using wearable sensors," *Sensors*, vol. 12, no. 2, pp. 2255–2283, Feb. 2012, doi: 10.3390/s120202255.

[21] L. Dai, "C_{60} and carbon nanotube sensors," in *Carbon Nanotechnology*. Amsterdam, The Netherlands: Elsevier, 2006, ch. 15, pp. 525–575.

[22] N. Bhalla, P. Jolly, N. Formisano, and P. Estrela, "Introduction to biosensors," *Essays Biochem.*, vol. 60, no. 1, pp. 1–8, Jun. 2016, doi: 10.1042/EBC20150001.

[23] A. Nag et al., "Graphene-based wearable temperature sensors: A review," *Mater. Des.*, vol. 221, pp. 1–17, Jan. 2022, doi: 10.1016/j.matdes.2022.110971.

[24] H.-S. Kim, J.-H. Kang, J.-Y. Hwang, and U. S. Shin, "Wearable CNTs-based humidity sensors with high sensitivity and flexibility for real-time multiple respiratory monitoring," *Nano Converg.*, vol. 9, no. 1, pp. 1–10, Aug. 2022, doi: 10.1186/s40580-022-00326-6.

[25] S. K. Ameri et al., "Imperceptible electrooculography graphene sensor system for human–robot interface," *NPJ 2D Mater. Appl.*, vol. 2, no. 1, pp. 1–7, Jul. 2018, doi: 10.1038/s41699-018-0064-4.

[26] H. Wang, W. Yan, and S. Liu, "Physical activity recognition using multi-sensor fusion and extreme learning machines," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Budapest, Hungary, Jul. 2019, pp. 1–7.

[27] S. Liu, R. X. Gao, D. John, J. W. Staudenmayer, and P. S. Freedson, "Multisensor data fusion for physical activity assessment," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 3, pp. 687–696, Mar. 2012, doi: 10.1109/TBME.2011.2178070.

[28] L. Utari et al., "Wearable carbon monoxide sensors based on hybrid graphene/ZnO nanocomposites," *IEEE Access*, vol. 8, pp. 49169–49179, 2020, doi: 10.1109/ACCESS.2020.2976841.

[29] M. Sharma, S. Kacker, and M. Sharma, "A brief introduction and review on galvanic skin response," *Int. J. Med. Res. Professionals*, vol. 2, no. 6, pp. 13–17, Dec. 2016, doi: 10.21276/ijmrp.2016.2.6.003.

[30] P. K. Baheti and H. Garudadri, "An ultra low power pulse oximeter sensor based on compressed sensing," in *Proc. 6th Int. Workshop Wearable Implant. Body Sensor Netw.*, Berkeley, CA, USA, Jun. 2009, pp. 1–12.

[31] A. M. A. Handojoseno, J. M. Shine, T. N. Nguyen, Y. Tran, S. J. G. Lewis, and H. T. Nguyen, "Analysis and prediction of the freezing of gait using EEG brain dynamics," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 5, pp. 887–896, Sep. 2015, doi: 10.1109/TNSRE.2014.2381254.

[32] R. E. Morley, E. J. Richter, J. W. Klaesner, K. S. Maluf, and M. J. Mueller, "In-shoe multisensory data acquisition system," *IEEE Trans. Biomed. Eng.*, vol. 48, no. 7, pp. 815–820, Jul. 2001, doi: 10.1109/10.930906.

[33] R. H. Roberts and Y. L. Mo, "Development of carbon nanofiber aggregate for concrete strain monitoring," in *Innovative Developments of Advanced Multifunctional Nanocomposites in Civil and Structural Engineering*, 1st ed. Swaston, U.K.: Woodhead Publishing, 2016, ch. 2, pp. 9–45.

[34] J. Cong, J. Jing, and C. Chen, "Development of a PVDF sensor array for measurement of the dynamic pressure field of the blade tip in an axial flow compressor," *Sensors*, vol. 19, no. 6, p. 1404, Mar. 2019, doi: 10.3390/s19061404.

[35] J. Lester, T. Choudhury, N. Kern, and G. Borriello, "A hybrid discriminative/generative approach for modeling human activities," Presented at the 19th Int. Joint Conf. Artif. Intell., Jul. 2005.

[36] A. Pentland, "Healthwear: Medical technology becomes wearable," *Computer*, vol. 37, no. 5, pp. 42–49, May 2004, doi: 10.1109/MC.2004.1297238.

[37] A. Farooq and S. Kamal, "Indoor positioning and tracking using sensors of a smart device," in *Proc. Int. Conf. Appl. Eng. Math.*, Taxila, Pakistan, Aug. 2019, pp. 99–104.

[38] Q. P. Chu and P. L. M. T. Van Woerkom, "GPS for low-cost attitude determination: A review of concepts, in-flight experiences, and current developments," *Acta Astronaut.*, vol. 41, nos. 4–10, pp. 421–433, Aug. 1997, doi: 10.1016/S0094-5765(98)00046-0.

[39] K. E. Jeon, J. She, P. Soonsawad, and P. C. Ng, "BLE beacons for Internet of Things applications: Survey, challenges, and opportunities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 811–828, Apr. 2018, doi: 10.1109/JIOT.2017.2788449.

[40] D. De, P. Bharti, S. K. Das, and S. Chellappan, "Multimodal wearable sensing for fine-grained activity recognition in healthcare," *IEEE Internet Comput.*, vol. 19, no. 5, pp. 26–35, Sep. 2015, doi: 10.1109/MIC.2015.72.

[41] M. Baxter and N. E. O'Connell, "Testing ultra-wideband technology as a method of tracking fast-growing broilers under commercial conditions," *Appl. Animal Behav. Sci.*, vol. 233, Dec. 2020, Art. no. 105150, doi: 10.1016/j.applanim.2020.105150.

[42] J. Ma, W. Gao, J. Wu, and C. Wang, "A continuous Chinese sign language recognition system," in *Proc. 4th IEEE Int. Conf. Autom. Face Gesture Recognit.*, Grenoble, France, Mar. 2000, pp. 1–3.

[43] L. Rodrigues and M. Mota, "Smart devices: Micro- and nano sensors," in *Bioinspired Materials for Medical Applications*, 1st ed. Swaston, U.K.: Woodhead Publishing, 2017, ch. 11, pp. 297–329.

[44] P. Kosky, R. Balmer, W. Keat, and G. Wise, *Mechatronics and Physical Computing, Exploring Engineering*, 5th ed. Cambridge, MA, USA: AP, 2021, ch. 20, pp. 453–477.

[45] M. Borghetti, E. Sardini, and M. Serpelloni, "Evaluation of bend sensors for limb motion monitoring," in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Lisboa, Portugal, Jun. 2014, pp. 1–5.

[46] G. Ogris, T. Stiefmeier, H. Junker, P. Lukowicz, and G. Troster, "Using ultrasonic hand tracking to augment motion analysis based recognition of manipulative gestures," in *Proc. 9th IEEE Int. Symp. Wearable Comput.*, Osaka, Japan, Mar. 2005, pp. 1–5.

[47] K. Van Laerhoven and O. Cakmakci, "What shall we teach our pants?" in *Dig. Papers, 4th Int. Symp. Wearable Comput.*, Atlanta, GA, USA, 2000, pp. 1–14.

[48] S. Hackett, Y. Cai, and M. Siegel, "Activity recognition from sensor fusion on fireman's helmet," Presented at the 12th Int. Congr. Image Signal Process., BioMed. Eng. Inform. (CISP-BMEI), Suzhou, China, Oct. 2019.

[49] S. Ruffieux, D. Lalanne, E. Mugellini, and O. A. Khaled, "A survey of datasets for human gesture recognition," Presented at the 16th Int. Conf., HCI Int., Heraklion Greece, Jun. 2014.

[50] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–40, May 2022, doi: 10.1145/3447744.

[51] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, pp. 1–33, Jan. 2014, doi: 10.1145/2499621.

[52] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tr, "Wearable activity tracking in car manufacturing," *IEEE Pervasive Comput.*, vol. 7, no. 2, pp. 42–50, Apr. 2008, doi: 10.1109/MPRV.2008.40.

[53] T. Stiefmeier, G. Ogris, H. Junker, P. Lukowicz, and G. Troster, "Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario," in *Proc. 10th IEEE Int. Symp. Wearable Comput.*, Montreux, Switzerland, Oct. 2006, pp. 1–10.

[54] U. Blanke, B. Schiele, M. Kreil, P. Lukowicz, B. Sick, and T. Gruber, "All for one or one for all? Combining heterogeneous features for activity spotting," in *Proc. 8th IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Mannheim, Germany, Mar. 2010, pp. 1–4.

[55] H. Nematallah and S. Rajan, "Comparative study of time series-based human activity recognition using convolutional neural networks," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, Dubrovnik, Croatia, May 2020, pp. 1–6.

[56] A. Arami, A. Poulakakis-Daktylidis, Y. F. Tai, and E. Burdet, "Prediction of gait freezing in parkinsonian patients: A binary classification augmented with time series prediction," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 9, pp. 1909–1919, Sep. 2019, doi: 10.1109/TNSRE.2019.2933626.

[57] T. Mahmud, S. S. Akash, S. A. Fattah, W.-P. Zhu, and M. O. Ahmad, "Human activity recognition from multi-modal wearable sensor data using deep multi-stage LSTM architecture based on temporal feature aggregation," in *Proc. IEEE 63rd Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Springfield, MA, USA, Aug. 2020, pp. 1–3.

[58] S. V. Perumal and R. Sankar, "Gait monitoring system for patients with Parkinson's disease using wearable sensors," in *Proc. IEEE Healthcare Innov. Point-Care Technol. Conf. (HI-POCT)*, Cancun, Mexico, Nov. 2016, pp. 21–24.

[59] A. K. Chowdhury, D. Tjondronegoro, V. Chandran, and S. G. Trost, "Physical activity recognition using posterior-adapted class-based fusion of multiaccelerometer data," *IEEE J. Biomed. Health Informat.*, vol. 22, no. 3, pp. 678–685, May 2018, doi: 10.1109/JBHI.2017.2705036.

[60] N. Inanç, M. Kayri, and Ö. F. Ertugrul, "Recognition of daily and sports activities," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Seattle, WA, USA, Dec. 2018, pp. 2216–2220.

[61] N. Kawaguchi et al., "HASC challenge: Gathering large scale human activity corpus for the real-world activity understandings," in *Proc. 2nd Augmented Human Int. Conf.*, Tokyo, Japan, Mar. 2011, pp. 1–5.

[62] M. Inoue, S. Inoue, and T. Nishida, "Deep recurrent neural network for mobile human activity recognition with high throughput," *Artif. Life Robot.*, vol. 23, no. 2, pp. 173–185, Jun. 2018, doi: 10.1007/s10015-017-0422-x.

[63] L. Chen, Y. Li, and Y. Liu, "Human body gesture recognition method based on deep learning," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Hefei, China, Aug. 2020, pp. 587–591.

[64] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 1307–1310.

[65] M. Zhang and A. A. Sawchuk, "USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proc. ACM Conf. Ubiquitous Comput.*, Sep. 2012, pp. 1036–1043.

[66] M. Shoaib, S. Bosch, O. Incel, H. Scholten, and P. Havinga, "Fusion of smartphone motion sensors for physical activity recognition," *Sensors*, vol. 14, no. 6, pp. 10146–10176, Jun. 2014, doi: 10.3390/s140610146.

[67] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, pp. 1–25, Jan. 2016, doi: 10.3390/s16010115.

[68] P. Zappi et al., "Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection," Presented at the 5th Eur. Conf. Wireless Sensor Netw., Bologna, Italy, Jan. 2008.

[69] J. W. Lockhart, G. M. Weiss, J. C. Xue, S. T. Gallagher, A. B. Grosner, and T. T. Pulickal, "Design considerations for the WISDM smart phone-based sensor mining architecture," in *Proc. 5th Int. Workshop Knowl. Discovery Sensor Data*, San Diego, CA, USA, Aug. 2011, pp. 1–4.

[70] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," in *Proc. 10th Int. Conf. Ubiquitous Comput.*, Seoul, South Korea, Sep. 2008, pp. 10–19.

[71] U. Blanke and B. Schiele, "Daily routine recognition through activity spotting," Presented at the Location- Context-Awareness, 3rd Int. Symp., Oberpfaffenhofen, Germany, Sep. 2007.

[72] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," in *Proc. 10th Int. Conf. Ubiquitous Comput.*, Sep. 2008, pp. 1–9.

[73] U. Blanke and B. Schiele, "Remember and transfer what you have learned–recognizing composite activities based on activity spotting," in *Proc. Int. Symp. Wearable Comput. (ISWC)*, Oct. 2010, pp. 1–8.

[74] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in *Proc. Joint Conf. Smart Objects Ambient Intell., Innov. Context-Aware Service Usages Technol.*, Grenoble, France, Oct. 2005, pp. 159–163.

[75] F. Torre, J. Hodgins, J. Montano, S. Valcarcel, R. Forcada, and J. Macey, "Guide to the Carnegie Mellon university multimodal activity (CMU-MMAC) database," CMU, Pittsburgh, PA, USA, Tech. Rep., CMU-RI-TR-08-22, 2008.

[76] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, Mar. 2011, doi: 10.1145/1964897.1964918.

[77] C. Chen, R. Jafari, and N. Kehtarnavaz, "UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Quebec City, QC, Canada, Sep. 2015, pp. 168–172.

[78] A. Stisen et al., "Smart devices are different: Assessing and Mitigating-Mobile sensing heterogeneities for activity recognition," in *Proc. 13th ACM Conf. Embedded Networked Sensor Syst.*, New York, NY, USA, Nov. 2015, pp. 127–140.

[79] H. Gjoreski et al., "The university of Sussex-Huawei locomotion and transportation dataset for multimodal analytics with mobile devices," *IEEE Access*, vol. 6, pp. 42592–42604, 2018, doi: 10.1109/ACCESS.2018.2858933.

[80] M. Shoaib, H. Scholten, and P. J. M. Havinga, "Towards physical activity recognition using smartphone sensors," in *Proc. IEEE 10th Int. Conf. Ubiquitous Intell. Comput. IEEE 10th Int. Conf. Autonomic Trusted Comput.*, Dec. 2013, pp. 80–87.

[81] D. Micucci, M. Mobilio, and P. Napoletano, "UniMiB SHAR: A dataset for human activity recognition using acceleration data from smartphones," *Appl. Sci.*, vol. 7, no. 10, pp. 1–19, 2017, doi: 10.3390/app7101101.

[82] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE Pervasive Comput.*, vol. 16, no. 4, pp. 62–74, Oct. 2017, doi: 10.1109/MPRV.2017.3971131.

[83] K. Kyritsis, C. L. Tatli, C. Diou, and A. Delopoulos, "Automated analysis of in meal eating behavior using a commercial wristband IMU sensor," in *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2017, pp. 2843–2846.

[84] B. Bruno, F. Mastrogiovanni, and A. Sgorbissa, "A public domain dataset for ADL recognition using wrist-placed accelerometers," in *Proc. 23rd IEEE Int. Symp. Robot Human Interact. Commun.*, Aug. 2014, pp. 738–743.

[85] M. Bennasar, B. A. Price, D. Gooch, A. K. Bandara, and B. Nuseibeh, "Significant features for human activity recognition using tri-axial accelerometers," *Sensors*, vol. 22, no. 19, p. 7482, Oct. 2022, doi: 10.3390/s22197482.

[86] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "UWave: Accelerometer-based personalized gesture recognition and its applications," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, Galveston, TX, USA, Mar. 2009, pp. 1–6.

[87] R. Jain, V. B. Semwal, and P. Kaushik, "Deep ensemble learning approach for lower extremity activities recognition using wearable sensors," *Expert Syst.*, vol. 39, no. 6, pp. 1–12, Jul. 2022, doi: 10.1111/exsy.12743.

[88] V. B. Semwal, A. Gupta, and P. Lalwani, "An optimized hybrid deep learning model using ensemble learning approach for human walking activities recognition," *J. Supercomput.*, vol. 77, no. 11, pp. 12256–12279, Nov. 2021, doi: 10.1007/s11227-021-03768-7.

[89] V. B. Semwal, N. Gaud, P. Lalwani, V. Bijalwan, and A. K. Alok, "Pattern identification of different human joints for different human walking styles using inertial measurement unit (IMU) sensor," *Artif. Intell. Rev.*, vol. 55, no. 2, pp. 1149–1169, Feb. 2022, doi: 10.1007/s10462-021-09979-x.

[90] M. Raj, V. B. Semwal, and G. C. Nandi, "Bidirectional association of joint angle trajectories for humanoid locomotion: The restricted Boltzmann machine approach," *Neural Comput. Appl.*, vol. 30, no. 6, pp. 1747–1755, Sep. 2018, doi: 10.1007/s00521-016-2744-3.

[91] V. B. Semwal, P. Lalwani, M. K. Mishra, V. Bijalwan, and J. S. Chadha, "An optimized feature selection using bio-geography optimization technique for human walking activities recognition," *Computing*, vol. 103, no. 12, pp. 2893–2914, Dec. 2021, doi: 10.1007/s00607-021-01008-7.

[92] V. B. Semwal, A. Mazumdar, A. Jha, N. Gaud, and V. Bijalwan, "Speed, cloth and pose invariant gait recognition-based person identification," in *Machine Learning: Theoretical Foundations and Practical Applications*. Berlin, Germany: Springer, 2021, pp. 39–56.

[93] V. Tuzlukov, *Signal Processing Noise (Electrical Engineering & Applied Signal Processing Series)*, 1st ed. Boca Raton, FL, USA: CRC Press, 2002.

[94] I. Khokhlov, L. Reznik, J. Cappos, and R. Bhaskar, "Design of activity recognition systems with wearable sensors," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Mar. 2018, pp. 1–6.

[95] G. Cola, M. Avvenuti, A. Vecchio, G. Yang, and B. Lo, "An on-node processing approach for anomaly detection in gait," *IEEE Sensors J.*, vol. 15, no. 11, pp. 6640–6649, Nov. 2015, doi: 10.1109/JSEN.2015.2464774.

[96] T. Le, T. Tran, and C. Pham, "The Internet-of-Things based hand gestures using wearable sensors for human machine interaction," in *Proc. Int. Conf. Multimedia Anal. Pattern Recognit. (MAPR)*, Ho Chi Minh City, Vietnam, May 2019, pp. 1–6.

[97] K. Leuenberger, R. Gonzenbach, E. Wiedmer, A. Luft, and R. Gassert, "Classification of stair ascent and descent in stroke patients," in *Proc. 11th Int. Conf. Wearable Implant. Body Sensor Netw. Workshops*, Zurich, Switzerland, Jun. 2014, pp. 11–16.

[98] Y. Li et al., "Hand gesture recognition and real-time game control based on a wearable band with 6-axis sensors," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–6.

[99] A. Nandy, J. Saha, C. Chowdhury, and K. P. D. Singh, "Detailed human activity recognition using wearable sensor and smartphones," in *Proc. Int. Conf. Opto-Electron. Appl. Opt. (Optronix)*, Kolkata, India, Mar. 2019, pp. 1–6.

[100] Y. Li, N. Yang, L. Li, L. Liu, and Y. Yang, "Finger gesture recognition using a smartwatch with integrated motion sensors," *Web Intell.*, vol. 16, no. 2, pp. 123–129, Jun. 2018, doi: 10.3233/WEB-180378.

[101] H. Li et al., "Hierarchical sensor fusion for micro-gesture recognition with pressure sensor array and radar," *IEEE J. Electromagn., RF Microw. Med. Biol.*, vol. 4, no. 3, pp. 225–232, Sep. 2020, doi: 10.1109/JERM.2019.2949456.

[102] R. Patil, "Noise reduction using wavelet transform and singular vector decomposition," *Proc. Comput. Sci.*, vol. 54, pp. 849–853, Jan. 2015, doi: 10.1016/j.procs.2015.06.099.

[103] T. Pan, W. Tsai, C. Chang, C. Yeh, and M. Hu, "A hierarchical hand gesture recognition framework for sports referee training-based EMG and accelerometer sensors," *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 3172–3183, May 2022, doi: 10.1109/TCYB.2020.3007173.

[104] C. Derr and F. Sahin, "Signer-independent classification of American sign language word signs using surface EMG," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Banff, AB, Canada, Oct. 2017, pp. 665–670.

[105] N. Hegde et al., "The pediatric SmartShoe: Wearable sensor system for ambulatory monitoring of physical activity and gait," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 2, pp. 477–486, Feb. 2018, doi: 10.1109/TNSRE.2017.2786269.

[106] Y. Yu, X. Chen, S. Cao, X. Zhang, and X. Chen, "Exploration of Chinese sign language recognition using wearable sensors based on deep belief net," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 5, pp. 1310–1320, May 2020, doi: 10.1109/JBHI.2019.2941535.

[107] P. Arnon, "Classification model for multi-sensor data fusion apply for human activity recognition," in *Proc. Int. Conf. Comput., Commun., Control Technol. (I4CT)*, Langkawi, Malaysia, Sep. 2014, pp. 415–419.

[108] R. C. Luo, C. C. Chang, and C. C. Lai, "Multisensor fusion and integration: Theories, applications, and its perspectives," *IEEE Sensors J.*, vol. 11, no. 12, pp. 3122–3138, Dec. 2011, doi: 10.1109/JSEN.2011.2166383.

[109] J. A. Ward, P. Lukowicz, G. Troster, and T. E. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1553–1567, Oct. 2006, doi: 10.1109/TPAMI.2006.197.

[110] J. Wannenburg and R. Malekian, "Physical activity recognition from smartphone accelerometer data for user context awareness sensing," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 12, pp. 3142–3149, Dec. 2017, doi: 10.1109/TSMC.2016.2562509.

[111] R. C. King, E. Villeneuve, R. J. White, R. S. Sherratt, W. Holderbaum, and W. S. Harwin, "Application of data fusion techniques and technologies for wearable health monitoring," *Med. Eng. Phys.*, vol. 42, pp. 1–12, Apr. 2017, doi: 10.1016/j.medengphy.2016.12.011.

[112] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *Comput. Inf. Syst.*, vol. 7, no. 1, pp. 1–10, 2000.

[113] H. Azami, B. Bozorgtabar, and M. Shiroie, "Automatic signal segmentation using the fractal dimension and weighted moving average filter," *J. Elect. Comput. Sci.*, vol. 11, no. 6, pp. 8–15, Oct. 2011.

[114] A. Bulling, J. A. Ward, and H. Gellersen, "Multimodal recognition of reading activity in transit using body-worn sensors," *ACM Trans. Appl. Perception*, vol. 9, no. 1, pp. 1–21, Mar. 2012, doi: 10.1145/2134203.2134205.

[115] T. Stiefmeier, D. Roggen, and G. Tröster, "Gestures are strings: Efficient online gesture spotting and classification using string matching," in *Proc. 2nd Int. Conf. Body Area Netw. BodyNets*, Florence, Italy, 2007, pp. 16:1–16:8.

[116] U. Blanke, R. Rehner, and B. Schiele, "South by south-east or sitting at the design: Can orientation be a place?" in *Proc. 15th Annu. Int. Symp. Wearable Comput.*, Jun. 2011, pp. 43–46.

[117] M. Bachlin et al., "Wearable assistant for Parkinson's disease patients with the freezing of gait symptom," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 436–446, Mar. 2010, doi: 10.1109/TITB.2009.2036165.

[118] I. Khairuddin, S. N. Sidek, A. P. A. Majeed, M. Razman, A. A. Puzi, and H. Yusof, "The classification of movement intention through machine learning models: The identification of significant time-domain EMG features," *PeerJ Comput. Sci.*, vol. 7, pp. 1–15, Feb. 2021, doi: 10.7717/peerj-cs.379.

[119] R. Jain, V. B. Semwal, and P. Kaushik, "Stride segmentation of inertial sensor data using statistical methods for different walking activities," *Robotica*, vol. 40, no. 8, pp. 2567–2580, Dec. 2021, doi: 10.1017/s026357472100179x.

[120] D. Jung et al., "Deep neural network-based gait classification using wearable inertial sensor data," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Berlin, Germany, Jul. 2019, pp. 3624–3628.

[121] C. Oz and M. C. Leu, "American sign language word recognition with a sensory glove using artificial neural networks," *Eng. Appl. Artif. Intell.*, vol. 24, no. 7, pp. 1204–1213, Oct. 2011, doi: 10.1016/j.engappai.2011.06.015.

[122] N. Wang, S. J. Redmond, E. Ambikairajah, B. G. Celler, and N. H. Lovell, "Can triaxial accelerometry accurately recognize inclined walking terrains?" *IEEE Trans. Biomed. Eng.*, vol. 57, no. 10, pp. 2506–2516, Oct. 2010, doi: 10.1109/TBME.2010.2049357.

[123] L. Peng, L. Chen, X. Wu, H. Guo, and G. Chen, "Hierarchical complex activity representation and recognition using topic model and classifier level fusion," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 6, pp. 1369–1379, Jun. 2017, doi: 10.1109/TBME.2016.2604856.

[124] P. Tahafchi and J. W. Judy, "Freezing-of-gait detection using wearable-sensor technology and neural-network classifier," in *Proc. IEEE Sensors*, Montreal, QC, Canada, Oct. 2019, pp. 1–4.

[125] A. Zinnen, C. Wojek, and B. Schiele, "Multi activity recognition based on body model derived primitives," Presented at the 4th Int. Symp. Location Context Awareness, Tokyo, Japan, May 2009.

[126] A. Zinnen, U. Blanke, and B. Schiele, "An analysis of sensor-oriented vs. model-based activity recognition," in *Proc. Int. Symp. Wearable Comput.*, Linz, Austria, Sep. 2009, pp. 1–4.

[127] R. Z. U. Rehman, S. Del Din, Y. Guan, A. J. Yarnall, J. Q. Shi, and L. Rochester, "Selecting clinically relevant gait characteristics for classification of early Parkinson's disease: A comprehensive machine learning approach," *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, Nov. 2019.

[128] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, and M. Lang, "Benchmark for filter methods for feature selection in high-dimensional classification data," *Comput. Statist. Data Anal.*, vol. 143, pp. 1–22, Mar. 2020, doi: 10.1016/j.csda.2019.106839.

[129] A. Jovic, K. Brkic, and N. Bogunovic, "A review of feature selection methods with applications," in *Proc. 38th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, Opatija, Croatia, May 2015, pp. 1200–1205.

[130] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *KAIS*, vol. 8, no. 1, pp. 1–37, Jul. 2005, doi: 10.1007/s10115-012-0487-8.

[131] A. Ben Brahim and M. Limam, "A hybrid feature selection method based on instance learning and cooperative subset search," *Pattern Recognit. Lett.*, vol. 69, pp. 28–34, Jan. 2016, doi: 10.1016/j.patrec.2015.10.005.

[132] M. Naseriparsa, A.-M. Bidgoli, and T. Varaee, "A hybrid feature selection method to improve performance of a group of classification algorithms," *Int. J. Comput. Appl.*, vol. 69, no. 17, pp. 28–35, May 2013, doi: 10.5120/12065-8172.

[133] A. Subasi, "Classification examples for healthcare," in *Practical Machine Learning for Data Analysis Using Python*. Cambridge, MA, USA: AP, 2020, ch. 4, pp. 203–322.

[134] T. I. Netoff, "The ability to predict seizure onset," in *Engineering in Medicine Advances and Challenges*. Cambridge, MA, USA: AP, 2019, ch. 14, pp. 365–378.

[135] M. Landgraf et al., "Gesture recognition with sensor data fusion of two complementary sensing methods," in *Proc. 7th IEEE Int. Conf. Biomed. Robot. Biomechatronics (Biorob)*, Aug. 2018, pp. 795–800.

[136] R. Su, X. Chen, S. Cao, and X. Zhang, "Random forest-based recognition of isolated sign language sub words using data from accelerometers and surface electromyographic sensors," *Sensors*, vol. 16, no. 1, pp. 1–15, Jan. 2016, doi: 10.3390/s16010100.

[137] N. Dawar and N. Kehtarnavaz, "Data flow synchronization of a real-time fusion system to detect and recognize smart TV gestures," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2018, pp. 1–4.

[138] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "SoundSense: Scalable sound sensing for people-centric applications on mobile phones," in *Proc. 7th Int. Conf. Mobile Syst., Appl., Services*, Jun. 2009, pp. 165–178.

[139] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Proc. Int. Workshop Wearable Implant. Body Sensor Netw.*, Cambridge, MA, USA, 2006, pp. 113–116.

[140] A. Bulling, J. A. Ward, H. Gellersen, and G. Tröster, "Eye movement analysis for activity recognition using electrooculography," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 741–753, Apr. 2011, doi: 10.1109/TPAMI.2010.86.

[141] B. T. Nukala et al., "A real-time robust fall detection system using a wireless gait analysis sensor and an artificial neural network," in *Proc. IEEE Healthcare Innov. Conf. (HIC)*, Seattle, WA, USA, Oct. 2014, pp. 219–222.

[142] B. T. Nukala et al., "An efficient and robust fall detection system using wireless gait analysis sensor with artificial neural network (ANN) and support vector machine (SVM) algorithms," *Open J. Appl. Biosensor*, vol. 3, no. 4, pp. 29–39, 2014, doi: 10.4236/OJAB.2014.34004.

[143] J. Taborri, S. Rossi, E. Palermo, F. Patanè, and P. Cappa, "A novel HMM distributed classifier for the detection of gait phases by means of a wearable inertial sensor network," *Sensors*, vol. 14, no. 9, pp. 16212–16234, Sep. 2014, doi: 10.3390/s140916212.

[144] L. Rong, D. Zhiguo, Z. Jianzhong, and L. Ming, "Identification of individual walking patterns using gait acceleration," in *Proc. 1st Int. Conf. Bioinf. Biomed. Eng.*, Wuhan, China, 2007, pp. 543–546.

[145] H. Li, A. Shrestha, H. Heidari, J. Le Kernec, and F. Fioranelli, "Bi-LSTM network for multimodal continuous human activity recognition and fall detection," *IEEE Sensors J.*, vol. 20, no. 3, pp. 1191–1201, Feb. 2020, doi: 10.1109/JSEN.2019.2946095.

[146] J. A. Paradiso, S. J. Morris, A. Y. Benbasat, and E. Asmussen, "Interactive therapy with instrumented footwear," in *Proc. CHI Extended Abstr. Hum. Factors Comput. Syst.*, Vienna, Austria, Apr. 2004, pp. 1341–1343.

[147] K. Stapor, "Evaluation of classifiers: Current methods and future research directions," in *Proc. Ann. Comput. Sci. Inf. Syst.*, Prague, Czech Republic, Sep. 2017, pp. 37–40.

[148] N. Tran, J. G. Schneider, I. Weber, and A. K. Qin, "Hyper-parameter optimization in classification: To-do or not-to-do," *Pattern Recognit.*, vol. 103, pp. 1–32, Jul. 2020, doi: 10.1016/j.patcog.2020.107245.

[149] F. Khan, S. Kanwal, S. Alamri, and B. Mumtaz, "Hyper-parameter optimization of classifiers, using an artificial immune network and its application to software bug prediction," *IEEE Access*, vol. 8, pp. 20954–20964, 2020, doi: 10.1109/ACCESS.2020.2968362.

[150] J. P. Mueller and L. Massaron, "Validating machine learning," in *Machine Learning for Dummies*, 2nd ed. Hoboken, NJ, USA: JWS, 2021, ch. 9.

[151] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," Presented at the Pervasive Comput., 2nd Int. Conf., Vienna, Austria, Apr. 2004.

[152] C. Ferri, J. Hernández-Orallo, and R. Modroiu, "An experimental comparison of performance measures for classification," *Pattern Recognit. Lett.*, vol. 30, no. 1, pp. 27–38, Jan. 2009, doi: 10.1016/j.patrec.2008.08.010.

[153] M. Hossin and M. N. Sulaima, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process*, vol. 5, no. 2, pp. 1–11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.

[154] R. Caruana and A. Niculescu-Mizil, "Data mining in metric space: An empirical analysis of supervised learning performance criteria," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2004.

[155] A. Gupta and V. B. Semwal, "Occluded gait reconstruction in multi person gait environment using different numerical methods," *Multimedia Tools Appl.*, vol. 81, no. 16, pp. 23421–23448, Jul. 2022, doi: 10.1007/s11042-022-12218-2.

[156] Y. Hsu, S. Yang, H. Chang, and H. Lai, "Human daily and sport activity recognition using a wearable inertial sensor network," *IEEE Access*, vol. 6, pp. 31715–31728, 2018, doi: 10.1109/ACCESS.2018.2839766.

[157] S. Kumar, T. Yogesh, M. Prithiv, S. Alam, M. Hashim, and R. Amutha, "Data mining technique based ambient assisted living for elderly people," in *Proc. 4th Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Erode, India, Mar. 2020, pp. 505–508.

[158] J. Wu, L. Sun, and R. Jafari, "A wearable system for recognizing American sign language in real-time using IMU and surface EMG sensors," *IEEE J. Biomed. Health Informat.*, vol. 20, no. 5, pp. 1281–1290, Sep. 2016, doi: 10.1109/JBHI.2016.2598302.

[159] S. U. Yunas, A. Alharthi, and K. B. Ozanyan, "Multi-modality fusion of floor and ambulatory sensors for gait classification," in *Proc. IEEE 28th Int. Symp. Ind. Electron. (ISIE)*, Vancouver, BC, Canada, Jun. 2019, pp. 1467–1472.

[160] S. Mekruksavanich and A. Jitpattanakul, "Classification of gait pattern with wearable sensing data," in *Proc. Joint Int. Conf. Digit. Arts, Media Technol. ECTI Northern Sect. Conf. Electr., Electron., Comput. Telecommun. Eng. (ECTI DAMT-NCON)*, Jan. 2019, pp. 137–141.

[161] N. Hnoohom, S. Mekruksavanich, and A. Jitpattanakul, "Human activity recognition using triaxial acceleration data from smartphone and ensemble learning," in *Proc. 13th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Jaipur, India, Dec. 2017, pp. 408–412.

[162] N. Hnoohom, A. Jitpattanakul, P. Inluergsri, P. Wongbudsri, and W. Ployput, "Multi-sensor-based fall detection and activity daily living classification by using ensemble learning," in *Proc. Int. ECTI Northern Sect. Conf. Electr., Electron., Comput. Telecommun. Eng.*, Chiang Rai, Thailand, Feb. 2018, pp. 111–115.

[163] A. Tharwat, "Classification assessment methods," *Appl. Comput. Informat.*, vol. 17, no. 1, pp. 168–192, Jan. 2021, doi: 10.1016/j.aci.2018.08.003.

[164] M. Zhang and Z. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014, doi: 10.1109/TKDE.2013.39.

[165] R. F. De Vellis, "Inter-rater reliability," in *Encyclopedia of Social Measurement*. Cambridge, MA, USA: AP, 2005, pp. 317–322.

[166] J. Han, J. Pei, and M. Kamber, "Classification: Basic concepts," in *Data Mining: Concepts and Techniques*, 3rd ed., Amsterdam, The Netherlands: Elsevier, 2011, ch. 8, pp. 327–391.

[167] A. Shaafi, O. Salem, and A. Mehaoua, "Improving human activity recognition algorithms using wireless body sensors," Presented at the Int. Wireless Commun. Mobile Comput. (IWCMC), Limassol, Cyprus, Apr. 2020.

[168] B. Clarkson, K. Mase, and A. Pentland, "Recognizing user's context from wearable sensors: Baseline system," *J. Neurol. Sci.*, vol. 248, pp. 1–10, Jan. 2000.

[169] P. Tahafchi and J. W. Judy, "Freezing-of-gait detection using wearable sensor technology and possibilistic K-nearest-neighbor algorithm," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Berlin, Germany, Jul. 2019, pp. 4246–4249.

[170] S. Chen et al., "Discriminative information added by wearable sensors for early screening—A case study on diabetic peripheral neuropathy," in *Proc. IEEE 16th Int. Conf. Wearable Implant. Body Sensor Netw. (BSN)*, Chicago, IL, USA, May 2019, pp. 1–4.

[171] G. Cola, M. Avvenuti, A. Vecchio, G. Yang, and B. Lo, "An unsupervised approach for gait-based authentication," in *Proc. IEEE 12th Int. Conf. Wearable Implant. Body Sensor Netw. (BSN)*, Cambridge, MA, USA, Jun. 2015, pp. 1–6.

[172] L. Rong, Z. Jianzhong, L. Ming, and H. Xiangfeng, "A wearable acceleration sensor system for gait recognition," in *Proc. 2nd IEEE Conf. Ind. Electron. Appl.*, Harbin, China, May 2007, pp. 1–4.

[173] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S. Makela, and H. Ailisto, "Identifying users of portable devices from gait pattern with accelerometers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Philadelphia, PA, USA, Mar. 2005, pp. 1–12.

[174] A. Bulling, J. A. Ward, H. Gellersen, and G. Tröster, "Robust recognition of reading activity in transit using wearable electrooculography," Presented at the 6th Int. Conf., Pervasive, Sydney, NSW, Australia, May 2008.

[175] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, pp. 1–12, Mar. 2015, doi: 10.1371/journal.pone.0118432.

[176] D. Gafurov, E. Snekkenes, and P. Bours, "Improved gait recognition performance using cycle matching," in *Proc. IEEE 24th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Perth, WA, Australia, Apr. 2010, pp. 1–6.

[177] R. R. Fletcher, M.-Z. Poh, and H. Eydgahi, "Wearable sensors: Opportunities and challenges for low-cost health care," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol.*, Buenos Aires, Argentina, Aug. 2010, pp. 1–14.

[178] *VN-100 Introduction*. Accessed: Jun. 8, 2023. [Online]. Available: https://www.vectornav.com/products/vn-100

[179] J. P. Gupta, N. Singh, P. Dixit, V. B. Semwal, and S. R. Dubey, "Human activity recognition using gait pattern," *Int. J. Comput. Vis. Image Process.*, vol. 3, no. 3, pp. 31–53, Jul. 2013, doi: 10.4018/ijcvip.2013070103.

[180] A. M. Sabatini, "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 7, pp. 1346–1356, Jul. 2006, doi: 10.1109/TBME.2006.875664.

[181] S. Sabatelli, M. Galgani, L. Fanucci, and A. Rocchi, "A double-stage Kalman filter for orientation tracking with an integrated processor in 9-D IMU," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 3, pp. 590–598, Mar. 2013, doi: 10.1109/TIM.2012.2218692.

[182] K. H. Tran and M. T. Chew, "Kalman filtering for wearable fitness monitoring," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, Pisa, Italy, May 2015, pp. 2020–2025.

[183] X. F. He and P. Niyogi, "Locality preserving projections," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2004, pp. 153–160.

[184] K. E. Hild, D. Erdogmus, K. Torkkola, and J. C. Principe, "Feature extraction using information-theoretic learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1385–1392, Sep. 2006, doi: 10.1109/TPAMI.2006.186.

[185] A. Akbari and R. Jafari, "An autoencoder-based approach for recognizing null class in activities of daily living in-the-wild via wearable motion sensors," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 3392–3396.

[186] X. Lai, Q. Liu, X. Wei, W. Wang, G. Zhou, and G. Han, "A survey of body sensor networks," *Sensors*, vol. 13, no. 5, pp. 5406–5447, Apr. 2013, doi: 10.3390/s130505406.

[187] M. Yaghoubi, K. Ahmed, and Y. Miao, "Wireless body area network (WBAN): A survey on architecture, technologies, energy consumption, and security challenges," *J. Sensor Actuator Netw.*, vol. 11, no. 4, p. 67, Oct. 2022, doi: 10.3390/jsan11040067.

[188] N. Javaid, A. Ahmad, Y. Khan, Z. A. Khan, and T. A. Alghamdi, "A relay based routing protocol for wireless in-body sensor networks," *Wireless Pers. Commun.*, vol. 80, no. 3, pp. 1063–1078, Feb. 2015, doi: 10.1007/s11277-014-2071-x.

[189] Y. Wang, S. Cang, and H. Yu, "A data fusion-based hybrid sensory system for older people's daily activity and daily routine recognition," *IEEE Sensors J.*, vol. 18, no. 16, pp. 6874–6888, Aug. 2018, doi: 10.1109/JSEN.2018.2833745.

[190] C. Zhu, Q. Cheng, and W. Sheng, "Human activity recognition via motion and vision data fusion," in *Proc. Conf. Rec. 44th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2010, pp. 332–336.

[191] Y. Zhang et al., "Static and dynamic human arm/hand gesture capturing and recognition via multiinformation fusion of flexible strain sensors," *IEEE Sensors J.*, vol. 20, no. 12, pp. 6450–6459, Jun. 2020, doi: 10.1109/JSEN.2020.2965580.

[192] Y. Bao, F. Sun, X. Hua, B. Wang, and J. Yin, "Operation action recognition using wearable devices with inertial sensors," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Nov. 2017, pp. 536–541.

[193] J. G. Colli-Alfaro, A. Ibrahim, and A. L. Trejos, "Design of user-independent hand gesture recognition using multilayer perceptron networks and sensor fusion techniques," in *Proc. IEEE 16th Int. Conf. Rehabil. Robot. (ICORR)*, Toronto, ON, Canada, Jun. 2019, pp. 1103–1108.

[194] K. Murao, T. Terada, A. Yano, and R. Matsukura, "Evaluating gesture recognition by multiple-sensor-containing mobile devices," in *Proc. 15th Annu. Int. Symp. Wearable Comput.*, San Francisco, CA, USA, Jun. 2011, pp. 55–58.

[195] C. Zhu and W. Sheng, "Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living," *IEEE Trans. Syst., Man, Cybern., A, Syst. Hum.*, vol. 41, no. 3, pp. 569–573, May 2011, doi: 10.1109/TSMCA.2010.2093883.

[196] J. Liu, J. Sohn, and S. Kim, "Classification of daily activities for the elderly using wearable sensors," *J. Healthcare Eng.*, vol. 2017, pp. 1–7, Jan. 2017, doi: 10.1155/2017/8934816.

[197] H. Huang, X. Li, and Y. Sun, "A triboelectric motion sensor in wearable body sensor network for human activity recognition," in *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Orlando, FL, USA, Aug. 2016, pp. 4889–4892.

[198] G. Bhat, J. Park, H. G. Lee, and U. Y. Ogras, "Sensor-classifier co-optimization for wearable human activity recognition applications," in *Proc. IEEE Int. Conf. Embedded Softw. Syst. (ICESS)*, Las Vegas, NV, USA, Jun. 2019, pp. 1–4.

[199] H. Kim et al., "Collaborative classification for daily activity recognition with a smartwatch," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Budapest, Hungary, Oct. 2016, pp. 3707–3712.

[200] A. Moschetti, L. Fiorini, D. Esposito, P. Dario, and F. Cavallo, "Daily activity recognition with inertial ring and bracelet: An unsupervised approach," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Singapore, May 2017, pp. 3250–3255.

[201] X. Zhou, W. Liang, K. I. Wang, H. Wang, L. T. Yang, and Q. Jin, "Deep-learning-enhanced human activity recognition for Internet of Healthcare Things," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6429–6438, Jul. 2020, doi: 10.1109/JIOT.2020.2985082.

[202] Z. Chen, C. Jiang, S. Xiang, J. Ding, M. Wu, and X. Li, "Smartphone sensor-based human activity recognition using feature fusion and maximum full a posteriori," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 3992–4001, Jul. 2020, doi: 10.1109/TIM.2019.2945467.

[203] T. Zebin, P. J. Scully, and K. B. Ozanyan, "Evaluation of supervised classification algorithms for human activity recognition with inertial sensors," in *Proc. IEEE Sensors*, Glasgow, U.K., Oct. 2017, pp. 1–3.

[204] S. Benatti et al., "A versatile embedded platform for EMG acquisition and gesture recognition," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 5, pp. 620–630, Oct. 2015, doi: 10.1109/TBCAS.2015.2476555.

[205] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and EMG sensors," *IEEE Trans. Syst. Man Cybern., A Syst. Hum.*, vol. 41, no. 6, pp. 1046–1076, Nov. 2011, doi: 10.1109/TSMCA.2011.2116004.

[206] Y. Li, X. Chen, X. Zhang, K. Wang, and Z. J. Wang, "A sign-component-based framework for Chinese sign language recognition using accelerometer and sEMG data," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 10, pp. 2695–2704, Oct. 2012, doi: 10.1109/TBME.2012.2190734.

[207] A. Bashir, F. Malik, F. Haider, M. Ehatisham-ul-Haq, A. Raheel, and A. Arsalan, "A smart sensor-based gesture recognition system for media player control," in *Proc. 3rd Int. Conf. Comput., Math. Eng. Technol. (iCoMET)*, Sukkur, Pakistan, Jan. 2020, pp. 1–6.

[208] G. Yuan, X. Liu, Q. Yan, S. Qiao, Z. Wang, and L. Yuan, "Hand gesture recognition using deep feature fusion network based on wearable sensors," *IEEE Sensors J.*, vol. 21, no. 1, pp. 539–547, Jan. 2021, doi: 10.1109/JSEN.2020.3014276.

[209] S. Shin and W. Sung, "Dynamic hand gesture recognition for wearable devices with low complexity recurrent neural networks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Montreal, QC, Canada, May 2016, pp. 2274–2277.

[210] K. Czuszynski, J. Ruminski, and A. Kwasniewska, "Gesture recognition with the linear optical sensor and recurrent neural networks," *IEEE Sensors J.*, vol. 18, no. 13, pp. 5429–5438, Jul. 2018, doi: 10.1109/JSEN.2018.2834968.

[211] Z. Lu, X. Chen, Q. Li, X. Zhang, and P. Zhou, "A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices," *IEEE Trans. Hum.-Mach. Syst.*, vol. 44, no. 2, pp. 293–299, Apr. 2014, doi: 10.1109/THMS.2014.2302794.

[212] O. Amft, H. Junker, and G. Troster, "Detection of eating and drinking arm gestures using inertial body-worn sensors," in *Proc. 9th IEEE Int. Symp. Wearable Comput.*, Osaka, Japan, Oct. 2005, pp. 160–163.

[213] A. Bulling and D. Roggen, "Recognition of visual memory recall processes using eye movement analysis," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, Beijing, China, Sep. 2011, pp. 1–4.

[214] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," Presented at the Int. Conf. Pervasive Comput., Dublin, Ireland, May 2006.

[215] T. Starner, J. Weaver, and A. Pentland, "A wearable computer based American sign language recognizer," in *Dig. Papers, 1st Int. Symp. Wearable Comput.*, Cambridge, MA, USA, Oct. 1997, pp. 241–250.

[216] T. Westeyn, K. Vadas, X. Bian, T. Starner, and G. D. Abowd, "Recognizing mimicked autistic self-stimulatory behaviors using HMMs," in *Proc. 9th IEEE Int. Symp. Wearable Comput.*, Osaka, Japan, Oct. 2005, pp. 164–167.

[217] G. Ogris, T. Stiefmeier, P. Lukowicz, and G. Troster, "Using a complex multi-modal on-body sensor system for activity spotting," in *Proc. 12th IEEE Int. Symp. Wearable Comput.*, Pittsburgh, PA, USA, Oct. 2008, pp. 1–4.

[218] A. Godfrey, R. Conway, D. Meagher, and G. Ó. Laighin, "Direct measurement of human movement by accelerometry," *Med. Eng. Phys.*, vol. 30, no. 10, pp. 1364–1386, Dec. 2008, doi: 10.1016/j.medengphy.2008.09.005.

[219] M. Bachlin et al., "Potentials of enhanced context awareness in wearable assistants for Parkinson's disease patients with the freezing of gait syndrome," in *Proc. Int. Symp. Wearable Comput.*, Linz, Austria, Sep. 2009, pp. 1–10.

[220] S.-W. Lee and K. Mase, "Activity and location recognition using wearable sensors," *IEEE Pervasive Comput.*, vol. 1, no. 3, pp. 24–32, Jul. 2002, doi: 10.1109/MPRV.2002.1037719.

[221] G. Pirkl, K. Stockinger, K. Kunze, and P. Lukowicz, "Adapting magnetic resonant coupling based relative positioning technology for wearable activitiy recogniton," in *Proc. 12th IEEE Int. Symp. Wearable Comput.*, Pittsburgh, PA, USA, Oct. 2008, pp. 1–3.

[222] J. Cheng, O. Amft, and P. Lukowicz, "Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition," Presented at the 8th Int. Conf., Pervasive, Helsinki, Finland, May 2010.

[223] U. Anliker et al., "AMON: A wearable multiparameter medical monitoring and alert system," *IEEE Trans. Inf. Technol. Biomed.*, vol. 8, no. 4, pp. 415–427, Dec. 2004, doi: 10.1109/TITB.2004.837888.

[224] N. Oliver and F. Flores-Mangas, "HealthGear: Automatic sleep apnea detection and monitoring with a mobile phone," *J. Commun.*, vol. 2, no. 2, pp. 1–12, Mar. 2007, doi: 10.4304/jcm.2.2.1-9.

[225] W.-J. Kang, J.-R. Shiu, C.-K. Cheng, J.-S. Lai, H.-W. Tsao, and T.-S. Kuo, "The application of cepstral coefficients and maximum likelihood method in EMG pattern recognition [movements classification]," *IEEE Trans. Biomed. Eng.*, vol. 42, no. 8, pp. 777–785, Aug. 1995, doi: 10.1109/10.398638.

[226] P. Lukowicz, F. Hanser, C. Szubski, and W. Schobersberger, "Detecting and interpreting muscle activity with wearable force sensors," in *Proc. 4th Int. Conf., Pervasive*, Dublin, Ireland, May 2006, pp. 101–116.

[227] C. Mattmann, O. Amft, H. Harms, G. Troster, and F. Clemens, "Recognizing upper body postures using textile strain sensors," in *Proc. 11th IEEE Int. Symp. Wearable Comput.*, Boston, MA, USA, Oct. 2007, pp. 1–3.

[228] S. J. Morris and J. A. Paradiso, "Shoe-integrated sensor system for wireless gait analysis and real-time feedback," in *Proc. 2nd Joint 24th Annu. Conf. Annu. Fall Meeting Biomed. Eng. Soc., Eng. Med. Biol.*, Houston, TX, USA, Oct. 2002, pp. 2468–2469.

[229] L. Liao, D. Fox, and H. Kautz, "Location-based activity recognition using relational Markov networks," Presented at the 19th Int. Joint Conf. Artif. Intell., Jul. 2005, pp. 773–778.

[230] J. Krumm and E. Horvitz, "Predestination: Inferring destinations from partial trajectories," Presented at the 8th Int. Conf. Ubiquitous Comput., Orange County, CA, USA, Sep. 2006.

[231] G. Schindler, C. Metzger, and T. Starner, "A wearable interface for topological mapping and localization in indoor environments," Presented at the 2nd Int. Conf. Location- Context-Awareness, Dublin, Ireland, May 2006.

[232] A. Cassinelli, C. Reynolds, and M. Ishikawa, "Augmenting spatial awareness with haptic radar," in *Proc. 10th IEEE Int. Symp. Wearable Comput.*, Montreux, Switzerland, Oct. 2006, pp. 61–64.

[233] V. E. Kosmidou and L. J. Hadjileontiadis, "Sign language recognition using intrinsic-mode sample entropy on sEMG and accelerometer data," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 12, pp. 2879–2890, Dec. 2009, doi: 10.1109/TBME.2009.2013200.

[234] V. E. Kosmidou, L. J. Hadjileontiadis, and S. M. Panas, "Evaluation of surface EMG features for the recognition of American sign language gestures," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, New York, NY, USA, Aug. 2006, pp. 6197–200.

[235] Q. Chen, A. El-Sawah, C. Joslin, and N. D. Georganas, "A dynamic gesture interface for virtual environments based on hidden Markov models," in *Proc. IREE Int. Worksho Haptic Audio Vis. Environments Appl.*, Ottawa, ON, Canada, 2005, pp. 1–3.

[236] S. T. Moore, H. G. MacDougall, and W. G. Ondo, "Ambulatory monitoring of freezing of gait in Parkinson's disease," *J. Neurosci. Methods*, vol. 167, no. 2, pp. 340–348, Jan. 2008, doi: 10.1016/j.jneumeth.2007.08.023.

[237] J. Camps et al., "Deep learning for detecting freezing of gait episodes in Parkinson's disease based on accelerometers," Presented at the 14th Int. Work-Conf. Artif. Neural Netw., Cadiz, Spain, Jun. 2017.

[238] C. Azevedo Coste, B. Sijobert, R. Pissard-Gibollet, M. Pasquier, B. Espiau, and C. Geny, "Detection of freezing of gait in Parkinson disease: Preliminary results," *Sensors*, vol. 14, no. 4, pp. 6819–6827, Apr. 2014, doi: 10.3390/s140406819.

[239] B. T. Cole, S. H. Roy, and S. H. Nawab, “Detecting freezing-of-gait during unscripted and unconstrained activity,” in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Boston, MA, USA, Aug. 2011, pp. 5649–5652.

[240] B. T. Cole, S. H. Roy, C. J. De Luca, and S. H. Nawab, “Dynamic neural network detection of tremor and dyskinesia from wearable sensor data,” in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol.*, Aug. 2010, pp. 6062–6065.

[241] P. Tahafchi et al., “Freezing-of-gait detection using temporal, spatial, and physiological features with a support-vector-machine classifier,” in *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2017, pp. 2867–2870.

[242] N. Shibuya et al., “A real-time fall detection system using a wearable gait analysis sensor and a support vector machine (SVM) classifier,” in *Proc. 8th Int. Conf. Mobile Comput. Ubiquitous Netw. (ICMU)*, Hakodate, Japan, Jan. 2015, pp. 1–5.

[243] T. Nakano et al., “Gaits classification of normal vs. patients by wireless gait sensor and support vector machine (SVM) classifier,” in *Proc. IEEE/ACIS 15th Int. Conf. Comput. Inf. Sci. (ICIS)*, Okayama, Japan, Jun. 2016, pp. 1–6.

[244] R. Williamson and B. J. Andrews, “Gait event detection for FES using accelerometers and supervised machine learning,” *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 3, pp. 312–319, Oct. 2000, doi: 10.1109/86.867873.

[245] M. Yuwono, S. W. Su, Y. Guo, B. D. Moulton, and H. T. Nguyen, “Unsupervised nonparametric method for gait analysis using a waist-worn inertial sensor,” *Appl. Soft Comput.*, vol. 14, pp. 72–80, Jan. 2014.

[246] U. Martinez-Hernandez, I. Mahmood, and A. A. Dehghani-Sanij, “Simultaneous Bayesian recognition of locomotion and gait phases with wearable sensors,” *IEEE Sensors J.*, vol. 18, no. 3, pp. 1282–1290, Feb. 2018, doi: 10.1109/JSEN.2017.2782181.

[247] T. Liu, Y. Inoue, and K. Shibata, “A wearable sensor system for human motion analysis and humanoid robot control,” in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Kunming, China, Dec. 2006, pp. 1–8.

[248] S. K. Al Kork, I. Gowthami, X. Savatier, T. Beyrouthy, J. A. Korbane, and S. Roshdi, “Biometric database for human gait recognition using wearable sensors and a smartphone,” in *Proc. 2nd Int. Conf. Bio-Eng. Smart Technol. (BioSMART)*, Paris, France, Aug. 2017, pp. 1–4.

[249] R. D. Gurchiek et al., “Remote gait analysis using wearable sensors detects asymmetric gait patterns in patients recovering from ACL reconstruction,” in *Proc. IEEE 16th Int. Conf. Wearable Implant. Body Sensor Netw. (BSN)*, Chicago, IL, USA, May 2019, pp. 1–4.

[250] M. S. H. Aung et al., “Automated detection of instantaneous gait events using time frequency analysis and manifold embedding,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 21, no. 6, pp. 908–916, Nov. 2013, doi: 10.1109/TNSRE.2013.2239313.

[251] T. Liu, Y. Inoue, K. Shibata, and X. Tang, “A wearable inertial sensor system for human motion analysis,” in *Proc. Int. Symp. Comput. Intell. Robot. Autom.*, Espoo, Finland, Jun. 2005, pp. 1–4.

[252] C. P. Burgos et al., “In-ear accelerometer-based sensor for gait classification,” *IEEE Sensors J.*, vol. 20, no. 21, pp. 12895–12902, Nov. 2020, doi: 10.1109/JSEN.2020.3002589.

[253] S. Potluri, A. B. Chandran, C. Diedrich, and L. Schega, “Machine learning based human gait segmentation with wearable sensor platform,” in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Berlin, Germany, Jul. 2019, pp. 588–594.

[254] V. Mandalapu, N. Homdee, J. M. Hart, J. Lach, S. Bodkin, and J. Gong, “Developing computational models for personalized ACL injury classification,” in *Proc. IEEE 16th Int. Conf. Wearable Implant. Body Sensor Netw. (BSN)*, Chicago, IL, USA, May 2019, pp. 1–4.

[255] K. Bichave, O. Brewer, M. Gusinov, P. P. Markopoulos, and I. Puchades, “Gait recognition based on tensor analysis of acceleration data from wearable sensors,” in *Proc. IEEE Western New York Image Signal Process. Workshop (WNYISPW)*, Rochester, NY, USA, Oct. 2018, pp. 1–5.

[256] K. Kitagawa, T. Uezono, T. Nagasaki, S. Nakano, and C. Wada, “Classification method of assistance motions for standing-up with different foot anteroposterior positions using wearable sensors,” in *Proc. Int. Conf. Inf. Commun. Technol. Robot.*, Sep. 2018, pp. 1–3.

[257] C. Caramia et al., “IMU-based classification of Parkinson’s disease from gait: A sensitivity analysis on sensor location and feature selection,” *IEEE J. Biomed. Health Informat.*, vol. 22, no. 6, pp. 1765–1774, Nov. 2018, doi: 10.1109/JBHI.2018.2865218.

[258] A. Gümüşçü, K. Karadag, M. Çaliskan, M. E. Tenekeci, and D. Akaslan, “Gender classification via wearable gait analysis sensor,” in *Proc. 26th Signal Process. Commun. Appl. Conf. (SIU)*, Izmir, Turkey, May 2018, pp. 1–4.

[259] Y. Xu, W. Chen, J. Wang, and J. Zhang, “Fuzzy control in gait pattern classification using wearable sensors,” in *Proc. IEEE 10th Conf. Ind. Electron. Appl. (ICIEA)*, Auckland, New Zealand, Jun. 2015, pp. 62–67.

[260] Z. Li, W. Chen, J. Wang, and J. Liu, “An automatic recognition system for patients with movement disorders based on wearable sensors,” in *Proc. 9th IEEE Conf. Ind. Electron. Appl.*, Hangzhou, China, Jun. 2014, pp. 1948–1953.

[261] R. L. Evans and D. K. Arvind, “Detection of gait phases using orient specks for mobile clinical gait analysis,” in *Proc. 11th Int. Conf. Wearable Implant. Body Sensor Netw.*, Zurich, Switzerland, Jun. 2014, pp. 149–154.

[262] S. Crea et al., “Development of gait segmentation methods for wearable foot pressure sensors,” in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2012, pp. 5018–5021.

[263] D. Novak et al., “Early recognition of gait initiation and termination using wearable sensors,” in *Proc. 4th IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics (BioRob)*, Rome, Italy, Jun. 2012, pp. 1937–1942.

[264] C. Wong, S. McKeague, J. Correa, J. Liu, and G. Yang, “Enhanced classification of abnormal gait using BSN and depth,” in *Proc. 9th Int. Conf. Wearable Implant. Body Sensor Netw.*, London, U.K., May 2012, pp. 166–171.

[265] M. ElSayed, A. Alsebai, A. Salaheldin, N. E. Gayar, and M. ElHelw, “Ambient and wearable sensing for gait classification in pervasive healthcare environments,” in *Proc. 12th IEEE Int. Conf. E-Health Netw., Appl. Service*, Lyon, France, Jul. 2010, pp. 240–245.

[266] M. Chen, J. Yan, and Y. Xu, “Gait pattern classification with integrated shoes,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, Oct. 2009, pp. 833–839.

[267] M. A. Hanson, H. C. Powell Jr., A. T. Barth, J. Lach, and M. Brandt-Pearce, “Neural network gait classification for on-body inertial sensors,” in *Proc. 6th Int. Workshop Wearable Implant. Body Sensor Netw.*, Berkeley, CA, USA, Jun. 2009, pp. 1–5.

[268] W. H. Wu, A. A. T. Bui, M. A. Batalin, D. Liu, and W. J. Kaiser, “Incremental diagnosis method for intelligent wearable sensor systems,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 11, no. 5, pp. 553–562, Sep. 2007, doi: 10.1109/TITB.2007.897579.

[269] B. Coley, B. Najafi, A. Paraschiv-Ionescu, and K. Aminian, “Stair climbing detection during daily physical activity using a miniature gyroscope,” *Gait Posture*, vol. 22, no. 4, pp. 287–294, Dec. 2005, doi: 10.1016/j.gaitpost.2004.08.008.

[270] M. Sekinea, T. Tamurab, T. Togawac, and Y. Fukuia, “Classification of waist-acceleration signals in a continuous walking record,” *Med. Eng. Phys.*, vol. 22, no. 4, pp. 285–291, May 2000, doi: 10.1016/s1350-4533(00)00041-2.

[271] N. Kern, B. Schiele, and A. Schmidt, “Multi-sensor activity context detection for wearable computing,” in *Proc. 1st Eur. Symp.*, Veldhoven, The Netherlands, Nov. 2003, pp. 220–232.

[272] A. Turner and S. Hayes, “The classification of minor gait alterations using wearable sensors and deep learning,” *IEEE Trans. Biomed. Eng.*, vol. 66, no. 11, pp. 3136–3145, Nov. 2019, doi: 10.1109/TBME.2019.2900863.

[273] C. Strohrmann et al., “Automated assessment of gait deviations in children with cerebral palsy using a sensorized shoe and active shape models,” in *Proc. IEEE Int. Conf. Body Sensor Netw.*, Cambridge, MA, USA, May 2013, pp. 1–6.

[274] S. Bamberg, A. Y. Benbasat, D. M. Scarborough, D. E. Krebs, and J. A. Paradiso, “Gait analysis using a shoe-integrated wireless sensor system,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 4, pp. 413–423, Jul. 2008, doi: 10.1109/TITB.2007.899493.

[275] I. P. I. Pappas, M. R. Popovic, T. Keller, V. Dietz, and M. Morari, “A reliable gait phase detection system,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 9, no. 2, pp. 113–125, Jun. 2001, doi: 10.1109/7333.928571.

[276] O. A. Malik, S. M. N. A. Senanayake, and D. Zaheer, “An adaptive interval type-2 fuzzy logic framework for classification of gait patterns of anterior cruciate ligament reconstructed subjects,” in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Beijing, China, Jul. 2014, pp. 1068–1075.

[277] J. Kim, K. B. Lee, and S. G. Hong, “Random forest based-biometric identification using smart shoes,” in *Proc. 11th Int. Conf. Sens. Technol. (ICST)*, Sydney, NSW, Australia, Dec. 2017, pp. 1–4.

[278] I. P. I. Pappas, T. Keller, S. Mangold, M. Popovic, V. Dietz, and M. Morari, “A reliable gyroscope-based gait-phase detection sensor embedded in a shoe insole,” *IEEE Sensors J.*, vol. 4, no. 2, pp. 268–274, Apr. 2004, doi: 10.1109/JSEN.2004.823671.

[279] A. H. Johnston and G. M. Weiss, "Smartwatch-based biometric gait recognition," in *Proc. IEEE 7th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Arlington, VA, USA, Sep. 2015, pp. 1–6.

[280] S. Kuen Ng and H. J. Chizeck, "Fuzzy model identification for classification of gait events in paraplegics," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 4, pp. 536–544, Nov. 1997, doi: 10.1109/91.649904.

[281] G. Pan, Y. Zhang, and Z. Wu, "Accelerometer-based gait recognition via voting by signature points," *Electron. Lett.*, vol. 45, no. 22, pp. 1116–1118, Nov. 2009, doi: 10.1049/el.2009.2301.

[282] S. Sprager and D. Zazula, "Gait identification using cumulants of accelerometer data," Presented at the 2nd WSEAS Int. Conf. Sensors, Signals Vis., Imag. Simulation Mater. Sci., Baltimore, MD, USA, Nov. 2009.

[283] M. Sekine, T. Tamura, T. Fujimoto, and Y. Fukui, "Classification of walking pattern using acceleration waveform in elderly people," in *Proc. 22nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Chicago, IL, USA, Jul. 2000, pp. 1356–1359.

[284] C. A. Kirkwood, B. J. Andrews, and P. Mowforth, "Automatic detection of gait events: A case study using inductive learning techniques," *J. Biomed. Eng.*, vol. 11, no. 6, pp. 511–516, Nov. 1989, doi: 10.1016/0141-5425(89)90046-0.

[285] D. Gafurov, E. Snekkenes, and P. Bours, "Gait authentication and identification using wearable accelerometer sensor," in *Proc. IEEE Workshop Autom. Identificat. Adv. Technol.*, Alghero, Italy, Jun. 2007, pp. 220–225.

[286] D. Gafurov, K. Helkala, and T. Soendrol, "Gait recognition using acceleration from MEMS," in *Proc. 1st Int. Conf. Availability, Rel. Secur. (ARES)*, Vienna, Austria, Apr. 2006, pp. 1–5.

[287] L. Palmerini, L. Rocchi, S. Mazilu, E. Gazit, J. M. Hausdorff, and L. Chiari, "Identification of characteristic motor patterns preceding freezing of gait in Parkinson's disease using wearable sensors," *Frontiers Neurol.*, vol. 8, pp. 1–12, Aug. 2017, doi: 10.3389/fneur.2017.00394.

[288] M. M. Skelly and H. J. Chizeck, "Real-time gait event detection for paraplegic FES walking," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 9, no. 1, pp. 59–68, Mar. 2001, doi: 10.1109/7333.918277.

[289] M. Song and J. Kim, "An ambulatory gait monitoring system with activity classification and gait parameter calculation based on a single foot inertial sensor," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 4, pp. 885–893, Apr. 2018, doi: 10.1109/TBME.2017.2724543.

[290] E. Sabelman, A. Fiene, and A. Timbie, "Accelerometric activity identification for remote assessment of quality of movement," in *Proc. 26th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, San Francisco, CA, USA, Sep. 2004, pp. 1–12.

[291] S. Yin, C. Chen, H. Zhu, X. Wang, and W. Chen, "Neural networks for pathological gait classification using wearable motion sensors," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Nara, Japan, Oct. 2019, pp. 1–4.

[292] F. Demrozi, R. Bacchin, S. Tamburin, M. Cristani, and G. Pravadelli, "Toward a wearable system for predicting freezing of gait in people affected by Parkinson's disease," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 9, pp. 2444–2451, Sep. 2020, doi: 10.1109/JBHI.2019.2952618.

[293] N. K. Orphanidou, A. Hussain, R. Keight, P. Lishoa, J. Hind, and H. Al-Askar, "Predicting freezing of gait in Parkinsons disease patients using machine learning," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8.

[294] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," Presented at the 4th Int. Conf. Ambient Assist. Living Home Care, Dec. 2012.

[295] R. San-Segundo, R. Cordoba, J. Ferreiros, and L. F. D'Haro-Enríquez, "Frequency features and GMM-UBM approach for gait-based person identification using smartphone inertial signals," *Pattern Recognit. Lett.*, vol. 73, pp. 60–67, Apr. 2016, doi: 10.1016/j.patrec.2016.01.008.

[296] L. Hong-Li and M. A. Yao-Feng, "Target recognition method based on multi feature fusion and hybrid kernel SVM," *J. Shenyang Univ. Technol.*, vol. 40, no. 4, pp. 441–446, 2018, doi: 10.7688/j.issn.1000-1646.2018.04.15.

[297] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Oritz, "A public domain dataset for human activity recognition using smartphones," Presented at the 21st Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn., Bruges, Belgium, Apr. 2013.

[298] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," 2016, *arXiv:1611.08050*.

[299] E. Tapia, S. Intille, L. Lopez, and K. Larson, "The design of a portable kit of wireless sensors for naturalistic data collection," in *Proc. Int. Conf. Pervasive Comput.*, Dublin, Ireland, May 2006, pp. 1–6.

[300] A. Y. Yang, R. Jafari, S. S. Sastry, and R. Bajcsy, "Distributed recognition of human actions using wearable motion sensor networks," *J. Ambient Intell. Smart Environ.*, vol. 1, no. 2, pp. 103–115, 2009.

[301] S. Dalal, M. Alwan, R. Seifrafi, S. Kell, and D. Brown, "A rulebased approach to the analysis of elders activity data: Detection of health and possible emergency conditions," Presented at the Fall Symp., Arlington, VI, USA, 2005.

[302] P. Patil, K. S. Kumar, N. Gaud, and V. B. Semwal, "Clinical human gait classification: Extreme learning machine approach," in *Proc. 1st Int. Conf. Adv. Sci., Eng. Robot. Technol. (ICASERT)*, Dhaka, Bangladesh, May 2019, pp. 1–6.

[303] N. Dua, S. N. Singh, V. B. Semwal, and S. K. Challa, "Inception inspired CNN-GRU hybrid network for human activity recognition," *Multimedia Tools Appl.*, vol. 82, no. 4, pp. 5369–5403, Mar. 2022, doi: 10.1007/s11042-021-11885-x.

[304] A. Gupta and V. B. Semwal, *Multiple Task Human Gait Analysis and Identification: Ensemble Learning Approach*. Berlin, Germany: Springer, 2020, pp. 185–197.

[305] N. Dua, S. N. Singh, and V. B. Semwal, "Multi-input CNN-GRU based human activity recognition using wearable sensors," *Computing*, vol. 103, no. 3, pp. 1–18, Jul. 2021, doi: 10.1007/s00607-021-00928-8.

[306] V. Bijalwan, V. B. Semwal, G. Singh, and T. K. Mandal, "HDL-PSR: Modelling spatio-temporal features using hybrid deep learning approach for post-stroke rehabilitation," *Neural Process. Lett.*, vol. 55, no. 1, pp. 279–298, Jan. 2022, doi: 10.1007/s11063-022-10744-6.

[307] S. K. Challa, A. Kumar, and V. B. Semwal, "A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data," *Vis. Comput.*, vol. 38, no. 12, pp. 4095–4109, Aug. 2021, doi: 10.1007/s00371-021-02283-3.

[308] V. Bijalwan, V. B. Semwal, and V. Gupta, "Wearable sensor-based pattern mining for human activity recognition: Deep learning approach," *Ind. Robot, Int. J. Robot. Res. Appl.*, vol. 49, no. 1, pp. 21–33, Jan. 2022, doi: 10.1108/IR-09-2020-0187.

[309] V. Bijalwan, V. B. Semwal, G. Singh, and R. G. Crespo, "Heterogeneous computing model for post-injury walking pattern restoration and postural stability rehabilitation exercise recognition," *Expert Syst.*, vol. 39, no. 6, pp. 1–21, Jul. 2022, doi: 10.1111/exsy.12706.

Eghbal Foroughi Asl received the B.S. degree in electrical engineering from the Imam Khomeini International University (IKIU), Qazvin, Iran, in 2015, and the M.S. degree in electrical engineering from Shahid Beheshti University, Tehran, Iran, in 2018. He is currently pursuing the Ph.D. degree in electrical engineering with the Iran University of Science and Technology, Tehran.

His research interests include artificial intelligence, the development of wearable sensor networks, machine learning, data fusion, and Internet-of-Things (IoT)-based applications.

Mr. Foroughi Asl's awards and honors include studying at the National Organization for Development of Exceptional Talents (NODET) School for seven consecutive years and earning the title of Top Student in three consecutive courses at that school.

Saeed Ebadollahi was born Tabriz, Iran, in 1981. He received the B.Sc. and M.Sc. degrees in electrical engineering from the School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran, in 2005 and 2008, respectively, and the Ph.D. degree in control systems from the K. N. Toosi University of Technology, Tehran, in 2012.

He joined the Department of Electrical Engineering, Iran University of Science and Technology, Tehran, in 2012, where he is currently an Assistant Professor of Control Systems Engineering. His fields of research include positioning, a broad spread of applications of sensor/data fusion, and information fusion concepts in mechatronics, robotics, estimation theory, and optimal filtering.

Prof. Ebadollahi's honors and awards include winning a silver medal in the Mathematical Olympiad and obtaining the first rank in the doctoral exam at Khajeh Nasir al-Din Tusi University.

Reza Vahidnia (Member, IEEE) was born in Tehran, Iran. He received the B.Sc. degree in electrical engineering from Tehran University, Tehran, in 2006, the M.Sc. degree in electrical engineering from Tarbiat Modares University, Tehran, in 2009, and the Ph.D. degree from Ontario Tech University, Oshawa, ON, Canada, in April 2014.

He has been a Senior Internet of Things (IoT) Specialist with Rogers Communications Inc., Toronto, ON, Canada, and TELUS, Vancouver, BC, Canada, since 2016. He is currently a Faculty Member with the Department of Electrical and Computer Engineering, British Columbia Institute of Technology (BCIT), Vancouver. His research interests include wireless communications, IoT, and signal processing.

Aliakbar Jalali (Member, IEEE) received the bachelor's degree in electronic engineering from Khajeh Nasir al-Din Tusi University, Tehran, Iran, in 1985, the master's degree in electrical control engineering from the University of Oklahoma, Norman, OK, USA, in 1989, and the Ph.D. degree and postdoctoral fellowships from West Virginia University, Morgantown, WV, USA, in 1993 and 1994, respectively.

He has been a Professor with the School of Electrical Engineering, Iran University of Science and Technology, Tehran, since 1994. Throughout his work at the Iran University of Science and Technology (IUST), Tehran, he has defined, led, and managed research and development teams in the areas of information and communication technology, the Internet of Things (IoT), 3-D printers, and control systems design and their applications. He is currently a Visiting Professor with the University of Maryland, College Park, MD, USA. He has been appointed as an Adjunct Professor at the Lane Department of Computer Science and Electrical Engineering, West Virginia University.

Prof. Jalali is a member of the Iranian Society of Instrument and Control Engineering (ISCI).