

## Research Article

# LoRa-Based Smart IoT Application for Smart City: An Example of Human Posture Detection

Jinkun Han <sup>1,2</sup>, Wei Song <sup>1,3</sup>, Amanda Gozho,<sup>1</sup> Yunsick Sung,<sup>4</sup> Sumi Ji,<sup>4</sup> Liangliang Song,<sup>5</sup> Long Wen,<sup>6</sup> and Qi Zhang<sup>7</sup>

<sup>1</sup>School of Information Science and Technology, North China University of Technology, Beijing 100144, China

<sup>2</sup>Department of Computer Science, Georgia State University, Atlanta, 30303 GA, USA

<sup>3</sup>Brunel London School, North China University of Technology, Beijing 100144, China

<sup>4</sup>Department of Multimedia Engineering, Dongguk University-Seoul, Seoul 04620, Republic of Korea

<sup>5</sup>Roadway Smart (Beijing) Technology Co., Ltd., Beijing, China

<sup>6</sup>Beijing Municipal Engineering Research Institute, Beijing, China

<sup>7</sup>Beijing Capital Road Development Group Co., Ltd., Beijing, China

Correspondence should be addressed to Wei Song; [sw@ncut.edu.cn](mailto:sw@ncut.edu.cn)

Received 18 May 2020; Revised 13 June 2020; Accepted 15 July 2020; Published 4 August 2020

Academic Editor: Wei Yu

Copyright © 2020 Jinkun Han et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Scientists have explored the human body for hundreds of years, and yet more relationships between the behaviors and health are still to be discovered. With the development of data mining, artificial intelligence technology, and human posture detection, it is much more possible to figure out how behaviors and movements influence people's health and life and how to adjust the relationship between work and rest, which is needed urgently for modern people against this high-speed lifestyle. Using smart technology and daily behaviors to supervise or predict people's health is a key part of a smart city. In a smart city, these applications involve large groups and high-frequency use, so the system must have low energy consumption, a portable system, and a low cost for long-term detection. To meet these requirements, this paper proposes a posture recognition method based on multisensor and using LoRa technology to build a long-term posture detection system. LoRa WAN technology has the advantages of low cost and long transmission distances. Combining the LoRa transmitting module and sensors, this paper designs wearable clothing to make people comfortable in any given posture. Aiming at LoRa's low transmitting frequency and small size of data transmission, this paper proposes a multiprocessing method, including data denoising, data enlarging based on sliding windows, feature extraction, and feature selection using Random Forest, to make 4 values retain the most information about 125 data from 9 axes of sensors. The result shows an accuracy of 99.38% of extracted features and 95.06% of selected features with the training of 3239 groups of datasets. To verify the performance of the proposed algorithm, three testers created 500 groups of datasets and the results showed good performance. Hence, due to the energy sustainability of LoRa and the accuracy of recognition, this proposed posture recognition using multisensor and LoRa can work well when facing long-term detection and LoRa fits smart city well when facing long-distance transmission.

## 1. Introduction

Modern medical science proves human behaviors influence human health. In this day, people continue in a posture for a long time when working until they get a warning from the body that is tired, which means their bodies already hurt but they are not aware of it. Therefore, people need a reminder to let them know when to change a posture or rest.

On the other hand, for some special groups of people like children or the elderly, supervising their daily activities guarantees basic exercise in a day. To build a smart city, people not only focus on making living and transportation smarter but also wish their health is smartly supervised. With the development of big data [1] and artificial intelligence technology on health [2–4], scientists research more relationships between human daily activities, or postures, and health, to

strengthen people's bodies or prevent diseases. Human posture recognition is also widely used in sports games, health monitoring, sports rehabilitation, sports competitions, and so on [5].

With the rise of hardware, human posture recognition is not only applicable to short-term, fast, real-time recognition based on videos or pictures but is also applicable to long-term posture recognition for specific groups based on IoT (Internet of Things) devices, such as LoRa, ZigBee, and Wi-Fi devices. Considering mobile object, long-distance, and low power requirements, LoRa performs better. At present, there are needs for human posture recognition in the market, but different solutions have some different defects [6], such as high energy consumption, limited space, and network requirements. The long-term posture recognition monitoring needs a large number of transmission nodes, complex transmission network, and high-frequency usage. At the same time, long-term recognition requires that the posture information acquisition module of the system should have the characteristics of low energy consumption, easy to carry, low cost, long transmission distance, and so on, to meet all possible specific needs. Posture recognition based on computer vision can easily obtain the trajectory, contour, and other information of human motion, but the existence of some blind angles and people walking out of the observation range will cause the recognition based on cameras to be less effective. Therefore, the long-term monitoring of the human posture recognition system should also be flexible, not limited to space.

To meet these requirements, this paper proposes a multisensor-based posture recognition system using LoRa (Long Range) [7, 8]. LoRa is a low-power wide-area network (LPWAN) technology, which has the advantages of low power consumption and extra long-range signal transmission [9]. The theoretical coverage of LoRa is 15 km for suburban and 5 km for urban areas [10], in which case, one LoRa gateway is adequate for an organization, like a school or scientific research institution, to deploy a LoRa system.

In this system, wearable multisensor clothing is designed as a LoRa node to make people comfortably wear it without posture limitations while doing any daily activities. The clothing is equipped with a nine-axis IMU (Inertial Measurement Unit) sensor, an Arduino Mega board, and a LoRa shield. The nine-axis IMU sensor consists of an accelerometer [11], an angular speedometer, and a magnetometer. People's daily activities are calculated and recognized by analyzing one group of data obtained by the nine-axis sensor. The LoRa shield transmits datasets to the LoRa gateway and server while the Arduino board connects and controls the LoRa shield and sensors.

The system consists of three parts, the terminal part (LoRa nodes), gateway and server part, and application part. The wearable clothing with a multisensor integrated is regarded as the terminal part to collect daily activity datasets, which include walking, running, going upstairs or downstairs, and standing activities. The gateway and server part is used to transmit and save the datasets in the server. The application is utilized to analyze and show the data to users or managers.

In the terminal part, due to the limited transmission limitation in LoRa technology, this paper proposes a serial method to compress datasets, such as extracting real human activity datasets to form new features like max-min values and the mean and standard deviation to conclude the information of posture data waves to achieve the dimension reduction, using Random Forest algorithm to calculate the importance of features and select the first few important features LoRa nodes required. Then, the LoRa nodes are reset to collect selected features. In the experiment, six machine learning algorithms are utilized to extract the best posture recognition method. Finally, after training 3239 groups of datasets, the result proves that Random Forest performs best with an accuracy of 95.06%. Different users tested the system, and it functioned efficiently. Hence, this paper proves that multisensor posture recognition using IoT and LoRa is available and works well for long-term detection.

The remainder of this paper is organized as follows. Section 2 provides an overview of related work. Section 3 introduces the proposed LoRa posture recognition system based on multisensor. Section 4 evaluates the performance of the proposed system, and Section 5 concludes the paper.

## 2. Related Work

Wearable sensor design for the healthcare problem should be preventive, predictive, preemptive, and personalized. Advanced smart clothing not only promotes a healthy lifestyle but also allows the detection of any potential health risks and facilitates the implementation of preventive measures at an earlier stage [12]. Aging people who wear such clothes are effectively protected. In the proposed system, the IMU sensor is used in smart clothing. The IMU sensor provides the acceleration, angular velocity, and magnetometer values, which are analyzed to recognize human activities like walking, running, and even falling down [13]. Prathivadi et al.'s team did the research and proved that IMU has good performance on activity recognition [14].

In the healthcare system, the traditional data acquisition methods are Bluetooth, Wi-Fi, and the Internet. Bluetooth has a short distance disadvantage and a low power advantage [15]. Using Wi-Fi or the Internet on a smartphone to get datasets lifts the limit of distances [16], however, requiring high spending and high power. A smartphone is integrated with many devices and functions, which are the main source of power consumption [17], while not detecting the state of health. Besides, the user does not always move about with their smartphone. When using an application on a smartphone, people may experience some serious privacy issues [18].

Nowadays, more medical devices use IoT technology to collect data [19]. Presently, the main IoT technologies utilized in the market are NB-IoT (Narrow Band Internet of Things), ZigBee, Sigfox, and LoRa. NB-IoT communication is authorized to use the frequency band below 1 GHz, which charges companies' money. NB-IoT [20, 21] cannot provide the same battery life as LoRa due to the consideration of service quality. If an organization needs good service, NB-IoT is a good choice while the features of low cost and a large

number of connections are perfect for LoRa service. ZigBee [22, 23] has advanced features of low complexity, low power, and low price but is limited by its transmission range, which is only 10-75 meters. Sigfox [24] has an ultralong transmission distance of 50 km, but to ensure the energy-saving characteristics, the number of daily transmissions will be limited. In comparison with the IoT method, using Kinect [25] or camera [26] is preferred to identify the postures. With this kind of detection based on the image, a blind angle of vision appears, resulting in poor user experience. Using posture detection based on continuous signal data avoids space limitation and visual blind area. Compared to picture or video information, pure digital signal data is easier to encrypt and its representation harder to understand [27, 28]. On the other hand, signal data describes important information in its wave changes [29], which explains why a person stands up after a long time of sitting without the detector's warning. Because there is an average supervision time in the detector, this person feels tired even after sitting for a while. Then, the application adjusts to fit personal habits.

Considering the limitation of LoRa transmission, extracting significant information from multisensor data with a reasonable communication cost becomes a crucial problem. In the study of He et al. [30], their team proposed approximate aggregation to extract significant information. Peng proposed using Random Forest to optimize the characteristic high voltage signal of a PD cable to improve the accuracy and efficiency of PD pattern recognition and visualization of PD parameters [31]. The results show that the calculation of the importance of Random Forest to features is reliable. Hence, this system tries to use Random Forest to select the key features LoRa nodes require. SVM, KNN, and Random Forest have outstanding performance in the classification of human posture [32–34]. For the continuous signal data collected by LoRa nodes, a classifier with the highest accuracy is required to highly improve the accuracy of posture recognition. Due to the limitation of LoRa transmission and too much data compression, the optimal classifier is necessary. Facing current smart cyberphysical systems, the data compression should not only save uploading energy but guarantee privacy [35].

### 3. Posture Recognition System

**3.1. System Framework.** The LoRa-based human posture recognition system consists of four modules, posture information sensor module, wireless transmission module, human posture recognition module, and user interface module as color-coded in Figure 1. The terminal device, including the posture information sensor module and LoRa shield, collects the posture datasets and transmits them to the nearest LoRa gateway. After receiving a complete dataset package, the LoRa gateway delivers the datasets to the net server, which saves it. The Random Forest (RF) algorithm then analyzes the datasets and recognizes the postures. Then, the output results are as required by the users' operations.

The posture sensor module consists of an accelerometer, gyroscope, and magnetometer, which are utilized to periodically collect the posture datasets like running, walking, stand-

ing, jumping, and going upstairs or downstairs. The integrated posture information sensor module performs data denoising, data segmentation, and feature extraction automatically. In practice, it is equipped within the clothes as a wearable smart sensor as is shown in Figure 2.

The wireless transmission module is composed of a LoRa shield, gateway, and server. The terminal device combines multisensors and LoRa shield as a mobile detector to collect datasets and transmit the datasets to the LoRa gateway, which is a transfer station and delivers the datasets to the net server. This LoRa-based module can wirelessly transmit data from the terminal node to the server with high quality, low power consumption, and over a long distance. The transmitting process is based on a broadcast protocol, which ensures that users connect to the gateway wherever they are and the devices can upload the datasets automatically.

Before using the human posture recognition module shown in Figure 1, the red rectangle, the server, needs to set up the local recognition application. Because the magnetometer used in the terminal device is influenced by the local buildings and the location of the city, a trained module from other places will not work as efficiently as one from the local environment. Setting a local recognition application needs five steps: collecting local datasets, data preprocessing like denoising and smoothing, feature extraction, training new modules, and resetting the terminal device with the new feature extracted data format. After setting a local module, then when a group of new datasets comes in, users can get local results.

**3.2. Posture Information Sensor Module.** After analyzing some of the most popular human posture recognition products, the accelerometer, gyroscope, and magnetometer are selected to collect the human activity information. These sensors collect three-axis signal data of the  $x$ -,  $y$ -, and  $z$ -axes. In total, nine types of original signals were collected. The CPU is the Arduino Mega board.

These three sensors detect the values in real-time with a frequency of 25 Hz and have their monitoring value ranges. Every sensor has three axes,  $x$ -,  $y$ -, and  $z$ -axes. In total, the three sensors have nine axes and collect 25 groups of datasets, where every single dataset includes nine digital values, in one second. The Arduino Mega board calculates the max and min values, mean, and standard deviation of  $25 \times 5$  groups of datasets every 5 seconds. Limited by the signal bandwidth of LoRa, every transmitting interval is approximately 10 seconds. Therefore, every transmission includes two groups of max and min values, the mean, and standard deviation to represent what postures the user is in within these 10 seconds. The terminal node with sensors and Arduino Mega board should be fixed on the user's back or chest to collect points, which are more globally characterized than other limb placements.

**3.3. Wireless Transmission Module.** LoRa uses linear spread spectrum modulation technology, and its link budget allows the communication distance to be 15 km or higher. However, its signal is blocked by the buildings in cities so its distance usually is 1~2 km in cities. The LoRa network works in

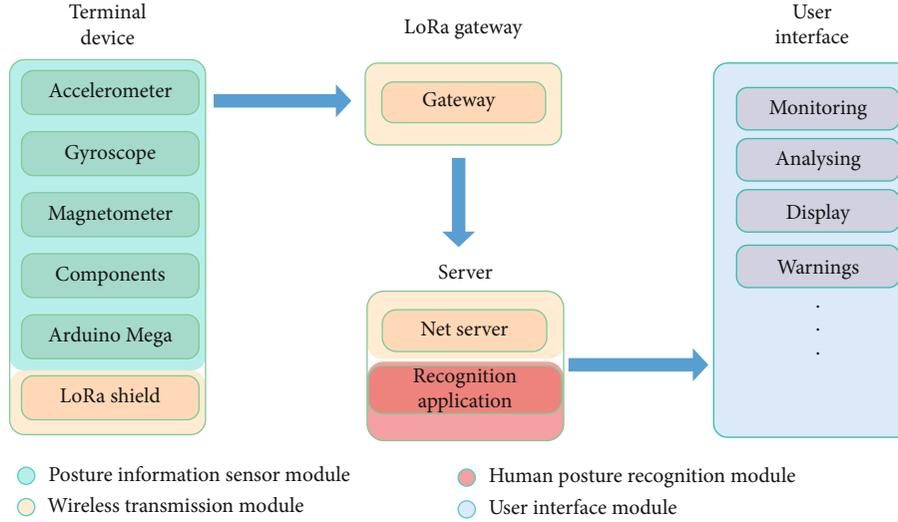


FIGURE 1: Four modules in LoRa posture recognition system.



FIGURE 2: LoRa wearable integrated sensors: (a) the position of a LoRa node on clothing; (b) person wearing the clothing; (c) the clothing inside view.

unlicensed bands such as 433 Hz, 868 Hz, and 915 Hz, and the infrastructure and operating costs of the LoRa WAN are thus low.

The wireless transmission module mainly consists of a LoRa node, LoRa gateway, and network server. The LoRa node, the terminal device, is mainly responsible for the transmission of electrical signals from the microcontroller. The LoRa gateway, as a transfer station, temporarily stores data and checks data integrity, then sends datasets to the network server. The LoRa WAN adopts a star network topology; that is, each node and gateway can communicate with each other in two directions. Broadcasting is utilized to achieve this requirement. Whatever the user's destination, the LoRa node always connects to the current nearest gateway instead of an exact gateway. This structure realizes the requirements of long-distance transmission and reduces the complexity of the network structure to reduce energy consumption and extend the internal battery life of nodes.

To ensure the feasibility of the star network, the LoRa gateway has the ability to deal with a large amount of information from each node. LoRa's high network capacity is achieved by using adaptive data rates and transceivers with multichannel and multiple modems in the gateway. Figure 3 illustrates the process of uploading datasets. Due to each group of data uploaded by broadcast protocol, a gate-

way may receive datasets from different nodes, which cause LoRa gateway itself to use different data receiving rates although the transmitting rate in each node is fixed. Because LoRa is modulation based on the spread spectrum, each signal is orthogonal. When the expansion factor changes, the effective data rate also changes. With this performance, the gateway can accept different data rates on the same channel simultaneously.

The most important factors to ensure that the gateway receives data at the maximum rate are the number of concurrent channels, data rate, load length, and the frequency of data transmission between nodes. Therefore, LoRa cannot transmit long data between nodes and gateway.

$$f(PL) = \max \left( \text{ceil} \left( \frac{PL + \text{Header} + \text{CRC} - 4SF - 20H}{4(SF - 2DE)} \right) \times CR, 0 \right). \quad (1)$$

PL represents the number of payload bytes, which is the posture dataset the system needs. Header is composed of preloaded information. CRC is used to check for cyclic redundancy. SF represents the spread-spectrum factor. When using the data header,  $H$  equals 0, otherwise,  $H$  equals 1. If the data transmission mode is set to low data rate

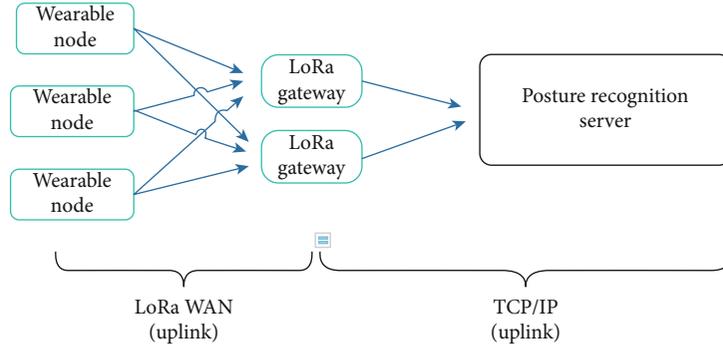


FIGURE 3: Structure of the LoRa wireless transmission module.

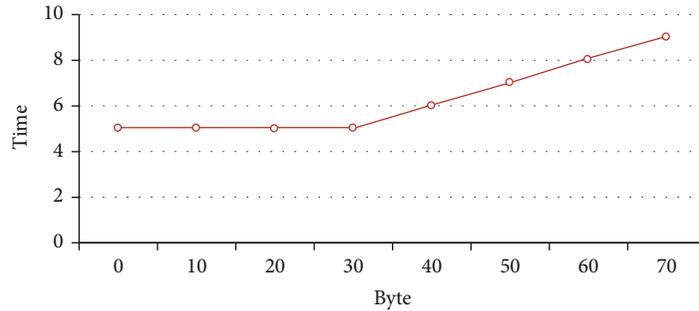


FIGURE 4: The change of time influenced by bytes in LoRa transmission.

optimization,  $DE = 1$ , otherwise,  $DE = 0$ . CR stands for coding rate, value is 1 to 4. The function  $\text{ceil}()$  returns a value that is bigger than the calculated result in parentheses or equals. Then, the max function is used to get the biggest value between 0 and  $\text{ceil}() \times CR$ . When other coefficients are fixed, PL becomes larger and then, the result,  $f(PL)$ , becomes larger, which means LoRa delivers more information in this transmission.

$$f(f(PL)) = T_{pre} + f(PL) \times T_{pay}, \quad (2)$$

$T_{pre}$  represents the time required for preamble transmission while  $T_{pay}$  is the selected transmitting mode. When  $f(PL)$  becomes larger and LoRa bandwidth is fixed, the overall transmitting time becomes larger. As posture datasets required, 25 groups of datasets are needed to be transmitted every second. LoRa also has itself min transmitting time, five seconds in Figure 4. What the system requires is that the whole posture datasets, 9-axis data times 25 frames each second and 5 seconds in total, should be compressed from 0 to 30 bytes. In the recognition module, a method using the Random Forest algorithm to catch key features of datasets will be explained.

Another way to compress information is to transfer the original data type into bytes. The received datasets are in the form of float type data, which takes up 4 bytes in Arduino Mega's memory. When transmitting to the LoRa gateway, the data type is transferred from float to string, which nearly doubles the size of the float.

TABLE 1: A comparison among the types of int, float, string, and char in Arduino Mega.

Data type	Int	Float	String	Char[2]
Size	4	4	8	2

As shown in Table 1, the char type takes the least memory when transmitting. Thus, Arduino Mega transfers float type data into  $\text{char}[2]$ , where  $\text{char}[0]$  represents the integer part while  $\text{char}[1]$  is the decimal part. In Table 2, an example is listed. At the beginning of a dataset package, a symbol position is added because char type only contains the values without positive or negative. Then, posture values are filled in after completing the dataset feature extraction. As for Arduino Mega board settings, every char type data consists of 8 bits. Symbol position is one char data and is composed of 8 bits representing positive or negative values of the sensors in Table 3.

In this module, the web server is responsible for managing all data originating from the gateway. All gateways communicate with the web server via a standard IP connection. If there are multiple gateways in one area, the network server automatically filters duplicate packets from different gateways to select the best quality packets. Besides, the web server verifies data security and sends confirmation characters to the gateway. Another core aspect of the web server is to periodically send data to the application server.

**3.4. Posture Recognition Module.** The posture recognition module is the core of this system and decides whether the

TABLE 2: Posture data format in Arduino Mega.

Symbol		Accelerometer—X					Gyroscope—mean			Magnetometer—Y			
F	E	0	0	1	2	0	0	D	1	0	0	f	2

TABLE 3: The format of positive and negative values.

1	0	0	1	1	1	0	0
+	—	—	+	+	+	Null	Null

system effectively recognizes postures. In Figure 5, what is required in this module is to do the data preprocessing, feature extraction, and feature selection; to train the module; and to improve the accuracy as much as possible.

**3.4.1. Data Preprocessing.** From the commercial posture detection products based on Bluetooth, we can see that the posture data is cyclical and smooth in Figure 6. In this system, the posture information sensor module is self-made so there are some errors in the measurement of the sensors. The errors are usually internal errors of the sensor and the jitter errors caused by wearing them on the human body, resulting in burrs, noises, etc. This kind of noise signal, which affects the amplitude of data fluctuation, will directly affect the subsequent feature extraction, classification, and recognition effect. At this time, the module needs to calibrate the sensor data in advance and denoise the collected original data.

The recursive average filtering method has a simple working mechanism and low computational complexity. The principle of the recursive average filtering method is to slide the sampling value step by step and then filter with the way of averaging the previous multiple samples. The effective sampling value is put into use and is obtained by the formula

$$y(i) = \sum_{n=0}^{m-1} \frac{x(i-n)}{m}, (i \geq n), \quad (3)$$

where  $x(i)$  is the real-time sensor signal data,  $y(i)$  is the recursive average value calculated at this time, and  $m$  is the number of sample values, which are randomly selected. It is found that when  $m = 5$ , the original fluctuation state of data is preserved, and the errors caused by burr and jitter are removed. Compared with when  $m = 3$ , the denoising effect is not obvious, and when  $m = 10$ , excessive denoising results in the elimination of some fluctuation features. The final value is 5.

Posture data is based on time serial and frames, and the transmission interval of LoRa technology is fixed. The datasets collected exclude the middle state values resulting in the above two factors. Thus, aiming at adding lost information, a mining value method, named sliding windows illustrated in Figure 7, is utilized for data preprocessing. The length and coverage of the sliding windows are two impor-

tant parameters. If the window is too small, a complete human posture is not captured. If the window is too large and contains many human body postures, the characteristics of a posture are weakened. Our experiment concluded that 3 frames and 5 frames are the two most suitable interval values as shown in Table 4. Utilizing varieties of recognition algorithms does not alter the results as the three-frame interval always provides the best accuracy. Due to the small amount of data in this experiment, the 3-frame value is used as the interval value for the experiment.

**3.4.2. Feature Extraction and Selection.** To further improve the frequency of human posture recognition and meet the conditions of the LoRa terminal node transmission, this study uses the feature selection algorithm of Random Forest to select a few samples of the most representative features out of 45 features extracted by the scientific analysis method for posture recognition training.

**(1) Feature Extraction.** In Figure 6, the features of various postures have certain differences, like the values of the peaks and troughs, dispersion degrees, max-min ranges, and means. In acceleration wave Figure 6, peaks and troughs of the waves represent the vigorous exercise, like jumping or running. In this case, only using this feature, the posture of stand and others can be easily distinguished.

$$\begin{aligned} x_{sa_{\max}} &= \max(x_{sa0}, x_{sa1} \cdots x_{sa125}), \\ x_{sa_{\min}} &= \min(x_{sa0}, x_{sa1} \cdots x_{sa125}). \end{aligned} \quad (4)$$

In this formula,  $x_{sa}$  represents sensor  $s$  and axis  $a$ . The data of every single axis has 25 values hence, in 5 seconds, 125 values. Therefore, the result is the max values of the 125 values from a single axis.

$$y = x_{sa_{\max}} - x_{sa_{\min}}, \quad (5)$$

where  $y$  represents the max range values minus min range values in a signal wave. In statistics, the range is often used to describe the dispersion degree of a group of data, as well as the variation range and dispersion range of variable distribution. The greater the range, the greater the degree of dispersion, while the smaller the range, the smaller the degree of dispersion.

$$\text{Mean}(x_{sa}) = \frac{\sum_{i=1}^n x_{sai}}{n}. \quad (6)$$

Mean describes the average values of the waveform.  $x_{sai}$  represents the data in a sliding window in formula (6). Then, these values in the same sliding window are summed and the summed value is divided by the total amount to get the mean. In Figure 6, the means of different features reach different

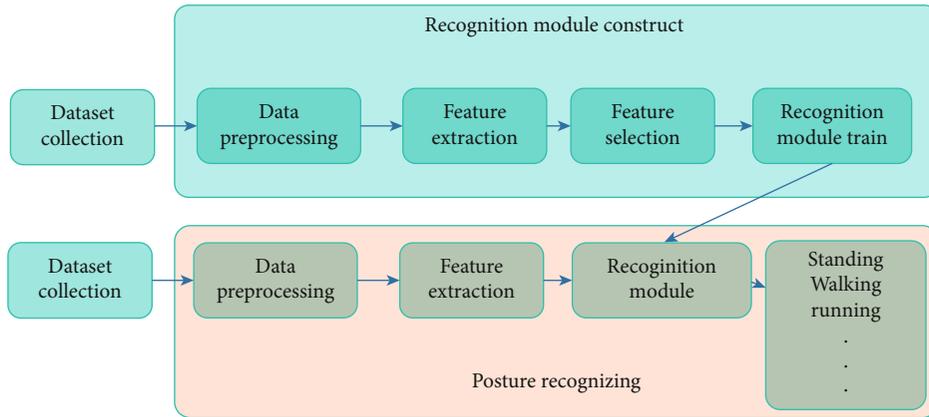


FIGURE 5: The processing of module construct and recognition in LoRa posture recognition module.

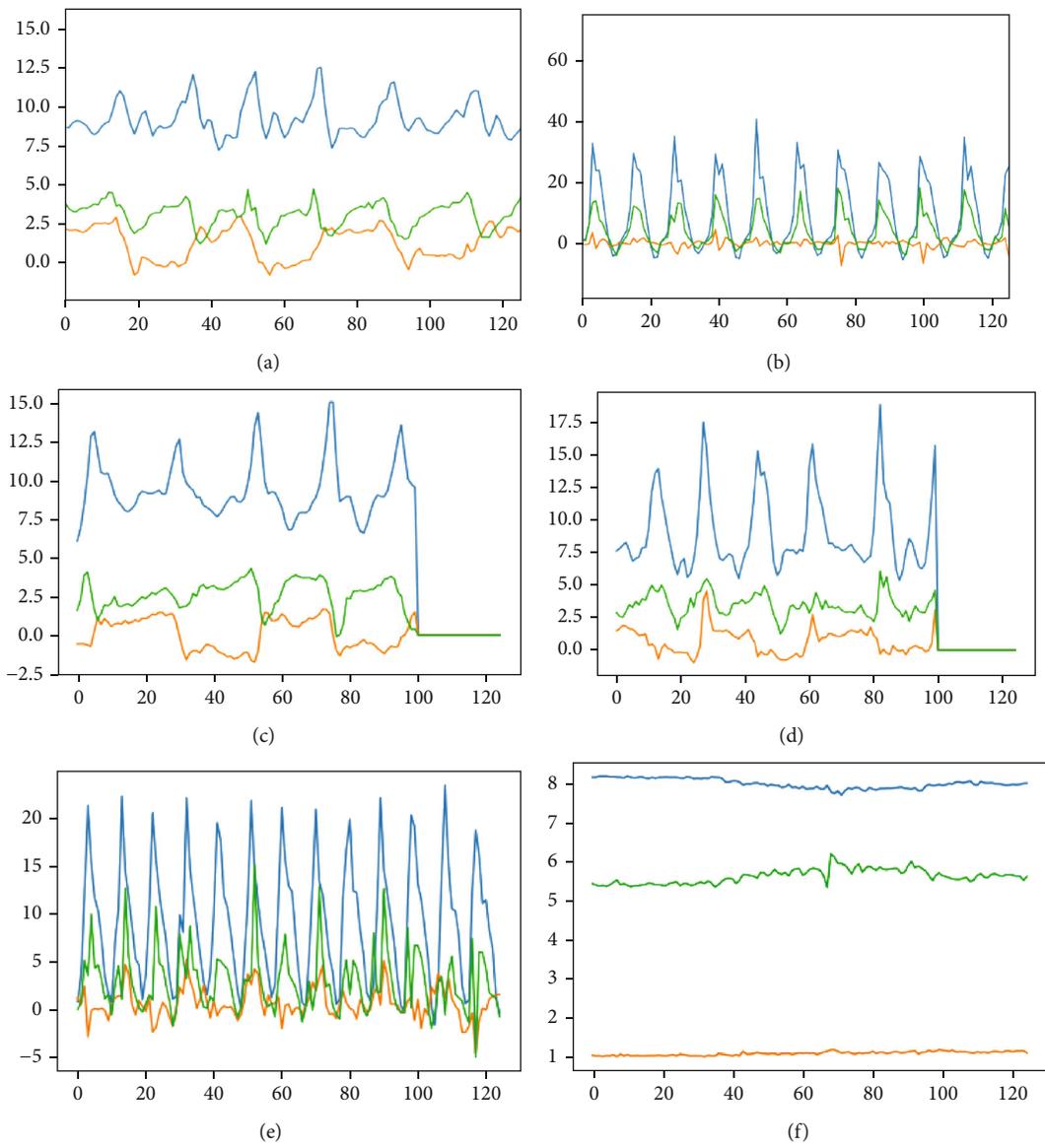


FIGURE 6: Acceleration triaxial signal graph of six postures: (a) walking; (b) jumping; (c) going upstairs; (d) going downstairs; (e) running; (f) standing.

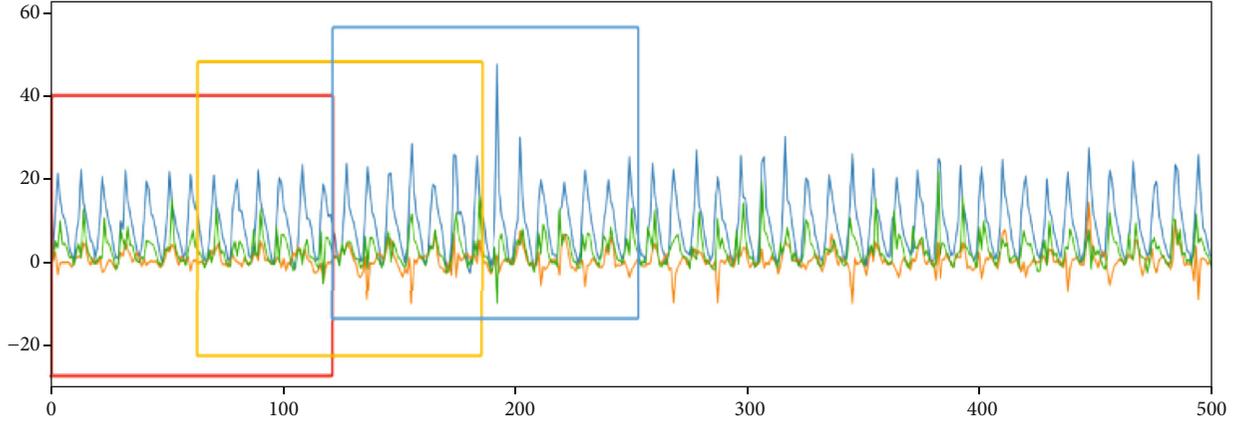


FIGURE 7: Posture data segmentation based on sliding windows.

TABLE 4: Comparison of accuracy with different interval time for sliding windows.

Interval	KNN	SVM	RF	NN	LR	Bayes
25 frames	0.66	0.83	0.85	0.82	0.49	0.75
15 frames	0.77	0.83	0.87	0.83	0.67	0.72
5 frames	0.88	0.94	0.92	0.82	0.64	0.82
3 frames	0.91	0.97	0.96	0.88	0.63	0.83

levels because these waves of postures are all nearly symmetrical and have their unique peak, with two features; a unique mean is generated and used as a significant feature in recognition.

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{\sum_{i=1}^n (x_{sai} - \text{mean}(x_{sa}))^2}{n}}, \quad (7)$$

where  $\sigma^2$  represents variance while  $\sigma$  represents the standard deviation. The standard deviation reflects the discrete degree of a data set. The standard deviation may not be the same for two groups of data with the same mean. Thus, to some extent, the standard deviation solves the problem, the same mean but different postures. Also, the standard deviation effectively calculates the degree of data dispersion. The standard deviation of a posture with a large range in the wave will be larger than that of a posture with a small range in the wave such as standing and speaking. Therefore, the standard deviation is used as one of the effective recognition features.

Every single axis of a sensor has 5 extracted features, maximum and minimum value, max-min range, mean, and the standard deviation. Therefore, one group dataset in five minutes has 45 features in total. Due to the limitation of LoRa transmission, what this system requires is to select the four most important features in these extracted features. In this study, four features are selected as the most important; however, in different areas and spaces, the importance may be altered by the environment and irresistible factors. Therefore, they are also useful even though not selected in this particular study.

(2) *Feature Selection.* Random Forest uses random resampling technology and node random splitting technology to generate multiple decision trees, and the final result is given by voting. The Random Forest calculates the importance of each variable in the training classification process, and this excellent property is open to development in many projects.

Every decision tree in the Random Forest algorithm uses the out-of-bag data method (an error estimation method that replaces the test set) to calculate the corresponding out-of-bag data error, which is recorded as erroob1. Random Forest randomly adds a noise interference to the features of all samples of out-of-bag data, which is used to randomly change the value of samples at feature  $x$ , and then calculates the error of out-of-bag data, which is recorded as erroob2. Supposing that there are  $n$  decision trees in a Random Forest algorithm, the importance of feature  $x$  is expressed as follows:

$$f(X) = \frac{1}{N} \sum (\text{erroob2} - \text{erroob1}). \quad (8)$$

This formula is used as a measure of the importance of variables. When random noise is added to a variable, the accuracy outside the bag decreases obviously; this shows that this feature has a great influence on the result of sample classification. In other words, its importance accounts for a relatively high proportion.

The feature selection is based on the importance of variables in the Random Forest data, and the dimension reduction is completed by selecting a series of data with high importance manually. When selecting features, we should pay attention to the selection of data groups with high correlation and importance, the selection of features with fewer data, and the selection of features that can maximize the reflection of prediction results. To generate the features, firstly, the characteristic variables in the Random Forest are sorted in a descending order or ascending order. Secondly, identify unnecessary features, remove data with particularly low correlation, and get new data groups. Then, establish a new Random Forest model with a new set of eigenvectors and calculate the importance of the features. Finally, repeat steps 1-3 until  $m$  features are left.

TABLE 5: Posture accuracy of different algorithms.

Algorithms	KNN	SVM	RandomForest	Neural network	Logistic regression	Bayes GaussianNB
Accuracy (%)	91.26	91.05	95.06	89.81	87.96	86.01

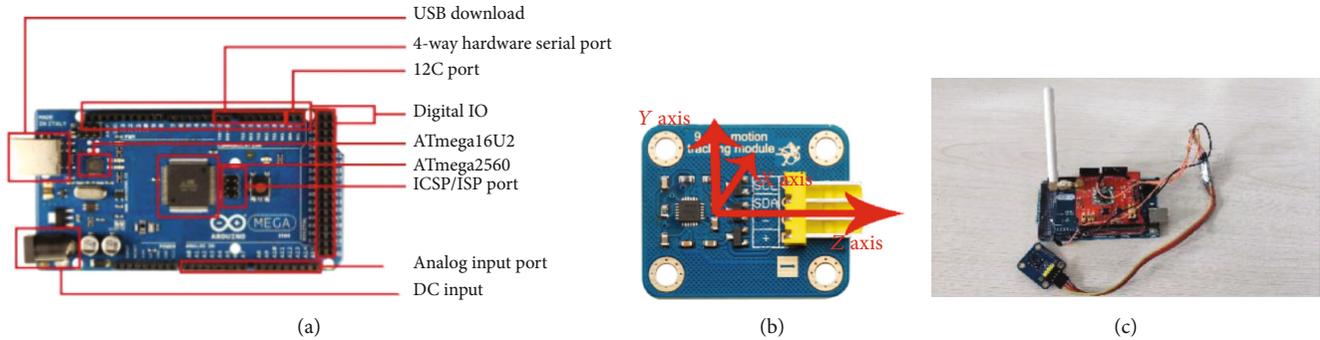


FIGURE 8: LoRa sensor and posture sensor introduction: (a) Arduino Mega 2560; (b) integrated sensors; (c) terminal equipment assembly.

After the above steps, the most important four features selected in this experiment are the max-min range of accelerometer  $z$ -axis, magnetometer  $y$ -axis, the standard deviation of the gyroscope  $x$ -axis, and the max-min range of magnetometer  $x$ -axis.

**3.4.3. Posture Recognition.** In the posture recognition module, multisource sensor information was used to recognize human postures. To guarantee a high level of accuracy as possible, the best classifier is required. In the study, 6 machine learning methods are utilized to train the four selected features and the accuracy is shown in Table 5. Hence, the Random Forest algorithm is selected as the classifier.

Random Forest is a supervised learning algorithm. From the name of Random Forest, one knows that it is a classifier composed of multiple decision trees. The essence of this classifier is based on the idea of ensemble learning in machine learning, which combines several small classifier models to form a larger classifier to make prediction and recognition. Each small classifier is independent of the other and studies and tests the samples independently. Finally, the classification results of multiple small classifiers are combined to form a large single classification prediction output. From the principle algorithm, the final result is better than each small individual classifier.

*(1) Generation of the Training Set.* The algorithms to realize the decision tree include ID3, C4.5, and CART algorithms. Among them, ID3 chooses information gain value as the evaluation standard of the split attribute, because information gain tends to rely on multivalued characteristic attributes in the calculation process, which makes the result biased and easy to fall into local optimum. Based on this problem, C4.5 improves. It changes the evaluation standard of attribute splitting to information gain rate and eliminates the resulting bias caused by quantity. In the CART [36] algorithm, the Gini index is used as the evaluation standard of attribute splitting, and the attribute with the least impurity is selected.

Besides, another feature of CART is to use dichotomy to form a binary tree for tree structure, and there are only two values of yes or no for feature division. This design discretizes the continuous attributes, making it possible to deal with the continuous attributes and achieve data regression prediction. Wholly, in addition to the evaluation criteria and tree structure mentioned above, the framework and mode of cart decision tree and common decision tree are consistent, that is, the three processes of attribute selection based on the evaluation, top-down recursive spanning tree structure, and decision tree pruning.

In general, the same sample can only build a decision tree. If we want to use the same batch of data samples to build multiple decision trees, we need to select multiple samples by way of resampling data and replacing them. Each selected sample can build a cart decision tree, and resample can build a cart decision tree classifier. The set of these classifiers constitutes a Random Forest. If the sampling method without replacement is adopted, the training samples used by each tree will be completely different, which will lead to a certain degree of deviation for each decision tree, and the voting mechanism of the subsequent Random Forest is the process of seeking the same, which will not be conducive for the construction of the model.

The randomness of Random Forest is reflected not only in the randomness of sample selection but also in the randomness of feature selection. Randomly selected features from all attributes of samples are used for the construction of the classifier. The purpose of this operation is to reduce the correlation between the decision trees. The lower the correlation, the lower the error rate.

*(2) Generation of Decision Tree.* The sampling method in the first step is used to generate training set samples and train each decision tree model. Decision trees in Random Forest do not particularly need to be pruned. The training samples of the decision tree are filtered by random samples and random features, which makes the decision tree unlikely to overfit in the construction process.

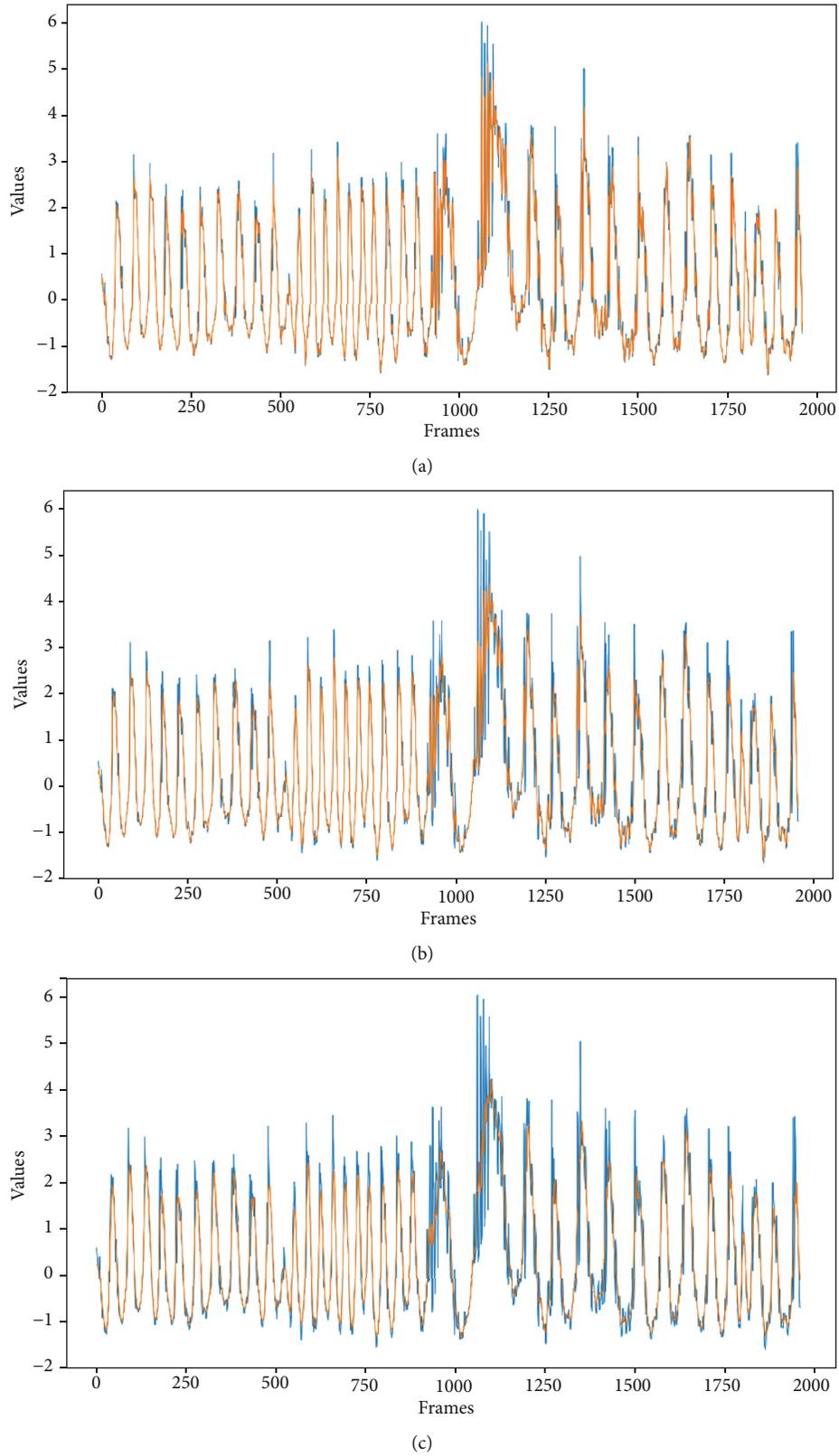


FIGURE 9: Different effects of the recursive average filtering when  $m = 3, 5, 10$ : (a)  $m = 3$ ; (b)  $m = 5$ ; (c)  $m = 10$ .

(3) *Formation of RF.* After training multiple decision trees, the classification results of the decision trees are summarized and the prediction results of the final model are selected by

voting rules. Common voting rules include a one-vote veto system, a minority subject to the majority, and a weighted majority. Here, minority subject to the majority is used.

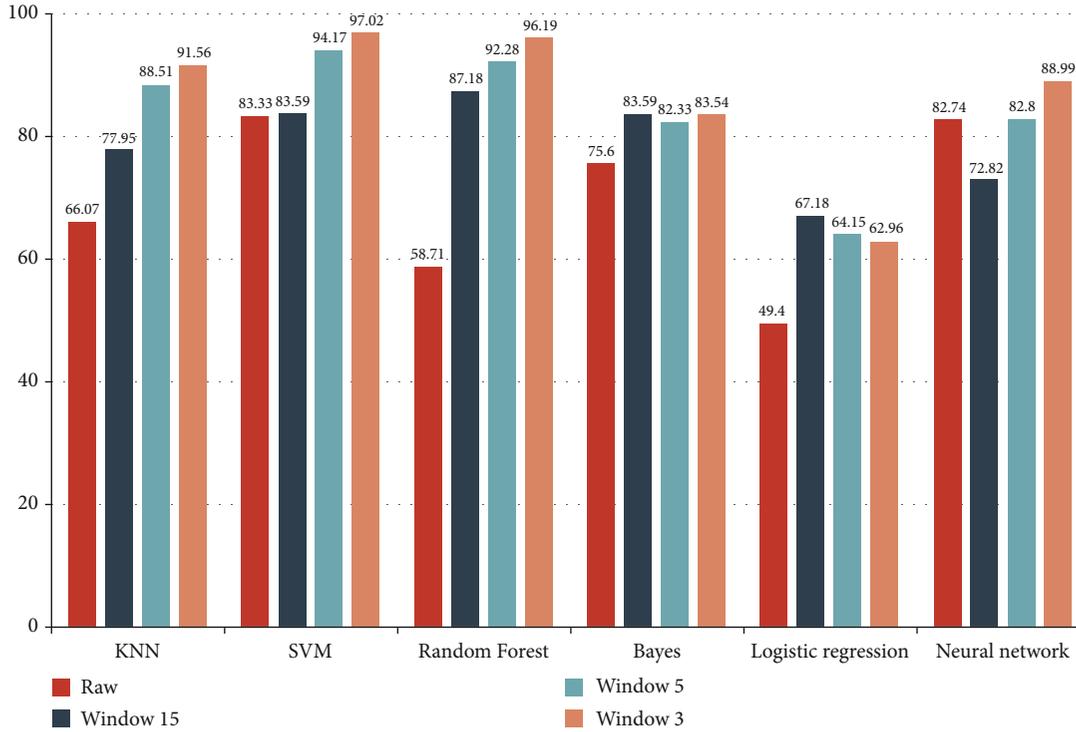


FIGURE 10: Accuracy of different cases: raw data, sliding window  $m = 15, 5$ , and  $3$ , respectively.

#### 4. Experiments and Analysis Conclusions

In the experimental part, the parameters of the LoRa nodes and gateway are given in detail. The key results of data denoising, feature extraction, and feature selection are displayed in the figures with different parameters. After a comparative analysis, the best classification algorithm is selected as the main recognition method in this system.

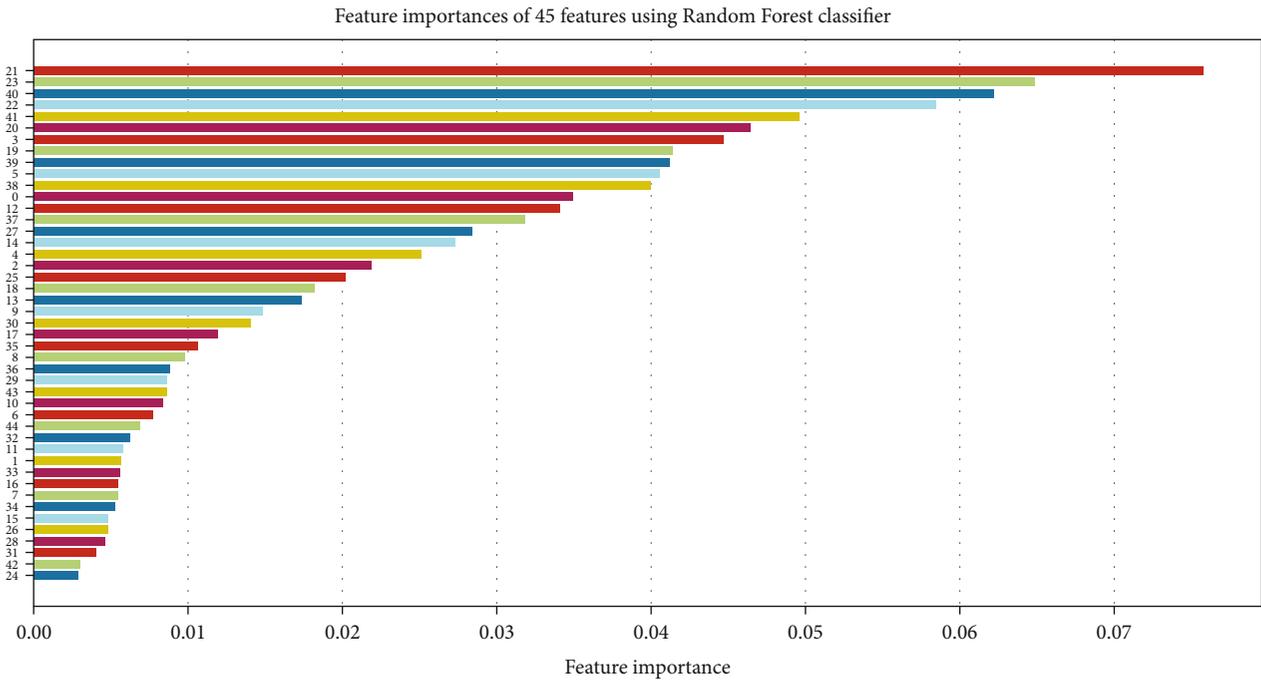
Figure 8(a) shows the integrated sensors with the accelerometer, gyroscopes, and magnetometers, which detect three-axis data, respectively. Their detecting frequency is 25 Hz. Figure 8(b) shows the Arduino Mega. In the sensor data acquisition module, Arduino Mega 2560 is selected as the microprocessor, which has high performance and low power consumption characteristics. Arduino Mega 2560 is a master development board based on ATmega2560. In addition to providing serial port data transmission, this processor supports Wi-Fi, Lora, and other wireless transmission modules, and users can select the appropriate transmission mode with the use of scenarios. In Figure 8(c), the red board is the LoRa shield, which is utilized for gateway connection and transmission. The LoRa shield chip parameters are as follows: equipment size  $62 \times 43 \times 23$  mm, equipment weight 22 g, package size  $11 \times 70 \times 36$  mm, and package weight 53 g. Lora shield is compatible with 3.3 V or 5 V I/O performance. Its frequency band is 915 MHz/868MHz/433 MHz three different optional intervals. This frequency band is factory configured. LoRa shield boards are compatible with Arduino Leonardo or Uno or Mega or due. LoRa shield’s signal transmission is through I-Pex external connector antenna for data

exchange and has the characteristics of low power consumption. Three parts are assembled to form the terminal equipment as a LoRa node.

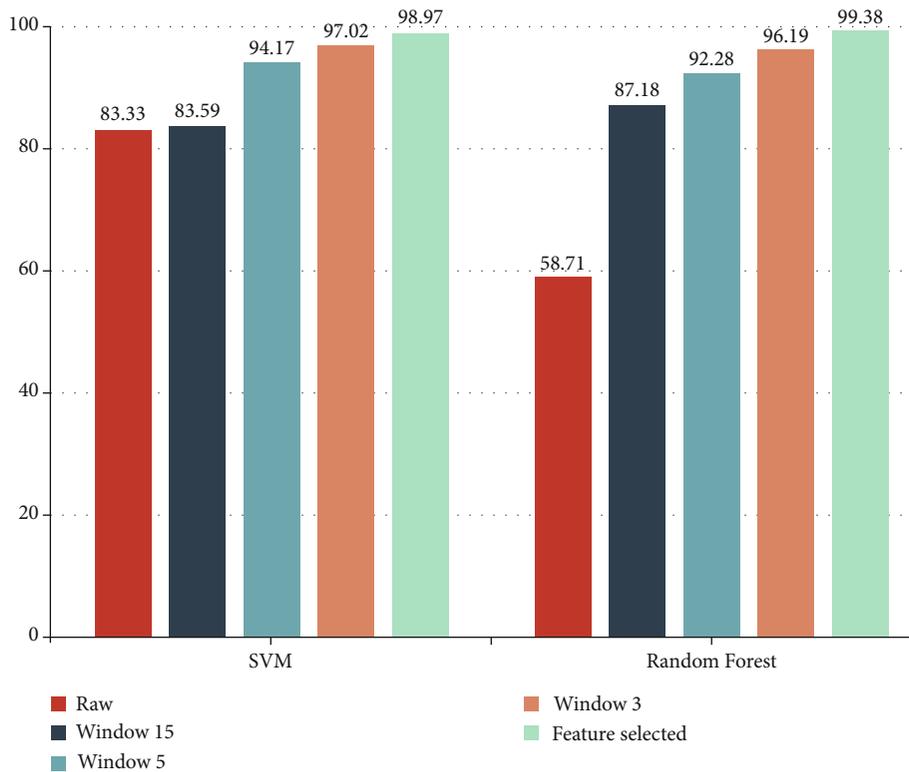
Due to the limitation of LoRa transmission, the posture datasets cannot be wirelessly transmitted directly. The terminal device is connected by PC to collect the raw datasets to go on the next analysis. In the part of collecting raw datasets, the sample rate is 25 frames per second. These raw datasets are saved in different CSV files according to the different types of postures.

Raw datasets have some certain noises, which are caused by the sensor itself or the environment. To guarantee the accuracy of the trained module, data denoising is required. The recursive average filtering method is utilized to smooth the waves. Figure 9 shows the effects of the recursive average filtering when  $m = 3, m = 5$ , and  $m = 10$ , respectively. It is clear that when  $m = 3$ , there are still some sharp changes and noise remaining. When  $m = 10$ , it is too smooth that it loses some key information like real peak values. When  $m = 5$ , it neither devalues the peaks nor retains noise. Therefore, all posture datasets are processed with  $m = 5$  and resaved in CSV files.

The raw datasets are based on frame sampling with a sampling frequency of 25 frames per second. This operation will cause some information between two consecutive samples to be lost. Hence, using the sliding window method enlarges datasets and fills in the lost posture information. Initially, raw datasets are 500 groups of posture datasets. After using the sliding window where the sliding step is 15, there are 647 groups of datasets. When the sliding window is 5,



(a)



(b)

FIGURE 11: (a) Importance of 45 features. (b) Accuracy including the selected feature.

there are 1942 groups of datasets, while when the sliding window is 3, there are 80975 lines of datasets, which are 3239 groups of datasets.

Figure 10 shows the accuracy of raw data and window = 15, 5, and 3, respectively. There are six learning algorithms,

KNN, SVM, Random Forest, Bayes, logistic regression, and neural network. Every algorithm has four histograms representing the accuracy of raw data and window = 15, 5, and 3. As shown, when the sliding window step gradually decreases and the amount of samples increases, the accuracy of most

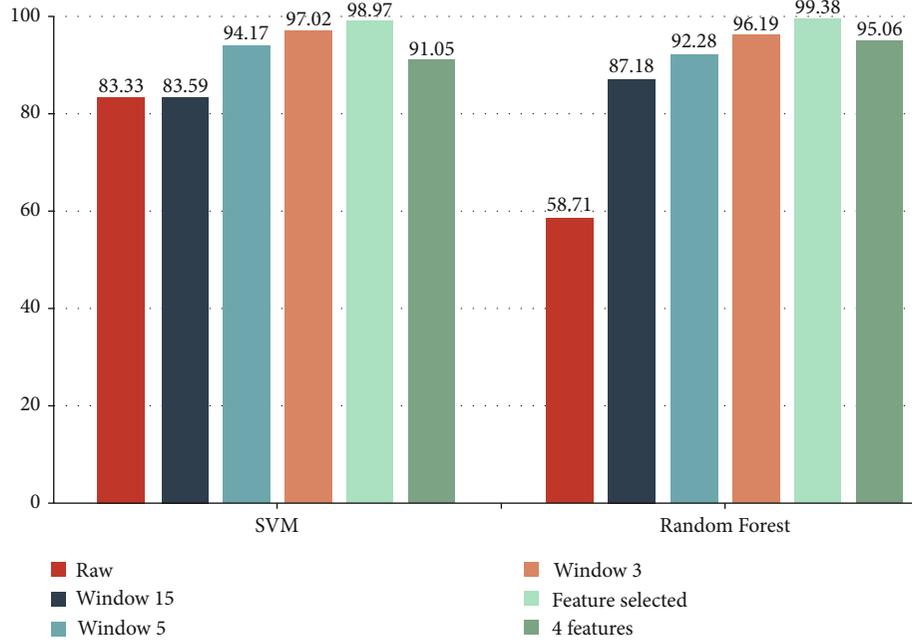


FIGURE 12: A comparison of the recognition accuracy among using 4 extracted features and the others.

algorithms greatly improves, even when the sliding window = 3, and the accuracy of SVM and Random Forest can reach 97.02% and 96.19%, respectively, which is approximately a 15% and 38% increase in comparison with the accuracy of raw data.

Up to now, the accuracy is improved and the sliding window is proved successful. However, LoRa transmission still requires a small data size. Hence, feature extraction and selection should be implemented.

All axes, the 5 features included, max values, min values, max-min range, mean, and standard deviation, are extracted as new input datasets. In total, features have 45 types generated from 9 axes. According to the equipment in this experiment, we used Random Forest to calculate the importance and selected the most important 4 features to meet the LoRa basic requirement. Figure 11(a) shows the descending order of 45 features. As seen, the 21st, 23rd, 40th, and 22nd features get the highest scores. These 4 features represent the max-min range of the accelerometer,  $z$ -axis; magnetometer,  $y$ -axis; the standard deviation of the gyroscope,  $x$ -axis; and the max-min range of the magnetometer,  $x$ -axis.

Figure 11(b) shows the best two learning algorithms with 5 processed datasets, where the last green histogram is selected feature datasets. This figure shows feature extraction and selection which more effectively conclude the information of datasets, and when these are applied in learning, the accuracy is directly improved. The raw data only tells what sample, pattern, or wave the data is, while extracted features tell what is behind the data, like whether it is symmetrical movement or violent movement.

When put into the learning module, the datasets of 4 selected features show slightly lowered accuracy in Figure 12. Due to the information lost when giving up the other 41 features, this case is understood and accepted. The

accuracy, 91.05% and 95.06% in SVM and Random Forest, also represents high correctness, and they prove that we selected the right features. Finally, due to the better performance of the Random Forest algorithm, it is selected as the final recognition module for the overall system structure.

According to previous experiments, the key features are found. Then, the terminal device, the LoRa node, is required to reset the transmission format and data values to meet the transmission basis. Now, the process of setting up the human posture module is finished.

To determine the accuracy and strength of the system, we tested the different human postures as follows. We collected 500 hundred groups of datasets, 100 datasets per posture, from 3 different people as shown in Table 6. The posture stand has perfect accuracy of 100% recognition. Going upstairs and downstairs is likely to be mistaken as walking while walking and running are likely to be mistaken as going upstairs and downstairs. Although there are some small errors, the recognition module works well when experiencing other posture datasets.

## 5. Conclusion

This paper presents a LoRa posture recognition system based on multisensor fusion. This paper describes how and why human posture recognition is important for a smart city. Human posture recognition helps people monitor their health and collect information for scientists to develop healthy exercise method and find the secret between health and daily postures. The system includes sensor modules based on accelerometer, gyroscope, and magnetometer to collect posture information. It uses LoRa wireless network technology to transmit sensor data to the network server. After receiving the data from the cloud server, the application

TABLE 6: Confusion matrix of the Random Forest model.

	Stand	Upstairs	Down-stairs	Walk	Run	Jump
Stand	100	0	0	0	0	0
Upstairs	0	86	6	8	0	0
Downstairs	0	4	92	4	0	0
Walk	0	1	2	96	1	0
Run	0	3	2	0	95	0

server recognizes and classifies the human posture. This paper proposes how to preprocess the frame-based posture data, extract features, and select features. It includes data denoising based on recursive average filtering and data stream segmentation based on the sliding window. In 5 seconds, 45 key features are extracted and the first 4 key features are selected to reset the terminal equipment. The experimental results show that the recognition rate of the four features is 95.06%. This achieves the goal of using a small number of features to maintain high classification accuracy. In future work, we will collect additional postures as a training set for further postural recognition. But the extra postures may need more features because it will not get high accuracy with only four features. Hence, we are actively looking for a stronger LoRa gateway.

### Data Availability

The data can be available upon request to the corresponding author.

### Conflicts of Interest

The authors declare no conflict of interest.

### Acknowledgments

This research was funded by the MSIT (Ministry of Science, ICT), Korea, under the High-Potential Individuals Global Training Program (2019-0-01585) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), National Nature Science Foundation of China (No. 61503005), the Great Wall Scholar Program (CIT&TCD20190304, CIT&TCD20190305), and "Yuyou" Project of North China University of Technology.

### References

- [1] J. Wang, Y. Yang, T. Wang, R. S. Sherratt, and J. Zhang, "Big data service architecture: a survey," *Journal of Internet Technology*, vol. 21, no. 2, pp. 393–405, 2020.
- [2] S. Kumar and M. Singh, "Big data analytics for healthcare industry: impact, applications, and tools," *Big Data Mining and Analytics*, vol. 2, no. 1, pp. 48–57, 2019.
- [3] B. Alkouz, Z. A. Aghbari, and J. H. Abawajy, "Tweetluenza: predicting flu trends from twitter data," *Big Data Mining and Analytics*, vol. 2, no. 4, pp. 273–287, 2019.
- [4] Y. Yu, M. Li, L. Liu, Y. Li, and J. Wang, "Clinical big data and deep learning: applications, challenges, and future outlooks," *Big Data Mining and Analytics*, vol. 2, no. 4, pp. 288–305, 2019.
- [5] S. Z. Reza, M. T. Hamidreza, and R. Rouhollah, "Smart-phone-centric human posture monitoring system," in *2014 IEEE Canada International Humanitarian Technology Conference - (IHTC)*, pp. 1–4, Montreal, QC, Canada, 2014.
- [6] Q. Dang, J. Yin, B. Wang, and W. Zheng, "Deep learning based 2D human pose estimation: a survey," *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 663–676, 2019.
- [7] A. Zourmand, A. L. K. Hing, C. W. Hung, and M. AbdulRehman, "Internet of Things (IoT) using LoRa technology," in *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pp. 324–330, Selangor, Malaysia, 2019.
- [8] A. Lavric and V. Popa, "Internet of Things and LoRa™ low-power wide-area networks challenges," in *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–4, Targoviste, Romania, 2017.
- [9] Y. Tian, T. Li, W. Song, S. Fong, L. Song, and J. Han, "Smart power management Internet of Things system with 5G and LoRa hybrid wireless network," in *5G-Enabled Internet of Things*, p. 13, Taylor & Francis Group, 2019.
- [10] V. A. Dambal, S. Mohadikar, A. Kumbhar, and I. Guvenc, "Improving LoRa signal coverage in urban and sub-urban environments with UAVs," in *2019 International Workshop on Antenna Technology (iWAT)*, pp. 210–213, Miami, FL, USA, 2019.
- [11] M. Sun, Y. Jiang, Q. Liu, and X. Liu, "An auto-calibration approach to robust and secure usage of accelerometers for human motion analysis in FES therapies," *Computers, Materials & Continua*, vol. 60, no. 1, pp. 67–83, 2019.
- [12] Y.-L. Zheng, X. R. Ding, C. C. Y. Poon et al., "Unobtrusive sensing and wearable devices for health informatics," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 5, pp. 1538–1554, 2014.
- [13] Y. Yan and Y. Ou, "Accurate fall detection by nine-axis IMU sensor," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 854–859, Macau, China, 2017.
- [14] Y. Prathivadi, J. Wu, T. R. Bennett, and R. Jafari, "Robust activity recognition using wearable IMU sensors," in *IEEE SEN-SORS 2014 Proceedings*, pp. 486–489, Valencia, Spain, 2014.
- [15] X. Shen, W. Cheng, and M. Lu, "Wireless sensor networks for resources tracking at building construction sites," *Tsinghua Science and Technology*, vol. 13, no. S1, pp. 78–83, 2008.
- [16] X. Su, H. Tong, and P. Ji, "Activity recognition with smart-phone sensors," *Tsinghua Science and Technology*, vol. 19, no. 3, pp. 235–249, 2014.
- [17] E. Ahmadoh and L. A. Tawalbeh, "Power consumption experimental analysis in smart phones," in *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 295–299, Barcelona, Spain, 2018.
- [18] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network Magazine*, vol. 32, no. 4, pp. 8–14, 2018.
- [19] Z. U. Ahmed, M. G. Mortuza, M. J. Uddin, M. H. Kabir, M. Mahiuddin, and J. Hoque, "Internet of Things based patient health monitoring system using wearable biomedical device," in *2018 International Conference on Innovation in Engineering and Technology (ICIET)*, pp. 1–5, Dhaka, Bangladesh, 2018.

- [20] Y. Lin, H. Tseng, Y. Lin, and L. J. Chen, "NB-IoTtalk: a service platform for fast development of NB-IoT applications," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 928–939, 2019.
- [21] N. Mangalvedhe, R. Ratasuk, and A. Ghosh, "NB-IoT deployment study for low power wide area cellular IoT," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–6, Valencia, Spain, 2016.
- [22] T. Elarabi, V. Deep, and C. K. Rai, "Design and simulation of state-of-art ZigBee transmitter for IoT wireless devices," in *2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 297–300, Abu Dhabi, UAE, 2015.
- [23] M. Zhang and Q. Hu, "A hybrid network smart home based on Zigbee and smart plugs," in *2017 7th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 389–392, Nagpur, India, 2017.
- [24] N. I. Osman and E. B. Abbas, "Simulation and modelling of LoRa and Sigfox low power wide area network technologies," in *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pp. 1–5, Khartoum, Sudan, 2018.
- [25] H. P. H. Shum, E. S. L. Ho, Y. Jiang, and S. Takagi, "Real-time posture reconstruction for Microsoft Kinect," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1357–1369, 2013.
- [26] D. Brulin, Y. Benezeth, and E. Courtial, "Posture recognition based on fuzzy logic for home monitoring of the elderly," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 5, pp. 974–982, 2012.
- [27] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart internet of things systems: a consideration from a privacy perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.
- [28] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.
- [29] S. Cheng, Z. Cai, and J. Li, "Curve query processing in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5198–5209, 2015.
- [30] Z. He, Z. Cai, S. Cheng, and X. Wang, "Approximate aggregation for tracking quantiles and range countings in wireless sensor networks," *Theoretical Computer Science*, vol. 607, pp. 381–390, 2015.
- [31] X. Peng, J. Li, G. Wang et al., "Random forest based optimal feature selection for partial discharge pattern recognition in HV cables," *IEEE Transactions on Power Delivery*, vol. 34, no. 4, pp. 1715–1724, 2019.
- [32] H. J. Lee, S. H. Hwang, S. M. Lee, Y. G. Lim, and K. S. Park, "Estimation of body postures on bed using unconstrained ECG measurements," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 6, pp. 985–993, 2013.
- [33] J. E. Estrada and L. A. Vea, "Real-time human sitting posture detection using mobile devices," in *2016 IEEE Region 10 Symposium (TENSymp)*, pp. 140–144, Bali, Indonesia, 2016.
- [34] C. H. Zhao, B. L. Zhang, J. He, and J. Lian, "Recognition of driving postures by contourlet transform and random forests," *IET Intelligent Transport Systems*, vol. 6, no. 2, pp. 161–168, 2012.
- [35] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.
- [36] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.