International Journal of Intelligent Systems

A Fully-Connected Deep Learning Approach to Upper Limb Gesture Recognition in a Secure FES Rehabilitation Environment

--Manuscript Draft--

Manuscript Number:	INT2.20200770R2
Full Title:	A Fully-Connected Deep Learning Approach to Upper Limb Gesture Recognition in a Secure FES Rehabilitation Environment
Article Type:	Research Article
Keywords:	Upper Limb Rehabilitation; Functional Electrical Stimulation; Fully Connected Neural Network; Gesture Recognition; Multi-Sensor Fusion; Security and Safety
Corresponding Author:	Qi Liu School of Computer & Software, Nanjing University of Information Science & Technology CHINA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	School of Computer & Software, Nanjing University of Information Science & Technology
Corresponding Author's Secondary Institution:	
First Author:	Qi Liu
First Author Secondary Information:	
Order of Authors:	Qi Liu
	Xueyan Wu
	Yinghang Jiang
	Xiaodong Liu
	Yonghong Zhang
	Xiaolong Xu
	Lianyong Qi
Order of Authors Secondary Information:	
Abstract:	Stroke is one of the leading causes of death and disability in the world. The rehabilitation of Patients' limb functions has great medical value, e.g. the therapy of FES (Functional Electrical Stimulation) systems, but suffers from effective rehabilitation evaluation. In this paper, six gestures of upper limb rehabilitation were monitored and collected using MEMS sensors, where data stability was guaranteed using data pre-processing methods, i.e. de-weighting, interpolation, and feature extraction. A fully connected neural network has been proposed investigating the effects of different hidden layers, and determining its activation functions and optimizers. Experiments have depicted that a 3-hidden-layer model with a softmax function and an adaptive gradient descent optimizer can reach an average gesture recognition rate of 97.19%. A stop mechanism has been used via recognition of dangerous gesture to ensure the safety of system, and the light-weight cryptography has been used via hash to ensure the security of system. Comparison to the classification models, e.g. k-NN, Logistic Regression and other random gradient descent algorithms was conducted to verify the outperformance in recognition of upper limb gesture data. This work also provides an approach to creating health profiles based on large-scale rehabilitation data and therefore consequent diagnosis of the effects of FES rehabilitation.
Additional Information:	
Question	**Response**

Please submit a plain text version of your cover letter here.	Cover Letter Dear Editors of International Journal of Intelligent Systems We are pleased to submit the revised paper entitled "A Fully-Connected Deep Learning Approach to Upper Limb Gesture Recognition in a Secure FES Rehabilitation Environment" (Manuscript ID: INT2.20200770R2) by Qi Liu, Xueyan Wu, Yinghang Jiang, Xiaodong Liu, Yonghong Zhang, Xiaolong Xu and Lianyong Qi for the consideration of publication in International Journal of Intelligent Systems. All the comments from reviewers have been responded with corresponding revision. Point-by-point responses to the reviewers' comments have been filled in the "Response to Reviewers" box. A Response-to-Comments file has been uploaded additionally. Sorry for missing the "Response to Reviewers" box in the previous revision submission. If any further information is needed, please let us know. Thank you. Yours Faithfully, Qi Liu, Xueyan Wu, Yinghang Jiang, Xiaodong Liu, Yonghong Zhang, Xiaolong Xu, Lianyong Qi 29 Jan. 2021
Do you or any of your co-authors have a conflict of interest to declare?	No. The authors declare no conflict of interest.

Title Page

A Fully-Connected Deep Learning Approach to Upper Limb Gesture Recognition in a Secure FES Rehabilitation Environment

Qi Liu[a,#], Xueyan Wu[b,#], Yinghang Jiang[b], Xiaodong Liu[c], Yonghong Zhang[d], Xiaolong Xu[a], Lianyong Qi[e,]*

[a]School of Computer & Software, Nanjing University of Information Science & Technology, 210044, Nanjing, China
[b]Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science & Technology, Nanjing, 210044
[c]School of Computing, Edinburgh Napier University, 10 Colinton Road, Edinburgh EH10 5DT, UK.
[d]School of Automation, Nanjing University of Information Science & Technology, Nanjing, Jiangsu 210044, China
[e]School of Computer, Qufu Normal University, Shandong 273199, China

A Fully-Connected Deep Learning Approach to Upper Limb Gesture Recognition in a Secure FES Rehabilitation Environment

Qi Liu[a,#], *Xueyan Wu*[b,#], *Yinghang Jiang*[b], *Xiaodong Liu*[c], *Yonghong Zhang*[d], *Xiaolong Xu*[a], *Lianyong Qi*[e*]

[a]*School of Computer & Software, Nanjing University of Information Science & Technology, 210044, Nanjing, China*
[b]*Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science & Technology, Nanjing, 210044*
[c]*School of Computing, Edinburgh Napier University, 10 Colinton Road, Edinburgh EH10 5DT, UK.*
[d]*School of Automation, Nanjing University of Information Science & Technology, Nanjing, Jiangsu 210044, China*
[e]*School of Computer, Qufu Normal University, Shandong 273199, China*

Abstract

Stroke is one of the leading causes of death and disability in the world. The rehabilitation of Patients' limb functions has great medical value, e.g. the therapy of FES (Functional Electrical Stimulation) systems, but suffers from effective rehabilitation evaluation. In this paper, six gestures of upper limb rehabilitation were monitored and collected using MEMS sensors, where data stability was guaranteed using data pre-processing methods, i.e. de-weighting, interpolation, and feature extraction. A fully connected neural network has been proposed investigating the effects of different hidden layers, and determining its activation functions and optimizers. Experiments have depicted that a 3-hidden-layer model with a softmax function and an adaptive gradient descent optimizer can reach an average gesture recognition rate of 97.19%. A stop mechanism has been used via recognition of dangerous gesture to ensure the safety of system, and the light-weight cryptography has been used via hash to ensure the security of system. Comparison to the classification models, e.g. k-NN, Logistic Regression and other random gradient descent algorithms was conducted to verify the outperformance in recognition of upper limb gesture data. This work also provides an approach to creating health profiles based on large-scale rehabilitation data and therefore consequent diagnosis of the effects of FES rehabilitation.

Keywords: Upper Limb Rehabilitation, Functional Electrical Stimulation, Fully Connected Neural Network, Gesture Recognition, Multi-Sensor Fusion, Security and Safety

1. Introduction

Stroke has become the second leading cause of death and the third leading cause of disability worldwide[11]. Post-stroke rehabilitation is all measures to prevent the onset and mitigate the effects of disability so that stroke patients can minimize neurological deficits, prevent complications, and improve their daily living skills[23]. Depending on the location of the brain injury, post-stroke neurological deficits include sensory and motor dysfunction, swallowing impairment, impairment in activities of daily living (ADL) and speech communication impairment, of which approximately 60% of patients will have distal fine functional impairment of the upper extremities[3]. Currently, the main treatment options include traditional physical therapy, electromyographic feedback therapy, mirror therapy, movement restriction therapy, rehabilitation play therapy, teletherapy, robot-assisted therapy, and functional electrical stimulation (FES) therapy. FES uses low-frequency electrical currents worn on the limbs to stimulate the muscles that have lost their innervation, increase the contraction capacity of the muscles, promote coordination between the antagonistic muscle groups and the diastolic capacity of the active muscle groups, and enable the limbs to perform various types of rehabilitation exercises to restore the function of the limbs, which is currently the most widely used rehabilitation therapy. Due to the complexity of the application environment and rehabilitation needs, various types of intelligent rehabilitation engineering technologies based on treatment plans have emerged. For example, intelligent VR game development based on rehabilitation training is used for the physical rehabilitation of stroke patients, which repeatedly promotes neurofeedback by encouraging patients to complete tasks and interactions in the virtual training environment. Based on automated program design of gait robot assisted rehabilitation, regular and orderly limb training. Based on the sensor device evaluation and real-time feedback rehabilitation model, the upper limb activity evaluation and feedback during training, due to the sensor fusion system has the advantages of small size and wide range of applications, higher practicality. However, in addition to professional large rehabilitation devices with embedded sensors, sensor-based rehabilitation applications commonly suffer from measurement errors, noise problems, data pre-processing problems, and gesture as-

*Corresponding author.
Email addresses: qi.liu@nuist.edu.cn (*Qi Liu*[a,#]), yexiplvos@hotmail.com (*Xueyan Wu*[b,#]), jyhking@vip.qq.com (*Yinghang Jiang*[b]), x.liu@napier.ac.uk (*Xiaodong Liu*[c]), zyh@nuist.edu.cn (*Yonghong Zhang*[d]), njuxlxu@gmail.com (*Xiaolong Xu*[a]), lianyongqi@qfnu.edu.cn (*Lianyong Qi*[e])

[#]Both are First Authors due to equal contribution to the work of this article.

sessment model accuracy problems. A MEMS sensor-based upper limb gesture recognition model has been proposed for FES rehabilitation system, which uses data pre-processing to de-weight, interpolate, and extract features from the fusion data collected by three sensors to ensure data stability, and design a fully connected neural network to recognize six upper limb gestures, which effectively improves the practicality of the gesture evaluation model and has strong practical significance.

2. Related Work

FES therapies as a rehabilitation method for stroke patients are widely applied nowadays, which is based on the principle that electrical current stimulation of the muscles acts on nerve cells to produce contraction, excite action potentials and generate nerve impulses. This stimulation through the action of external electrical currents is functionally consistent with voluntary muscle contraction, whist continuous neuromuscular stimulation can promote the repair of necrotic nerves and the recovery of motor function in stroke patients. Therefore, systems associated with FES need to be increasingly improved, such as gesture recognition during functional recovery.

Gesture recognition is mainly divided into image-based recognition and sensor-based recognition, which originated from the biological motion perception model designed by Johansson in 1973 [1], tracking the movement of biological joints and limbs through a camera or sensor to perceive the biological motion process[36].

2.1. Image-Based Gesture Recognition

In the field of image-based gesture recognition, a 3D structured light depth sensor consisting of an RGB camera and an infrared camera using the "Kinect" sensor developed by Microsoft Corporation is a widely used image extraction method. Kinect camera perception information for human activity recognition, combined with K-means clustering, support vector machine (SVM) and hidden markov model (HMM) algorithms to detect and model the actions involved in the activity, experiments have shown an accuracy of 77.3%, a recall rate of 76.7%, and the ability to display the activity scene when it is active [9].Min-Chun's team divided the recognition into retrieval phase and learning phase. In retrieval phase they use the nonlinear time warping (NTW) algorithm, which evaluates gesture through the difference between static and dynamic during the learning phase, and the experiment achieved an average accuracy of 90.8% [6]. Sriparna et al. used the six-stage method to segment the image colors, the gesture was amplified to create the original posture skeletal key-point map, and the Radon transform was performed using the corresponding primitive integral map for different postures, converting the data into a linear integral map with Euler numbers for judgment, with an accuracy of 91.35% [11]. Hiroomi designed a hardware-based recognition and self-organizing neuron. The hybrid recognition network of classifier was used to establish a single-layer feedforward neural network by mapping the gesture features extracted from the image to a multidimensional map of SOM neurons, and it was experimentally demonstrated that the system could complete the recognition at 60 frames/s with an average accuracy of 97.1% [32].

In 2016 Wein et al. used a light-field camera to build a 3D gesture depth image model, used principal component analysis (PCA) to obtain feature vectors without preprocessing, and used the k-NN algorithm with a genetic algorithm (GA) for feature selection, experimental showed that this method can maintain the stability of recognition when the noise exceeds 5% [35]. In 2017, Sameh's team extracted 15 human skeleton joint data through Kinect sensors, used ConvNets to input RGBD video frames for pose recognition, and used SVM classifiers for high-precision pose estimation, avoiding the tedious pre-processing of scenes upfront [2]. Ding designed a linear subspace based on Grassmannian manifold using the 3D rigid body relationship matrix (RMRB3D) established by the rotational motion of the human body pose, extracted the pose and generated symbolic sequences by spectral clustering between points, and finally established action sequences by dynamic time warping and HMM up to 72%. In 2018, Munoz et al. designed a Kinect-based multimodal learning analysis model as AdaBoost, for behavior recognition by detecting students' learning state through body and gesture postures [22]. Chin et al. collected datasets of office sitting postures through Kinect and designed a posture recognition model based on SVM and artificial neural network (ANN), comparing and finding that linear SVM linearly Nuclei have the highest accuracy [4]. Hyun-Gook et al. obtained human features in image sequences by cosine discrete transform before truncating the singular value decomposition (SVD), with improved classification speed and a 30% increase in accuracy [39].In 2019, Kamel designed an Action-Fusion-based model from depth map and pose data to recognize human movements, continuous depth maps with moving joint descriptors were used as input using three-channel Convolutional Neural Networks (CNNs), and later training with two depth motion images (DMIs) to achieve the final movement classification, which experimentally demonstrated the efficiency of this method[14]. Yu et al. proposed a robust and fixed-time zeroing neural dynamics (RaFT-ZND) for time-varying nonlinear equations using nonlinear activation functions in order to solve the problem of convergence time dependence on the initial state in zeroing neural networks, and the simulation results showed the advantages of its problem solving[39]. Jin et al. proposed an improved zeroing neural network model to solve the convergence problem of time series by finding a prediction scheme for time-invariant nonlinear equation (TINE) equation and time-varying nonlinear equation (TVNE) equation in finite time[13]. In 2020, Ren's team improved the posture recognition algorithm of autonomous assistive robots for patients requiring care by combining fuzzy logic and SVM algorithms via Kinect, and experimental results showed a 97.1% accuracy in recognizing full-body lying posture data for 32 test subjects[29]. Takano et al. used the same two-dimensional data for posture recognition methods to improve motion Identifying the HMM model and generating textual descriptive information from image categorized observations, establishing a probabilistic framework using words linked to motor primitives to enable accurate grammatical alignment, has been shown to be effective in improving the effectiveness of taking action in geriatric care [31].Mohamed used an RGB-D sensor to establish a visual data-based supervision system by extracting from two-dimensional images the convolutional features, using the body joint configuration in 3D space for SVM classification, effectively improve the robustness of the posture classification model [7].

2.2. Sensor-Based Gesture Recognition

Due to the low-power, low-cost, small size and high-performance characteristics of sensors, the field of sensor-based recognition is more integrated with the application field.[5] Hu et al. established an ANN network-based posture classification model by placing six flexible sensors under an office chair, which was implemented on a Spartan6 programmable gate array , and experiments showed that the model floating-point evaluation accuracy is 97.78% and the maximum propagation delay is 8.714 ns, which is highly applicable[12]. Sazonov's team used wearable sensors to build an SVM-based framework for SVM and polynomial logic recognition , and used the labelled data for PAC/EE algorithm training, using fast artificial neural networks (FANNs) to train MLDs model, the model has been experimentally demonstrated to be effective in reducing execution time for real-time bio-feedback systems [27]. In 2016, Xu identified sleep posture through pressure-sensitive sensor sheets, using the moving distance to use the pressure-sensitive image as a weighted 2D image, combined with EMD and euclidean metric for similarity measurement, and experimentally demonstrated that the model improved accuracy by 8.01% compared to traditional sleep posture recognition methods [38].In 2017, Alessio, by giving the experimenter a device to calculate distance between the experimenter and the gyroscope is used to infer the postural changes of the human body, and the data is classified according to the ultra-wideband transceiver with two-way ranging mode combined with accelerometer and gyroscope data, but this has certain requirements for the stature of the experimenter, and the accuracy is significantly lower in the case of extreme stature of the subject [33].Lin et al. designed a smart insole that recognizes the activities of the patient for the nursing field, and calculated the temporal and spatial distance based on the built-in pressure sensor, quantifying motion similarity and classifying human posture, and experimental results showed that the classification accuracy for eight common activities in nursing rooms was 91.7% [18].Kitzig's team developed an intelligent vital sign detection system for home-bound elderly people, using sensors embedded in the bed and chair to collect information on the subjects' detection parameters and movement/sleep patterns, establishing a multilevel pattern recognition system, experimentally shown to have an accuracy of 93.2%[20]. However, this method requires the simulation of each movement of the subject and the advance establishment of physiological parameters such as body weight to ensure the accuracy of the model elements [16].

In 2019, Rhee established a four-channel information fusion model based on accelerometer and EMG for electromyography (EMG)-based finger and arm movements, compensated for the interference caused by the acceleration signal by fitting the gravity model, and used the quantized wavelength algorithm combined with the nearest-neighbor method to establish an action classification model, with an experimental average accuracy of 85.7%, reducing the effect of different arm movements on EMG signal interference [17].Wang et al. proposed an adaptive neural control strategy by considering a quantization control approach, which is able to analyze the stability in time and eliminate the quantization error of a stochastic nonlinear system with finite time[34]. Ma established a sitting posture recognition system based on triaxial accelerometers, transformed the acceleration data into feature vectors for component analysis, and used SVM and K-means clustering to classify sitting postures, and experimentally demonstrated the superiority of the SVM algorithm on sitting posture classification [21]. Liu et al. designed a home monitoring and assessment model for the activities of the elderly[2]. A back propagation neural network based model was designed using triaxial accelerometer and pressure sensor data, and the validity of the model was experimentally demonstrated [19]. Li's team designed an information fusion-based D-S theory using body sensor networks (BSNs), and the usability of D-S in the field of posture detection was experimentally demonstrated[10]. Segerra used sensor fusion data with gyroscopes and accelerometers to design an inertial data estimation orientation model based on Kalman filtering with Mahony filtering, which experimentally demonstrated the excellence of Kalman filtering, but upfront subjects had to be individually pre-processed to correct for bias[30]. In 2020, Permatasari's team optimized the gait recognition problem by using accelerometer and gyroscope data to encode covariance matrices, and since non-star covariance matrices are symmetric positive definite matrices, the SPD matrix was designed as a feature fusion of point pairs of data in the Riemannian-plane, which experimentally proved to be effective in overcoming the large datasets required in traditional gait recognition and the long computation time problem[26]. Zebin analysed the effects of different parameters[3], features and sensor locations on the overall recognition based on a model in which a wearable inertial sensor inputs a multichannel time series signal and automatically outputs a classification of human body activities, and showed the importance of establishing a data set for different activities to classify activities of multiple genera[41].

Zhao et al. used a multi-sensor obtain lots of data to establish a self-supervised learning model for sleep recognition, increase data capacity through self-supervised pre-training, processing frequency domain information, use the rotational view t-SNE to represent multidimensional data features, and use the LSTM fusion condition random field, the test proved the effectiveness of the algorithm[42]. Wang collected four swimming style data through inertial sensors arranged at the waist, based on the HMM extracted the data fusion information with high recognition rate[37]. In 2019, Feng realized multi-source information fusion through a multivariate LoRa system, completed smooth filtering and data de-noising by pre-processing sensor data and feature extraction, used sliding windows for stream segmentation and frequency domain feature extraction, and constructed an MRMR-SFS-RF-based pose recognition model, which experimentally proved that in a small number of identification accuracy in the data was 98.9% [8]. Sharma established a posture classification model based on myoelectricity sensors through a multi-sensor wearable device, placing 8-channel SEMG sensors equidistantly on the forearm and using filters to remove irrelevant features, but the drawback is that the accuracy of sensor data classification depends largely on the precision of placement[40]. Andre et al. established human activity recognition through wearable multi-sensor insoles classifier, which is an ultrasonic sensor to detect lower limb motion in an unsupervised environment, but also requires accurate sensor placement during data acquisition to determine the orientation of each sensor and attachment location in advance[28]. Nweke et al. designed a multi-view ensemble algorithm to fuse multi-sensor data for medical applications, using logistic regression and k-approximation algorithms for posture recognition, and fused the synthetic over minority sampling technique improves data balance, and experiments have demonstrated the effectiveness of the algorithm in classification[24]. Paola's team designed a multi-sensor data fusion system to improve the probability of determining contextual information in a multi-user smart scenario, collecting data from heterogeneous sensors in a smart environment and noise reduction through a dynamic Bayesian network to dynamically configure the sensor state based on the data[15]. Upgrade the ability to generalize on the system[25].

3. Algorithm

3.1. Algorithm Structure Design

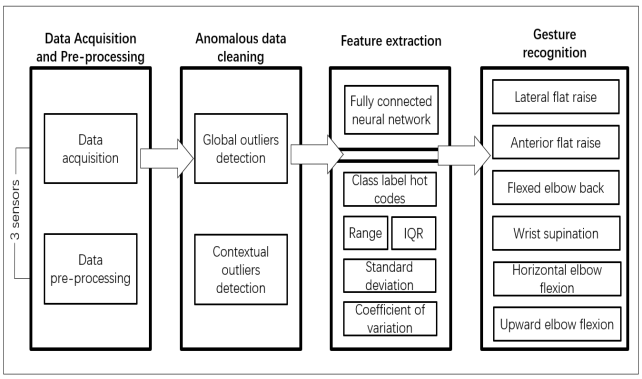


Figure 1: Pipeline of the four main stages of algorithmic architecture design

A fully connected neural network, also known as a multilayer perceptual machine, is a network in which, in addition to the output layer, each neuron in two adjacent layers is connected to each neuron in the next layer, with the first layer of the network serving as the input layer, the last layer as the output layer, and the remaining layers collectively referred to as the hidden layer. The following figure shows the structure of a simple fully connected neural network.

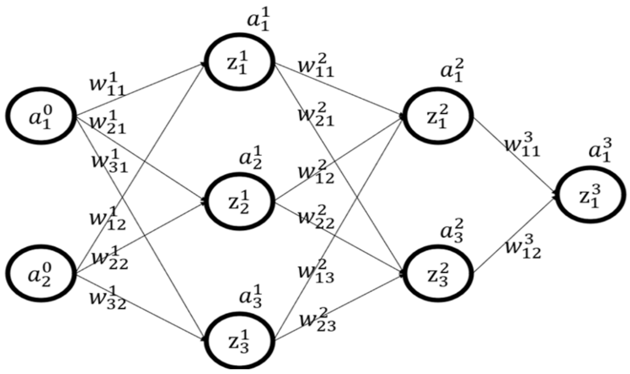


Figure 2: Structural diagram of a three-layer fully connected neural network

Where a_i^l represents the output of the neuron, where l represents the number of layers, i represents the neuron number, z_i^l represents the output of the inactivated neuron, l represents the number of layers, w_{ij}^l represents the weighting factor of the neuron, where i represents the neuron number corresponding to the next neuron layer, and j represents the neuron number corresponding to the previous neuron layer. The formula for calculating each parameter is as follows:

$$z_i^l = w_{ij}^l a_i^{l-1} + b_i^l \tag{1}$$

$$a_i^l = \sigma\left(z_i^l\right) \tag{2}$$

Where b_i^l denotes the bias coefficient, $\sigma\left(z_i^l\right)$ denotes the activation function of the fully connected neural network. The fully connected neural network takes the output of the previous layer of each neural network as input, then calculates the output value of that neuron through the formula (1), and uses the activation function to normalize the result by mapping the output value domain between (0, 1) to prevent the problem that the input result is too large to lead to the poor training effect of the network. The network parameters are optimized using forward propagation and backward propagation algorithms.

1) Forward Propagation

Forward propagation is a linear weighted summation process. By performing a weighting operation on the previous layer of neurons and the corresponding weights, plus the bias parameter b, the output of the local layer of neurons is obtained using a non-linear activation function (e.g., sigmoid, ReLU, etc.) with the following formula:

$$a^l = \sigma\left(z^l\right) = W^l a^{l-1} + b^l \tag{3}$$

2) Back Propagation

Back propagation, as a common method to train artificial networks, iteratively calculates the gradient of the loss function in the network through the chain rule of composite functions, and updates the neuronal weights through the feedback of this gradient to minimize the loss function. BP algorithm can learn adaptively according to the preset parameters, efficiently calculate the partial derivatives of the parameters, and has strong function Reproducibility and non-linear mapping capability. After the completion of forward propagation of each batch of data in a fully connected neural network, a loss function is established between the true output value corresponding to the input value and the forward propagation output value of the neural network.

$$J\left(a^l\right) = J\left(\sigma\left(z^l\right)\right) = J\left(\sigma\left(W^l a^{l-1} + b^l\right)\right) \tag{4}$$

Where W is the weighting factor matrix, a is the learning rate. For layer l output layers use the chain rule to solve the gradient:

$$\frac{\partial J(a^l)}{\partial W^l} = \frac{\partial J(a^l)}{\partial z^l}\left(a^{l-1}\right)^T \tag{5}$$

$$\frac{\partial J(a^l)}{\partial b^l} = \frac{\partial J(a^l)}{\partial z^l}(E)^T = \frac{\partial J(a^l)}{\partial z^l} \tag{6}$$

Taking the common term $\frac{\partial J(a^l)}{\partial z^l}$ as the error term of the neural network and setting it to δ^L, the model shows as follow:

$$\delta^l = \frac{\partial J(a^l)}{\partial z^l} = \frac{\partial J(a^l)}{\partial a^l} \odot \sigma'\left(z^l\right) \tag{7}$$

For the l hidden layer uses the chain rule to solve the gradient, from the above equation:

$$\delta^l = \frac{\partial J^l}{\partial z^l} = \frac{\partial a^l}{\partial z^l}\frac{\partial z^{l+1}}{\partial a^l}\frac{\partial J}{\partial z^{l+1}} = \frac{\partial a^l}{\partial z^l}\frac{\partial z^{l+1}}{\partial a^l}\delta^{l+1} \tag{8}$$

Solving for the intermediate term and recurring the error terms of the hidden layer yields the gradient formula:

$$\frac{\partial J(a^l)}{\partial W^l} = \frac{\partial J(a^l)}{\partial z^l}\left(a^{l-1}\right)^T = \delta^l\left(a^{l-1}\right)^T = \left(W^{l+1}\right)^T \delta^{l+1} \odot \sigma'\left(Z^l\right) \tag{9}$$

$$\frac{\partial J(a^l)}{\partial b^l} = \frac{\partial J(a^l)}{\partial z^l}\left(a^{l-1}\right)^T = \delta^l(E)^T = \left(W^{l+1}\right)^T \delta^{l+1} \odot \sigma'\left(z^l\right) \tag{10}$$

Then the parameter update formula is:

$$W^l = W^l - a\frac{\partial J(a^l)}{\partial W^l} \tag{11}$$

$$b^l = b^l - a\frac{\partial J(a^l)}{\partial b^l} \tag{12}$$

The fully connected neural network parameters are updated according to Eq. (11), (12). Whenever a batch of data completes forward propagation, the weights and bias parameters are updated using backward propagation, and the process is repeated until the loss value is less than the set threshold or the network update reaches the number of iterations to stop the BP update and obtain the output value, which is output to the activation function for result normalization.

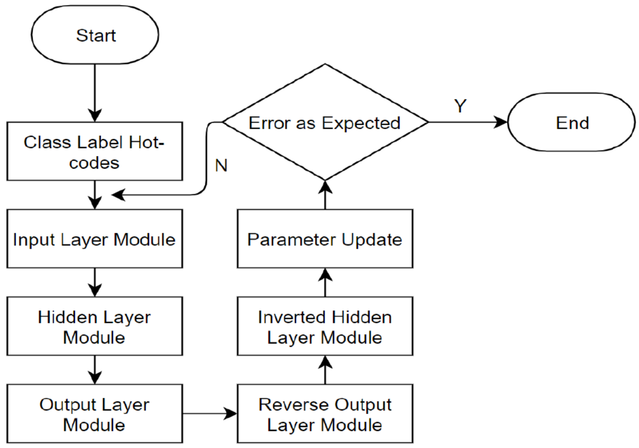


Figure 3: Gesture recognition schemes for fully connected neural networks

One-hot encoding is performed on the labels of the pose dataset to extend the values of the discrete features to the Euclidean space to rationalize the computational distance and improve the model computational efficiency and non-linearity capability.

The fully connected network model constructed in this paper consists of four components. The first one is the input layer module, which is responsible for inputting the format of the pose data and the initialization task of the neuron parameters at each layer during the first execution, set to read a set of 1590 6 pose matrix data at a time. The hidden layer module consists of hidden layers containing 30 neurons, the number of layers is determined by comparing the recognition rate, and is responsible for weighted summation of the output data of the upper layer neurons and activation by the activation function to generate the input values of the lower layer neurons. The output layer module is responsible for obtaining the predicted probability values of the six postures from the incoming data of the upper layer neurons. The tuning module is responsible for calculating the activation values for each neuron, and based on the activation values, it calculates the losses and parameter gradients for each layer, and makes parameter adjustments from the output layer forward.The gesture dataset is trained by the above method to derive the final recognition model.

3.2. Activation Function

In an artificial neural network, the Activation function of a neuron defines the mapped output of that neuron at a given input, which introduces non-linearities to the neuron and avoids the problem of linear functions where the output of each layer is the input of the previous layer, allowing the neural network to approximate the non-linear function and apply it to the non-linear model. The activation function can effectively improve model robustness, reduce the gradient vanishing problem, accelerate model convergence, also can perform non-linear transformations, and facilitates the use of back-propagation to update parameters during artificial neural network training. This section describes a portion of the activation functions used in the experiment.

1) The *tanh* function

$$y = \tanh(x) = \frac{2}{1+e^{-2x}} - 1 \tag{13}$$

The *tanh* function can map the input to the range of $[-1, 1]$. When the input is 0, the output is also 0, and it is better to use the *tanh* function when the data features are obvious. However, when the activation value is close to 0 or 1, the function gradient tends to be close to 0, which will easily cause the gradient saturation problem caused by the parameter diffusion in the back propagation, making the training efficiency of the neural network low.

2) The *ReLu* function

$$y = \max(x, 0) \tag{14}$$

Compared to the *tanh* function, *ReLu* has a tremendous accelerating effect on the convergence of random gradient descent. *tanh* contains exponential operations in its derivation, while *ReLu* derivation has almost no computation at all. Since the *ReLu* function is constant in the non-negative range of the gradient, it can effectively avoid the gradient disappearance problem, so that the model convergence remains steady state and convergence is faster. As a segmentation function, *ReLu* has the property of unilateral inhibition, which can convert the output value of all negative inputs to zero, and the positive value remains unchanged, which makes the neuron sparsely activated and better exploits the data features to fit the training data when training a deep classification model. However, in the case where the learning rate is set too high, the large gradient will make some of the neuron weights update too much and fall into the hard saturation region, resulting in other neurons not being activated, and in order to avoid this situation, the parameter update process needs to be adjusted automatically.

3) The $softplus$ function

$$y = \log(1 + e^x) \tag{15}$$

$softplus$ is a form of analytic function which is a smooth approximation of *ReLU* function, due to the range of the independent variable is $(0, ++)$, there is unilateral inhibition, can generate β and σ parameters based on the normal distribution, due to $softmax$ is an exponential function, has a wide range of excitation boundary, the use of $softplus$ as an activation function can effectively avoid the parameters in the back propagation of the calculation Volume over fitting problem. Neurons with an output of 0 provide some neural network sparsity, reducing the probability of over-fitting situations.

3.3. Optimizer Model

1) Stochastic gradient descent

The random gradient descent algorithm takes m small batch samples and computes their gradient means to obtain unbiased estimates of the gradient through the distribution generated by the data, and adjusts the network parameters using one example of the samples as a way to approximate all the samples. Stochastic gradient descent (SGD) is a first-order optimization algorithm with the property of being able to make single updates, which effectively avoids the problem of redundant computation caused by batch gradient descent. The calculation is fast and can be updated online. However, there are problems such as gradual decrease in speed when approaching the local minimal value, easy to fall into the local optimal solution which leads to each update not following the correct direction, optimization easy to fluctuate and so on.

2) Adagrad

Adagrad is an adaptive gradient algorithm that adapts the learning rate and frequent parameters, making it suitable for dealing with sparse data. By automatically setting the learning rate inversely proportional to the sum of the historical parameter modes as the global learning rate, and adaptively adjusting the learning rate with the changes of the gradient, the smaller gradient at the early stage of the training as an incentive stage, and the larger gradient at the late stage of the training as a penalty stage, to reduce the learning rate, effectively solving the problem that the random gradient descent algorithm is easy to fall into the local optimum solution.

3) Adam

Adaptive moment estimation optimizer (Adam) is another method to adjust the network parameters by calculating the adaptive learning rate of each parameter, which combines the advantages of Momentum and uses first-order gradient to optimize the stochastic objective function. As the invariance of Adam's diagonal scaling is suitable for solving large-scale data problems and large-noise non-stationary problems, it requires less memory, is computationally efficient, and requires only a small amount of parameter adjustment to achieve parameter optimization, with very good training results and outstanding performance in the field of machine learning.

3.4. Security and Safety Mechanism

Safety: stop mechanism via recognition of dangerous gesture. The model is proposed for the medical field, in order to ensure the safety of use, an emergency stop mechanism has been embedded based on dangerous gesture detection, when it detects that the user's limb angle change feature exceeds the maximum threshold value or the movement change speed exceeds the maximum threshold value,the protective emergency stop module and start the electrical stimulation interlock function have been started to ensure the safety of the model. The flowchart is shown below.

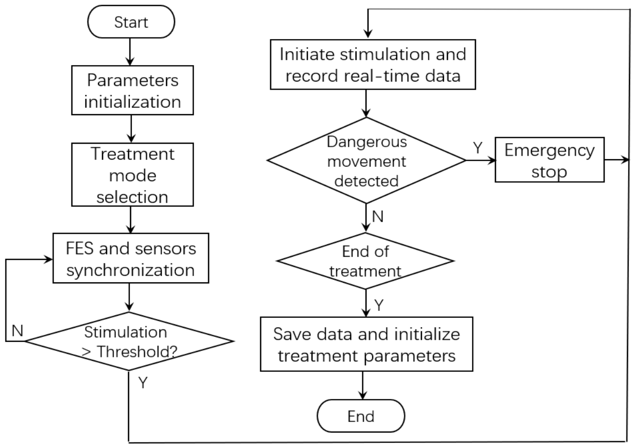


Figure 4: Stop mechanism flowchart

Security: light-weight cryptography via hash. Due to the high privacy of medical data, in order to ensure the data security, a lightweight hash algorithm has been embedded to encrypt the sensor data and generate the data into a hash string of the same length through one-way hash encryption, to achieve irreversible mapping and MD5 has been used to validate data consistency, when data is transferred, the transmission secret key is calculated into a value, then it is saved with the system. The stored value is compared, and the value of data itself is not performed during the transmission to ensure the security transmission of data.

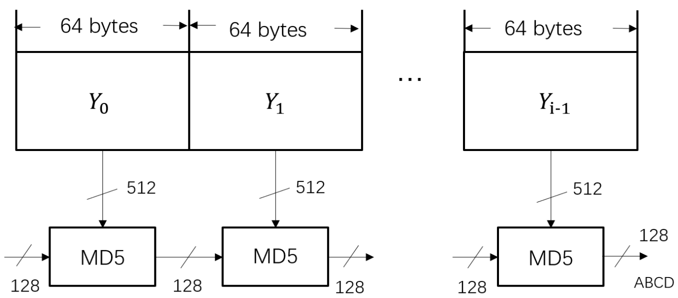


Figure 5: Lightweight encryption method for MD5 group processing

The added data is an integer multiple of 64 bytes, and then divided into i groups according to 64 bytes, with the intermediate hash function value used as input for the next group. The four linked variables A,B,C,D are used as data summaries when all data processing is complete.

4. Experiment and Results

4.1. Experiment Environment and Preparation

The data acquisition module is the research basis of the gesture recognition, providing the most primitive gesture data for classification model training and evaluation, and its data quality has a great influence on the final performance of the recognition model.

The MPU6050 sensor module has been implied as a data acquisition device, which has the following features.

1) Lightweight: the device is small enough and light enough to be easily worn by the human body and does not interfere with joint movement.

2) Appropriate frequency: the sampling frequency can be reasonably adjusted for different environments to ensure the integrity of human action information and meet the real-time requirements of the system.

3) Robust and stable: the acquisition equipment will not be interrupted during the data acquisition process and will remain operational in non-human conditions.

4) Reliable transmission: there is no loss of data while the acquisition device is in operation.

5) Reliable transmission: there is no loss of data while the acquisition device is in operation.

6) Integrated: InvenSensed MotionFusion and runtime calibration firmware to ensure optimal performance of sensor fusion algorithms.

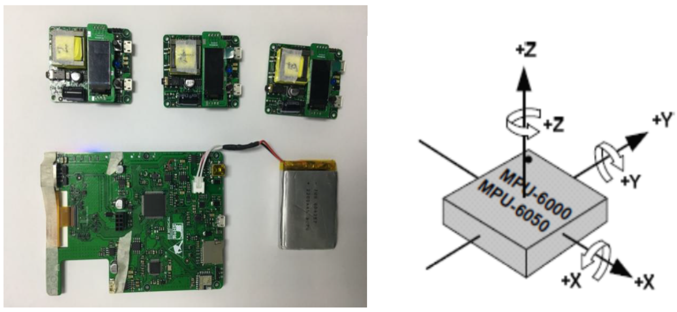


Figure 6: Experimental data acquisition equipment

The MPU6050 is a scalable digital motion processor with an integrated MEMS 3-axis accelerometer and gyroscope that accurately tracks high-speed and low-speed motion. With a wide range of user-defined sensing ranges, the accelerometer senses velocities of ±2g, ±4g, ±8g and ±16g, and angular velocities of ±250, ±500, ±1000 and ±2000/sec (dps). During the data acquisition process, the MPU6050 puts the calculated values into registers, then the microcontroller reads them via I2C and sends them to the computer via serial or Bluetooth module.

Since upper limb gesture involves movement of multiple joints, the amplitude and intensity of each movement gesture varies, the different ages and physiques of the test subjects affect the accuracy of gesture recognition. Some researchers use single sensor for human upper limb gesture recognition, and the incomplete information collected will reduce the accuracy of gesture recognition. Multiple sensors for data acquisition can also cause inconvenience to the subjects and affect the movement while improving the recognition accuracy.

So the three-channel data acquisition has been chosen in method., using the acceleration signal and gyroscope signal collected by three sensors worn on the upper limb of the identification object as the sample data, the data acquisition process is shown in the figure below.

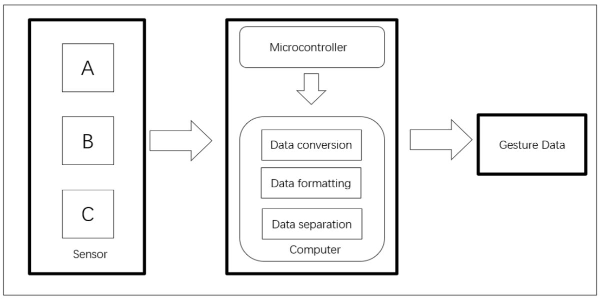


Figure 7: Data acquisition process

4.2. *Data Acquisition and Pre-processing*

This section introduces data acquisition and pre-processing for the experiment, the data acquisition device is firstly worn on the upper limb arm at the centre of the back of the hand, the middle of the small arm and the middle of the large arm.

4.2.1. *Data Acquisition*

The following figure shows the placement of the sensor device. The arm using three-dimensional coordinates to establish a spatial model, in the established coordinates, it is stipulated that the direction of the finger is parallel to the x-axis direction of the MPU6050 sensor, the chip z-axis is perpendicular to the upper limb, the chip y-axis points to the inner side of the arm. The x-axis, y-axis and the upper arm are in the same plane, and the plane is perpendicular to the z-axis.

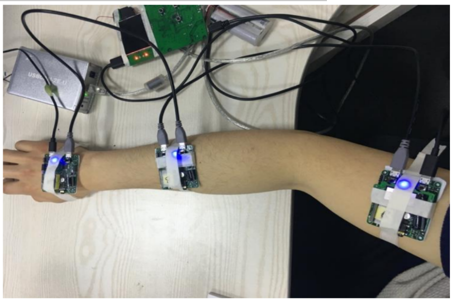


Figure 8: Sensor data acquisition

Based on the recommendations of relevant health care professionals and medical data, six upper limb gestures have been defined applicable to stroke patients: anterior flat raise, lateral flat raise, upward elbow flexion, flexed elbow back, wrist supination and horizontal elbow flexion. In order to improve the accuracy of the data and ensure that the limb movements re not affected by the environment during data collection, the collected data were formatted in the same way, with 100 Hz set as the sensor frequency but with acceleration and gyroscopic data in the following format: A python program has been implied to sort and filter the three sensors data. The experiment collected three stroke patients with 60 repetitions of each set of movements and contained approximately 780,000 gesture data.

4.2.2. *Data Pre-processing*

1) Drop duplicate

Table 1: Experimental data after formatting

	Timestamp	accX	accY	accZ	GyroX	GyroY	GyroZ
A	1555830289723	+0.998697	-0.139870	-0.030114	+0.580153	-0.681527	+0.512977
B	1555830289723	+0.996256	-0.092974	+0.328092	-0.347023	-0.014656	-0.092974
C	1555830289723	+0.976256	-0.098415	+0.234809	-0.044198	+0.302595	-0.098415

The raw acceleration and angular velocity signals of the gesture are visualized as follows, using the B-sensor data from the lateral lift as an example.

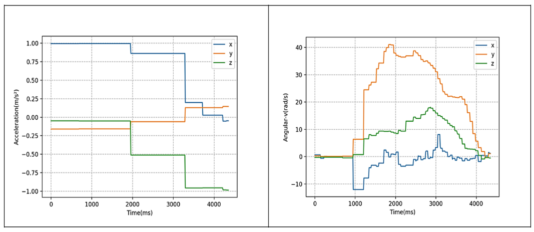


Figure 9: Lateral lift data waveform

The sensor sensitivity issue can cause noise and jaggedness in the raw data waveform graph. The duplicate datasets have been dropped, using the gyroscope data as an example, and the drop duplicate waveform plots are as follows:

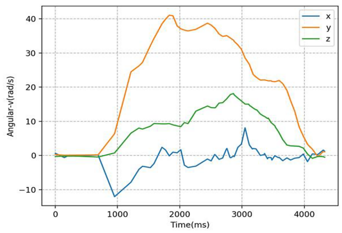


Figure 10: Gyroscope data waveforms after drop duplicate

2) Interpolate

In order to fill in the missing data, eliminate data jagging, and ensure the smoothness of the timing data, three sample bar interpolation has been used to pre-process the dataset, and the three sensor signals A, B, and C for the six gestures were processed as follows:

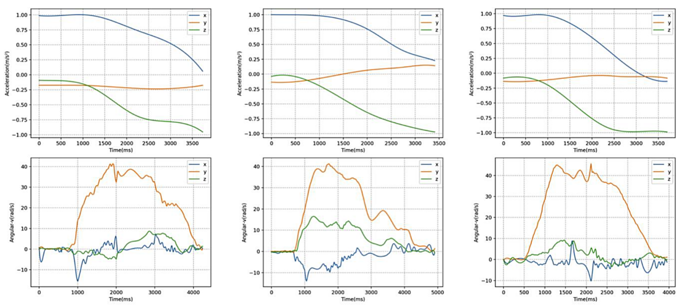


Figure 11: Pre-processed waveform of lateral flat raise data

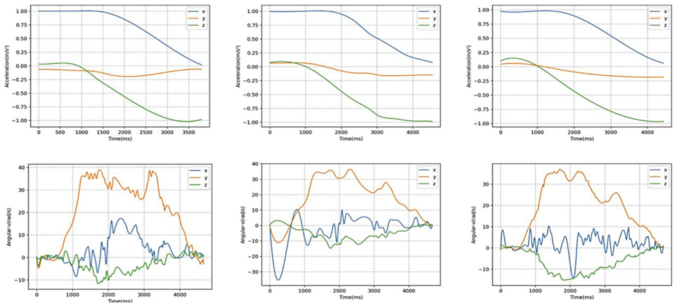


Figure 12: Pre-processed waveform of anterior flat raise data

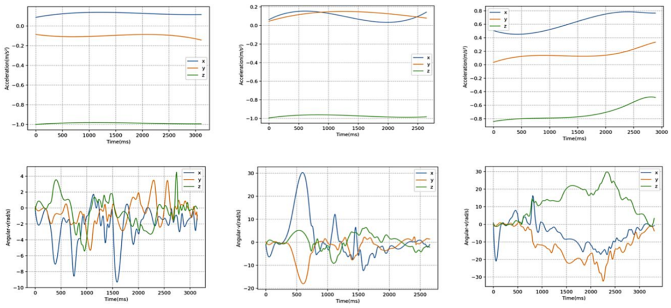


Figure 13: Pre-processed waveform of upward elbow flexion data

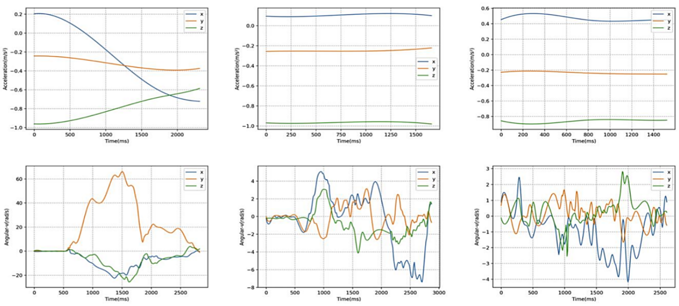


Figure 14: Pre-processed waveform of wrist supination data

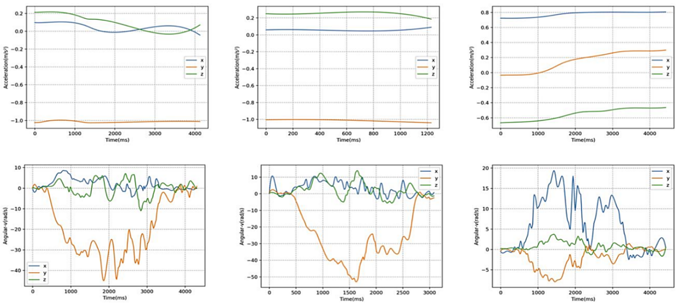


Figure 15: Pre-processed waveform of horizontal elbow flexion data

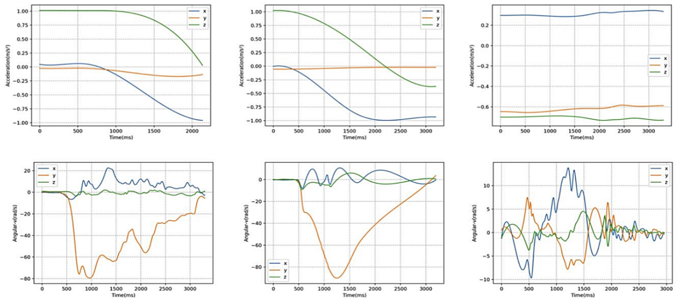


Figure 16: Pre-processed waveform of upward elbow flexion data

3) Interpolate Feature Extraction

The time domain analysis method used in this paper is a signal feature extraction method mainly applied to low speed and variable speed devices, with N denoting the number of rows of data in a time window and i denoting the i^{th} row of data, the selected features are as follows:

(a) Range: the difference between the maximum value and the minimum value of the total signal sample, it is the simplest measurement of the dispersion of the signal and can show the range of data variation.

$$R = X_{\max} - X_{\min} \tag{16}$$

(b) Interquartile range:a robustness to represent the dispersion of variables in a signal sample system and method to determine the third quartile and first quartile, respectively, by calculating the first quartile (Q1), which is the number at 25%, the median, and the third quartile (Q3), which is the number at 75%, through the inner and outer limits of the anomaly truncation point.

$$\mathrm{IQR} = Q_3 - Q_1 \tag{17}$$

(c) Standard deviation:used to measure the degree of dispersion of a set of signal samples, being the square root of the arithmetic mean of the square of the deviation from the mean.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2\,,\mu} = \frac{1}{N}(x_i+\cdots+x_N) \tag{18}$$

(d) Coefficient of variation: As a normalized measure of the dispersion of the probability distribution of signal data, the coefficient of variation does not need to refer to the average value of the data, and can be used as a reference when comparing data with different means or different factors, but its disadvantage is that when the average value is close to zero, even the smallest perturbation can have a huge impact, resulting in the coefficient accuracy is not good enough.

$$c_v = \frac{\sigma}{\mu} \tag{19}$$

By adding the original signal data, the data window size is made consistent to ensure that the original pose information is complete. Before the experiment, the longest pose sample has been found out by the program, and the completion time is 5.3 seconds, then add windows to the data set by interpolating all the sensor data three times.

Table 2: Identify results of different number of hidden layers

Hidden layer number	Average recognition rate	Time(sec)
1	91.27%	14.576
2	81.34%	18.743
3	94.08%	20.492
4	84.18%	26.533
5	89.87%	23.33
6	79.41%	31.106

Table 3: Identification results of different activation functions

Activation function	Average accuracy
ReLu	91.31%
softplus	93.07%
sigmoid	37.84%
tanh	81.96%
softsign	90.33%

Table 4: Identification of the results of different optimizers

Optimizer	Average accuracy
Adam	94.25%
RMSProp	96.01%
SGD	13.07%
AdaDelta	93.17%
AdaGrad	97.19%
Adamax	94.35%
NAdam	96.11%

4.3. Results and Evaluation

In order to investigate the effect of the number of hidden layers on recognition accuracy and recognition efficiency, this paper investigates the recognition accuracy and time-consuming of hidden layers 1 to 6, with the number of neurons all 30, and the results are shown in the following table From the table, it can be concluded that increasing the number of hidden layers does not result in a continuous improvement of the recognition rate. However, the multilayer hidden layer neural network structure takes more execution time compared to the structure with fewer hidden layers. Therefore,a fully connected neural network with 3 hidden layers has been adopt for gesture recognition, considering the average recognition rate and time cost.

In order to investigate the effect of the activation function on the recognition effect, improve the robustness of the model, reduce the problem of gradient disappearance, and accelerate the convergence of the model, the recognition rate of fully connected neural network models has been investigated under different activation functions and compared the results as follows.

The experimental results show that the best recognition is achieved when the activation function of the hidden layer is softplus, and the recognition rate can reach 93.07%.

In order to find the optimal parameters of a fully connected neural network, the choice of the optimizer model is also an important factor. Different optimizers are used to reduce the error through the corresponding algorithm in an iterative manner until the model reaches the optimal state. The above experiments were conducted under the condition that the optimizer is adaptive moment estimation (Adam), in order to find the optimizer suitable for this dataset, the recognition rate of the neural network model corresponding to different optimizers was studied in this paper and the comparison results are as follows.

The random gradient descent optimizer is not applicable to the fully connected neural network model in this paper, and its recognition rate is only 13.07%, which is equivalent to no effect. rmsprop, adagrad, and nadam optimizers all achieved good recognition results, with an accuracy of more than 96%, while adagrad has the best recognition effect, with an average recognition rate of 97.19%.

After the above experiments, the fully connected neural network model in this paper selects the hidden layer containing three activation functions as softplus, identifies the attitude dataset using the adaptive gradient descent optimizer, and performs three ten-fold cross-validation to take the mean value.

Logistic regression is employed since it well fits the context of scenarios in this paper; that is, it supports multi-classification and prediction of the probability of event occurrences, facilitates the analysis of factors influencing the occurrences by means of the values of characteristic parameters, and can therefore be applied to situations handling a subsequent amount of incoming data. The k-NN, logistic regression and random gradient descent algorithms has been performed experiments by using the same data and preprocessing methods as follow tables.

Table 5: k-NN identification results after feature extraction

Data window (milliseconds)	Group number	Accuracy	Running time (seconds)	Average accuracy	Time spent
1000	1	0.91	0.0471	89.82%	0.047
	2	0.8867	0.0469		
	3	0.8978	0.047		
2000	1	0.9267	0.0496	91.96%	0.0504
	2	0.9	0.0533		
	3	0.9322	0.0483		
4000	1	0.9444	0.0503	94.87%	0.0507
	2	0.9556	0.049		
	3	0.9461	0.0528		

Table 6: Logistic regression identification results after feature extraction

Data window (milliseconds)	Group number	Accuracy	Running time (seconds)	Average accuracy	Time spent
1000	1	0.8633	0.3805	87.27%	0.3946
	2	0.8983	0.3914		
	3	0.8567	0.4121		
2000	1	0.9444	0.461	93.14%	0.5032
	2	0.9167	0.397		
	3	0.9333	0.6516		
4000	1	0.9511	0.4125	95.09%	0.403
	2	0.9572	0.3326		
	3	0.9444	0.464		

Table 7: Logistic regression identification results after feature extraction

Data window (milliseconds)	Group number	Accuracy	Running time (seconds)	Average accuracy	Time spent
1000	1	0.8556	1.206	84.03%	1.112
	2	0.8433	1.1974		
	3	0.8222	0.9326		
2000	1	0.93	1.057	92.16%	1.143
	2	0.9183	1.215		
	3	0.9166	1.157		
4000	1	0.9267	0.9428	93.16%	1.137
	2	0.935	1.196		
	3	0.9333	1.273		

Summarizing the above comparison experiments, the recognition rates and time spent by each algorithm are as follows:

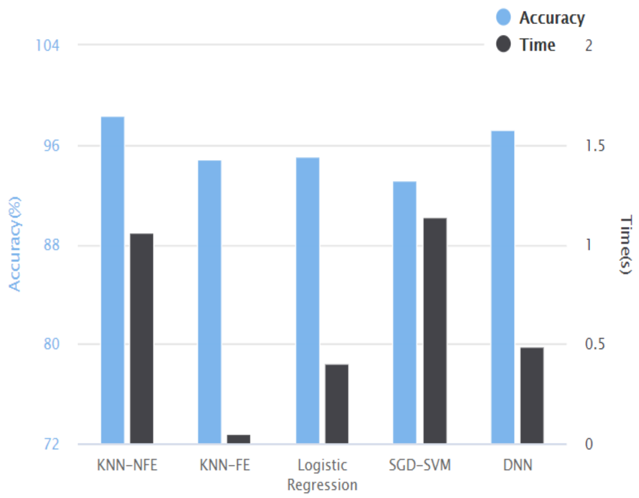


Figure 17: Accuracy and computation time of each algorithm

5. Conclusion

Through the above experimental analysis, the data window size represents the recognition speed of the model, and the data set without feature extraction can achieve good recognition effect with the kNN algorithm, and the accuracy can reach 96.13% when only 2 seconds of data are acquired, but its calculation time is longer. After feature extraction, the computation time of kNN is reduced by an order of magnitude, but the accuracy is decreased. Using the logistic regression algorithm for attitude recognition can improve the recognition rate without increasing the computation time, and the recognition rate reaches 95.09% when the sensor data of 4 seconds is acquired.

The logistic regression model outperforms the stochastic gradient descent SGD classifier using a linear support vector machine classifier in terms of recognition rate and computation time, and improves the recognition rate compared to the feature-extracted kNN model. In addition, the fully connected neural network model has a similar recognition rate and less computation time compared to the KNN-NFE, which has the highest recognition rate. Therefore, combining recognition accuracy and time efficiency, and considering that it does not have any requirement on the size of the input data, so it is able to guarantee the integrity of the information, fully connected neural networks are superior for gesture recognition.

Acknowledgements: This work has received funding from the Key Laboratory Foundation of National Defence Technology under Grant 61424010208, National Natural Science Foundation of China (No. 41911530242, 41975142), 5150 Spring Specialists (05492018012, 05762018039), Major Program of the National Social Science Fund of China (Grant No.17ZDA092), 333 High-Level Talent Cultivation Project of Jiangsu Province (BRA2018332), Royal Society of Edinburgh, UK and China Natural Science Foundation Council (RSE Reference: 62967_Liu_2018_2) under their Joint International Projects funding scheme and basic Research Programs (Natural Science Foundation) of Jiangsu Province (BK20191398).

References

[1] R. Alazrai, M. Mohammad, D. Mohammad. Fall detection for elderly from partially observed depth-map video sequences based on view-invariant human activity representation, *Applied Sciences*, **2017**, *vol. 7*, no. 4, pp: 316-323.

[2] V. Arya, S. Kumar. A novel TODIMVIKOR approach based on entropy and JensenTsalli divergence measure for picture fuzzy sets in a decision-making problem, *International Journal of Intelligent Systems*, **2020**, *vol. 35*, no. 12, pp: 2140-2180.

[3] A. Campagner, V. Dorigatti, D. Ciucci. Entropybased shadowed set approximation of intuitionistic fuzzy sets, *International Journal of Intelligent Systems*, **2020**, *vol. 35*, no. 12, pp: 2117-2139.

[4] L. C. K. Chin, K. S. Eu, T. T. Tay, C. Y. Teoh, K. M. A. Yap. Posture Recognition Model Dedicated for Differentiating between Proper and Improper Sitting Posture with Kinect Sensor, 2019 IEEE International Symposium on Haptic, *Audio and Visual Environments and Games*, **2019**, pp: 1-5.

[5] W. Ding, K. Liu, H. Chen, F. Tang. Human action recognition using similarity degree between postures and spectral learning, *Iet Computer Vision*, **2018**, *vol. 12*, no. 1, pp: 110-117.

[6] K. Dohyung, K. Dong-Hyeon, K. Keun-Chang. Classification of K-Pop Dance Movements Based on Skeleton Information Obtained by a Kinect Sensor, *Sensors*, **2017**, *vol. 17*, no. 6, pp: 12-21.

[7] M. E. A. Elforaici, I. Chaaraoui, W. Bouachir, Y. Ouakrim, N. Mezghani. Posture recognition using an RGB-D camera : exploring 3D body modeling and deep learning approaches, *IEEE life sciences conference*, **2018**, pp: 69-72.

[8] N. Feng, W. Song, J. Han, J. Xie, Y. Gao, L. Song. LoRa Posture Recognition System Based on Multi-Source Information Fusion, *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, **2019**, pp: 895-902. DOI 10.1109.

[9] S. Gaglio, L. R. Giuseppe, M. Morana. Human Activity Recognition Process Using 3-D Posture Data, *IEEE Transactions on Human-Machine Systems*, **2017**, *vol. 45*, no. 5, pp: 586-597.

[10] R. Gravina, P. Alinia, H. Ghasemzadeh, G. Fortino. Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges, *Information Fusion*, **2017**, no. 35, pp: 68-80.

[11] J. Gunnar. Visual perception of biological motion and a model for its analysis, *Perception & Psychophysics*, **1973**, no. 14, pp: 201-211.

[12] Q. Hu, X. Tang, W. Tang. A Smart Chair Sitting Posture Recognition System Using Flex Sensors and FPGA Implemented Artificial Neural Network, *IEEE Sensors Journal*, **2020**, *vol. 20*, no.14, pp: 8007-8016.

[13] J. Jin, L. Zhao, M. Li, F.Yu, Z. Xi. Improved zeroing neural networks for finite time solving nonlinear equations, *Neural Computing and Applications*, **2020**, *vol. 32*, no. 9, pp: 4151-4160.

[14] A. Kamel, B. Sheng, P. Yang, P. Li, R. Shen, D. D. Feng. Deep Convolutional Neural Networks for Human Action Recognition Using Depth Maps and Postures, *IEEE Transactions on Systems Man & Cybernetics Systems*, **2018**, pp: 1-14.

[15] H. G. Kang, S. H. Lee. Human body posture recognition with discrete cosine transform, *International Conference on Big Data & Smart Computing, IEEE*, **2016**, pp: 423-426.

[16] A. Kitzig, E. Naroska, G. Stockmanns, R. Viga, A. Grabmaier. A novel approach to creating artificial training and test data for an HMM based posture recognition system, *IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, **2016**, pp: 1-6.

[17] R. Kiwon, S. Hyun-Chool. Finger Motion Recognition Robust to Diverse Arm Postures Using EMG and Accelerometer, *2018 International Conference on Information Networking*, **2018**, pp: 834-836. DOI: 10.1109/ICOIN.2018.8343237.

[18] F. Lin, A. Wang, L. Cavuoto, W. Xu. Toward Unobtrusive Patient Handling Activity Recognition for Injury Reduction Among At-Risk Caregivers, *IEEE Journal of Biomedical and Health Informatics*, **2017**, *vol. 21*, no. 3, pp: 682-695.

[19] Z. Liouane, T. Lemlouma, P. Roose, F. Weis, H. Messaoud. An improved elman neural network for daily living activities recognition, *International Conference on Intelligent Systems Design and Applications*, **2016**, pp: 697-707.

[20] X. Liu, H. Liu, Y. Lin. Video frame interpolation via optical flow estimation with image inpainting, *International Journal of Intelligent Systems*, **2020**, *vol. 35*, no. 12, pp: 2087-2102.

[21] S. Ma, W. H. Cho, C. H. Quan, S. Lee. A sitting posture recognition system based on 3 axis accelerometer, *Computational Intelligence in Bioinformatics & Computational Biology IEEE*, **2016**, pp: 1-3.

[22] R. Munoz, T. S. Barcelos, R. Villarroel, R. Guinez, E. Merino. Body Posture Visualizer to Support Multimodal Learning Analytics, *IEEE Latin America Transactions*, **2018**, *vol. 16*, no. 11, pp: 2706-2715.

[23] S. Neili, S. E. Gazzah, M. A. Yacoubi, N. E. B. Amara. Human posture recognition approach based on ConvNets and SVM classifier, *International Conference on Advanced Technologies for Signal & Image Processing. IEEE*, **2017**, pp: 1-6.

[24] H. F. Nweke, Y. W. Teh, G. Mujtaba, U. R. Alo, M. A. Al-garadi. Multisensor fusion based on multiple classifier systems for human activity identification, *Human-centric Computing and Information Sciences*, **2019**, *vol. 9*, no. 1, pp: 9-34.

[25] A. D. Paola, P. Ferraro, S. Gaglio, G. L. Re. Context-awareness for multisensor data fusion in smart environments, AI*IA 2016 Advances in Artificial Intelligence, *Springer International Publishing*, **2016**, pp: 377-391.

[26] J. Permatasari, T. Connie, T. S. Ong. Inertial sensor fusion for gait recognition with symmetric positive definite Gaussian kernels analysis, *Multimedia Tools and Applications*, **2020**, no. 1, pp: 1-28.

[27] I. P. E. S. Putra, J. Brusey, E. Gaura, R. Vesilo. An event-triggered machine learning approach for accelerometer-based fall detection, *Sensors*, **2018**, *vol. 18*, no. 1, pp: 20-31.

[28] P. A. Rafael, A. Raposo, H. Fuks. Using Foot and Knee Movement and Posture Information to Mitigate the Probability of Injuries in Functional Training, Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management, *Human Body and Motion*, **2019**, no. 1, pp: 153-169.

[29] W. Ren, O. Ma, H. Ji, X. Liu. Human Posture Recognition Using a Hybrid of Fuzzy Logic and Machine Learning Approaches, *IEEE Access*, **2020**, no. 99, pp: 1-8.

[30] D. Segarra, J. Caballeros, W. G. Aguilar, A. Sam, R. D. Martn. Orientation Estimation Using Filter-Based Inertial Data Fusion for Posture Recognition, *Computer-Based Analysis of the Stochastic Stability of Mechanical Structures Driven by White and Colored Noise*, **2019**, no. 1, pp: 220-233.

[31] W. Takano, L. Haeyeon. Action Description from 2D Human Postures in Care Facilities, *IEEE Robotics and Automation Letters*, **2020**, *vol. 5*, no. 2, pp: 774-781.

[32] H. Tang, H. S. Liu. Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion. *Neurocomputing*, **2019**, no. 331, pp: 424-433.

[33] A. Vecchio, F. Mulas, G. Cola. Posture recognition using the inter distances between wearable devices, *IEEE Sensors Letters*, **2017**, *vol. 1*, no. 4, pp: 1-4.

[34] F. Wang, L. Zhang, S. Zhou, Y. Huang. Neural network-based finite-time control of quantized stochastic nonlinear systems, *Neurocomputing*, **2019**, no. 362, pp: 195-202.

[35] J. W. Wang, N. T. Le, C. C.Wang, J. S. Lee. Hand Posture Recognition Using a Three-Dimensional Light Field Camera, *IEEE Sensors Journal*, **2016**, *vol. 16*, no. 11, pp: 4389-4396.

[36] K. Wang, Z. Ma, S. Chen, J. Yang, K. Zhou, T. Li. A benchmark for clothes variation in person reidentification, *International Journal of Intelligent Systems*, **2020**, *vol. 35*, no. 12, pp: 1881-1898.

[37] Z. Wang, X. Shi, J. Wang, F. Gao, J. Li, M. Guo, S. Qiu. Swimming Motion Analysis and Posture Recognition Based on Wearable Inertial Sensors, *2019 IEEE International Conference on Systems, Man and Cybernetics(SMC)*, **2019**, pp: 3371-3376.

[38] X. Xu, F. Lin, A. Wang, Y. Hu, M. C. Huang, W. L. Xu. Body-Earth Movers Distance: A Matching-Based Approach for Sleep Posture Recognition, *IEEE Transactions on Biomedical Circuits and Systems*, **2016**, *vol. 10*, no. 5, pp: 1023-1035.

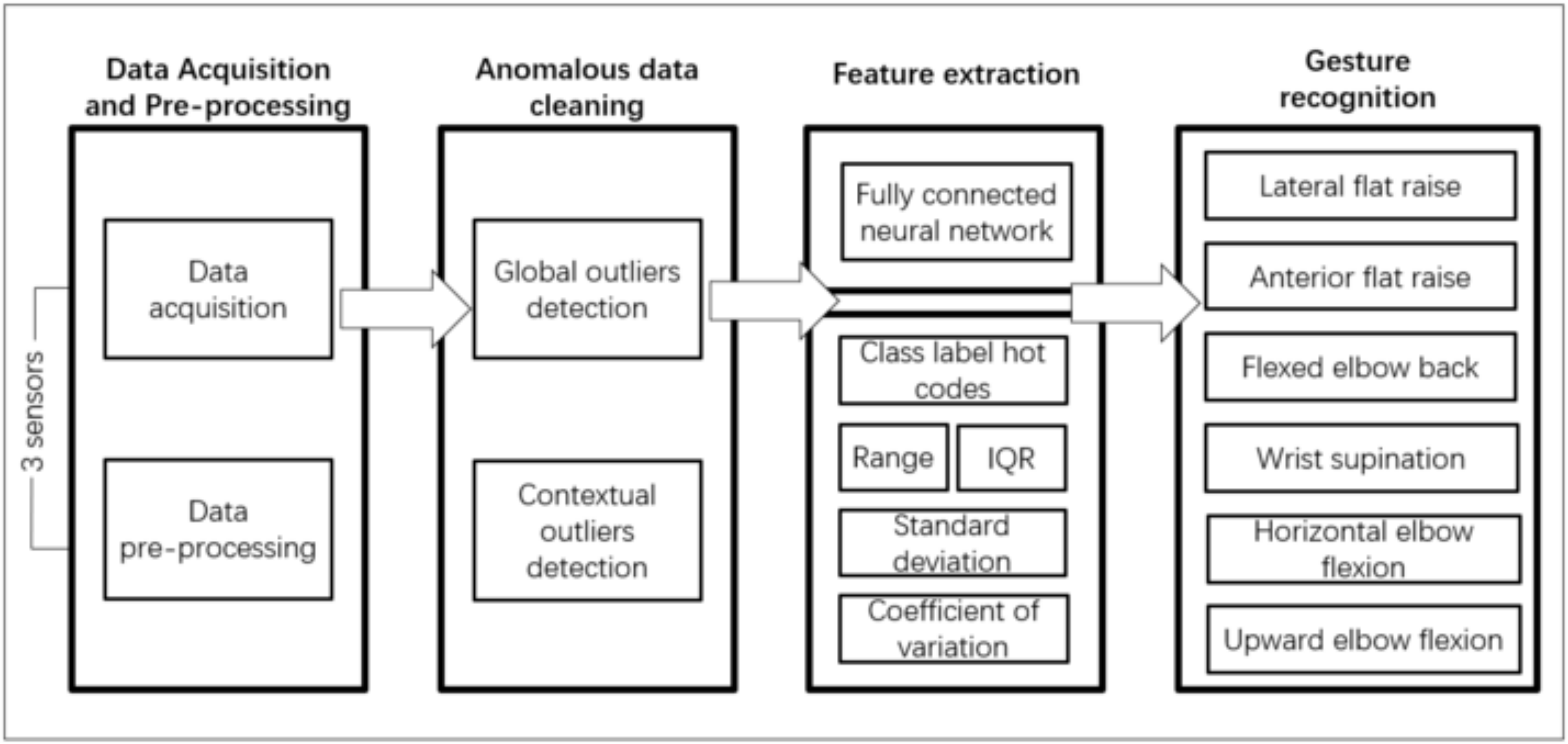
[39] F. Yu, L. Liu, L. Xiao, K. Li, S. Cai. A robust and fixed-time zeroing neural dynamics for computing time-variant nonlinear equation using a novel nonlinear activation function, *Neurocomputing*, **2019**, no. 350, pp: 108-116.

[40] M. Zadghorban, M. Nahvi. An algorithm on sign words extraction and recognition of continuous Persian sign language based on motion and shape features of hands, *Pattern Analysis & Applications*, **2016**, no. 21, pp: 323-335.

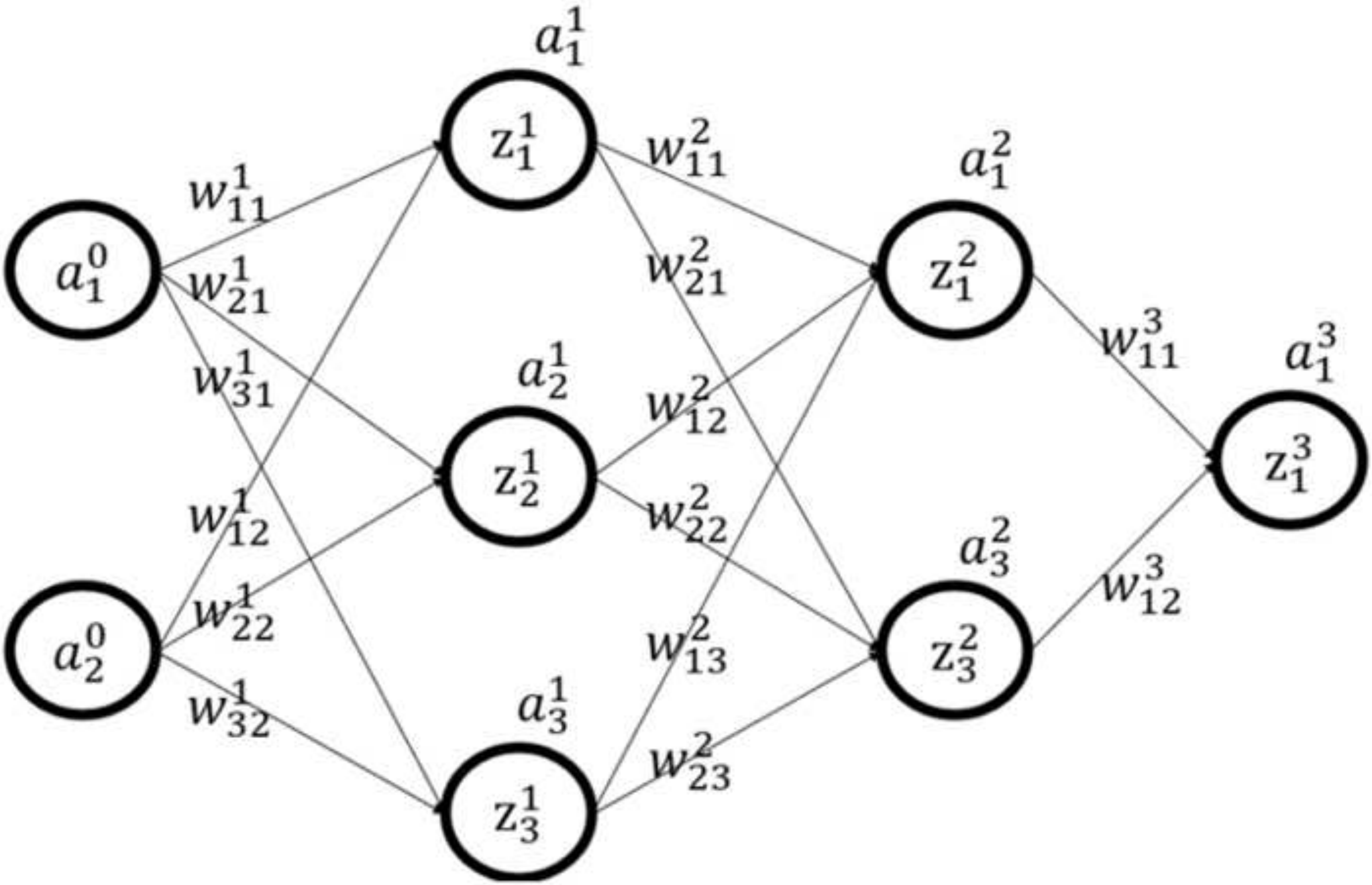
[41] T. Zebin, P. J. Scully, K. B. Ozanyan. Inertial Sensor Based Modelling of Human Activity Classes: Feature Extraction and Multi-sensor Data Fusion Using Machine Learning Algorithms, *eHealth 360. Springer International Publishing*, **2017**, pp: 306-314.

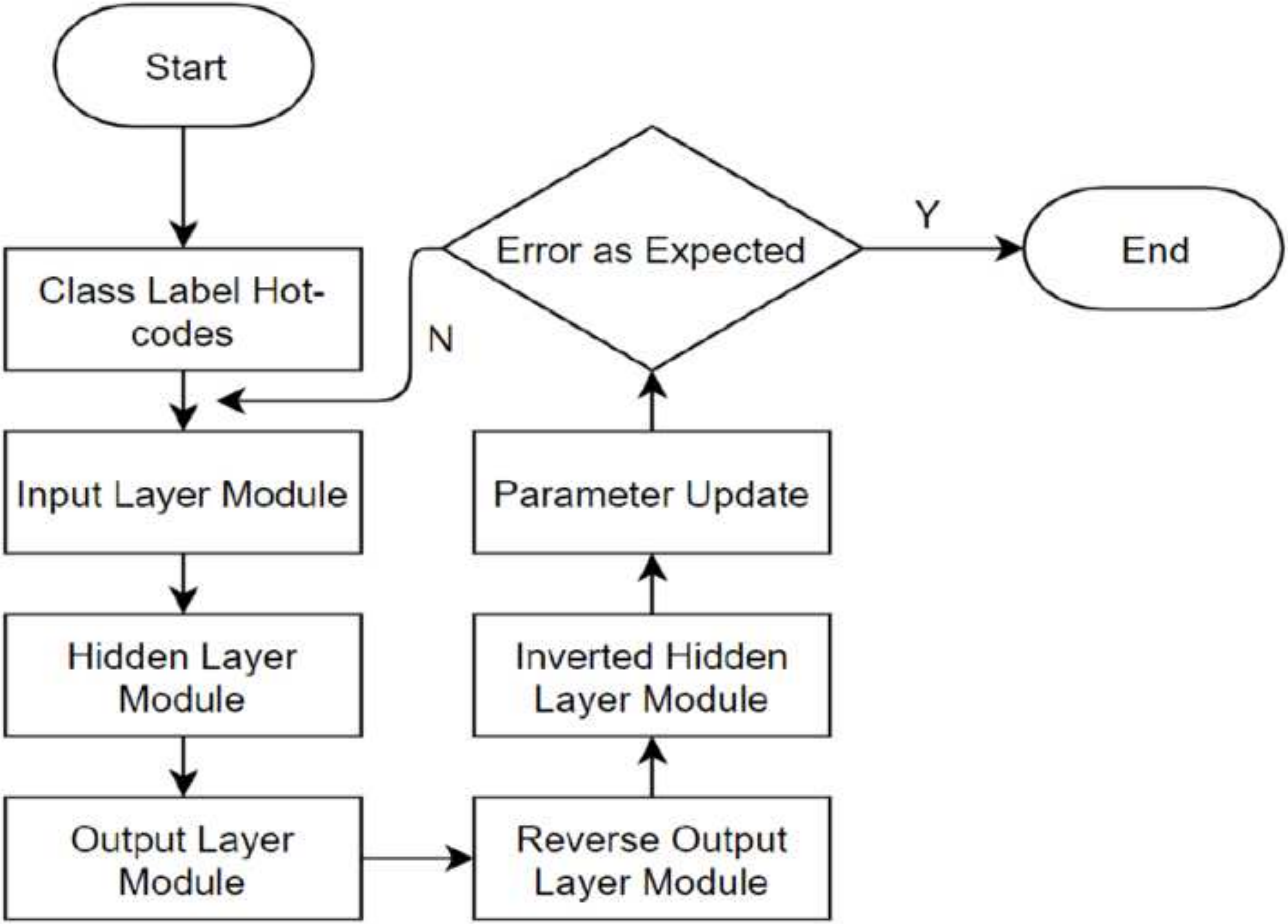
[42] A. Zhao, J. Dong, H. Zhou. Self-Supervised Learning From Multi-Sensor Data for Sleep Recognition, *IEEE Access*, **2020**, *vol. 8*, pp: 93907-93921.

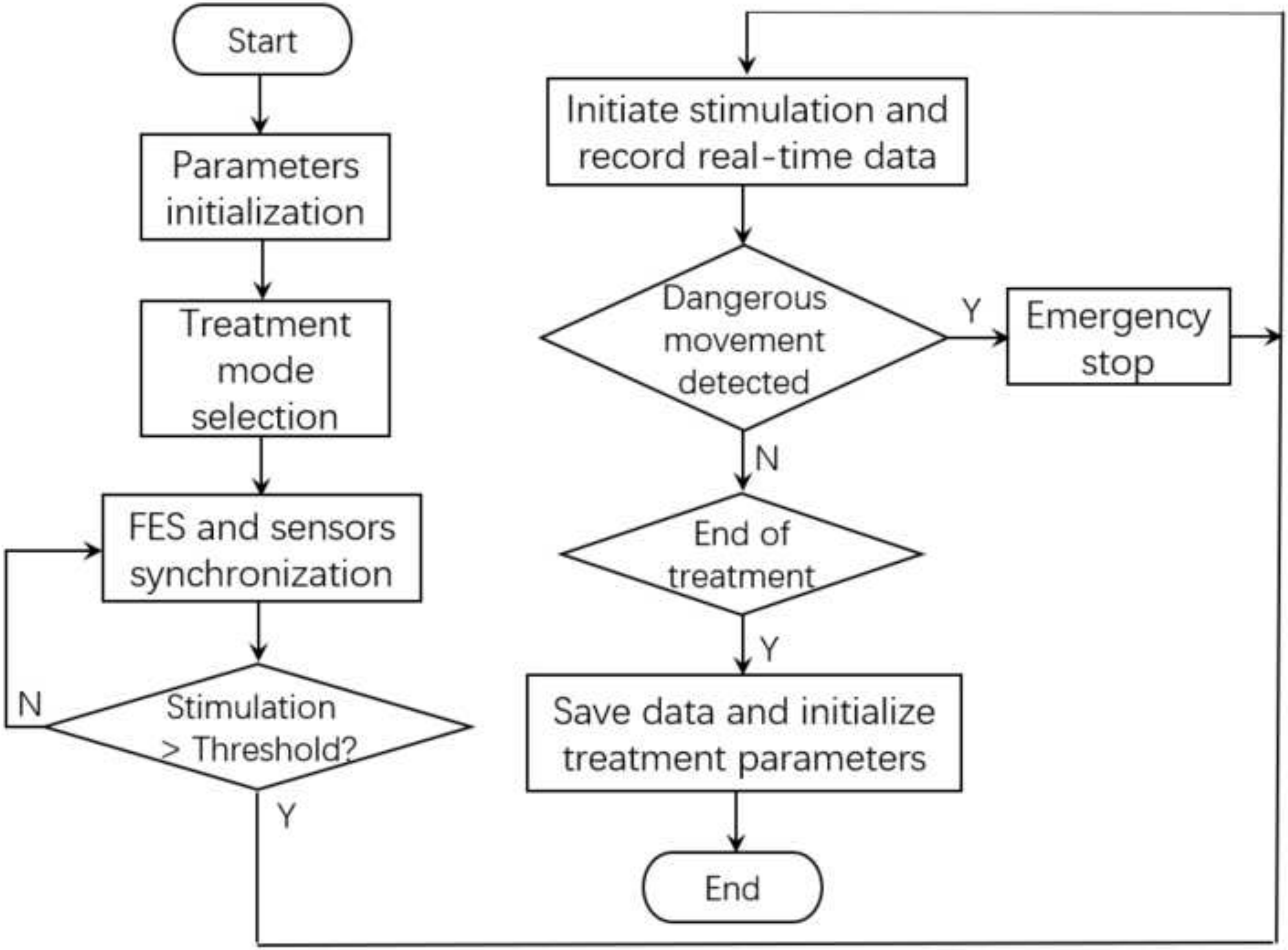
Click here to access/download;Figure;1.PNG

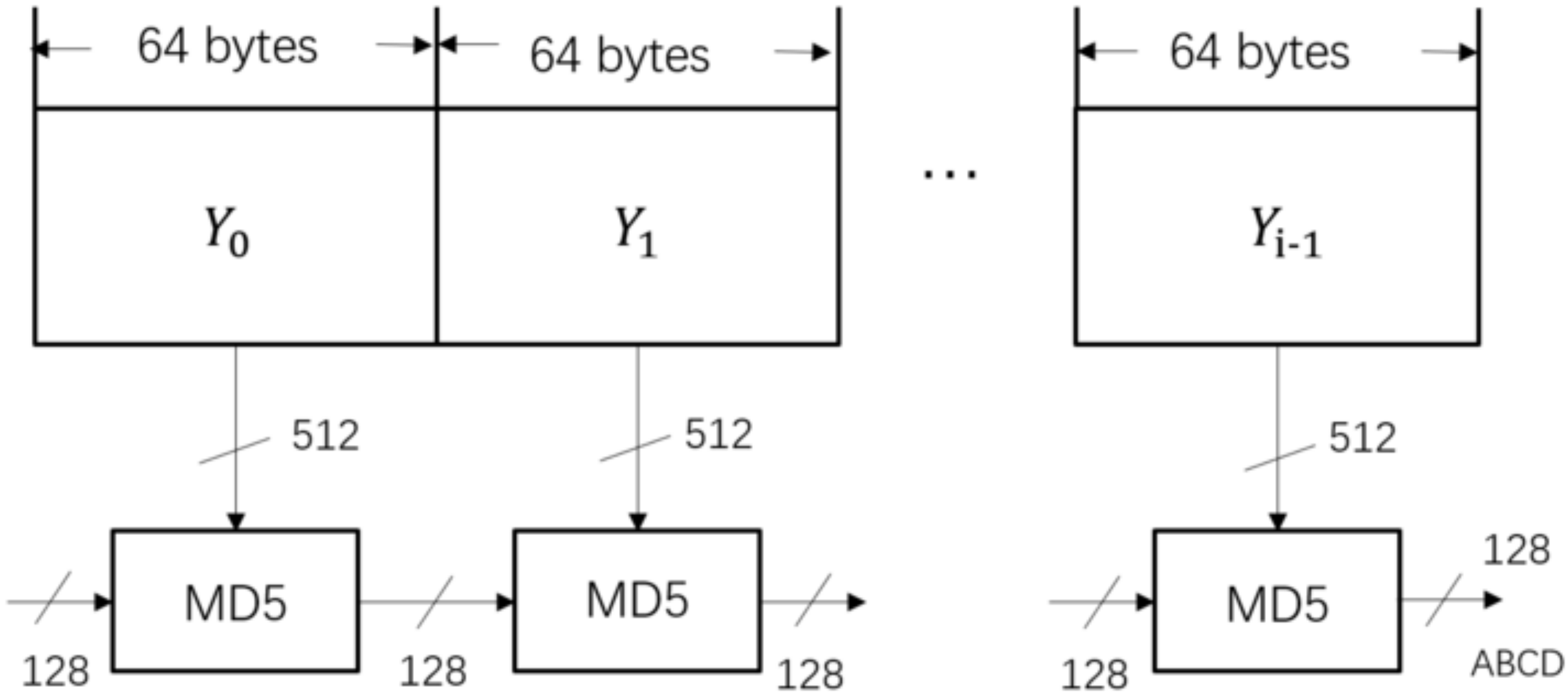


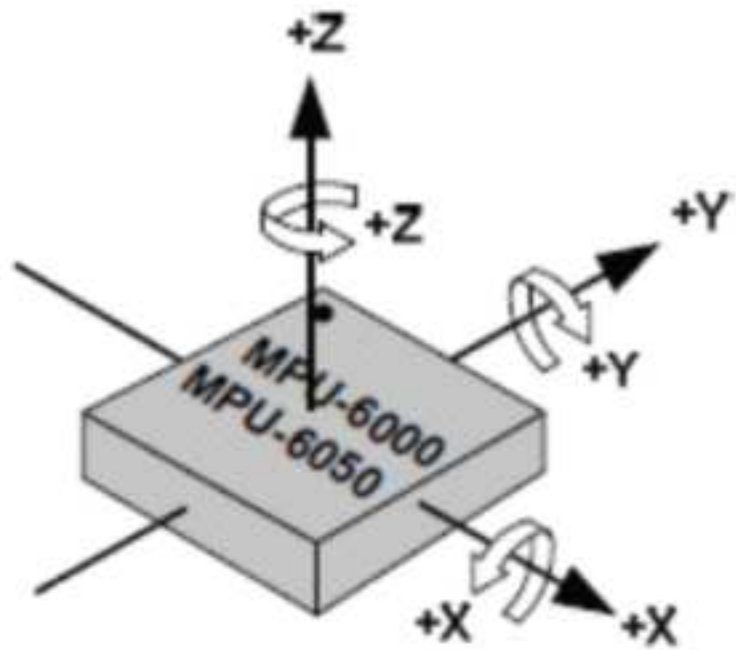
Click here to access/download;Figure;2.PNG

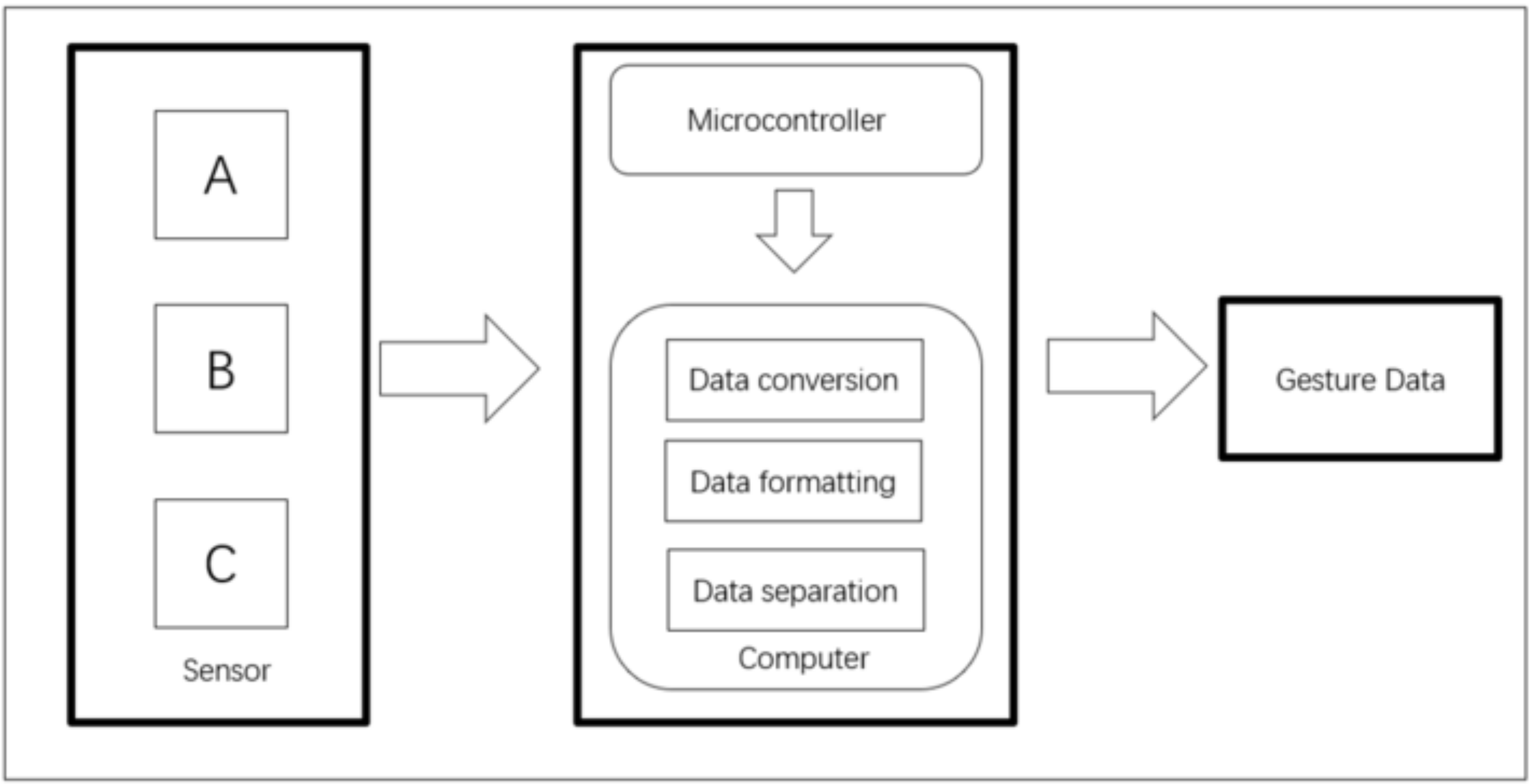


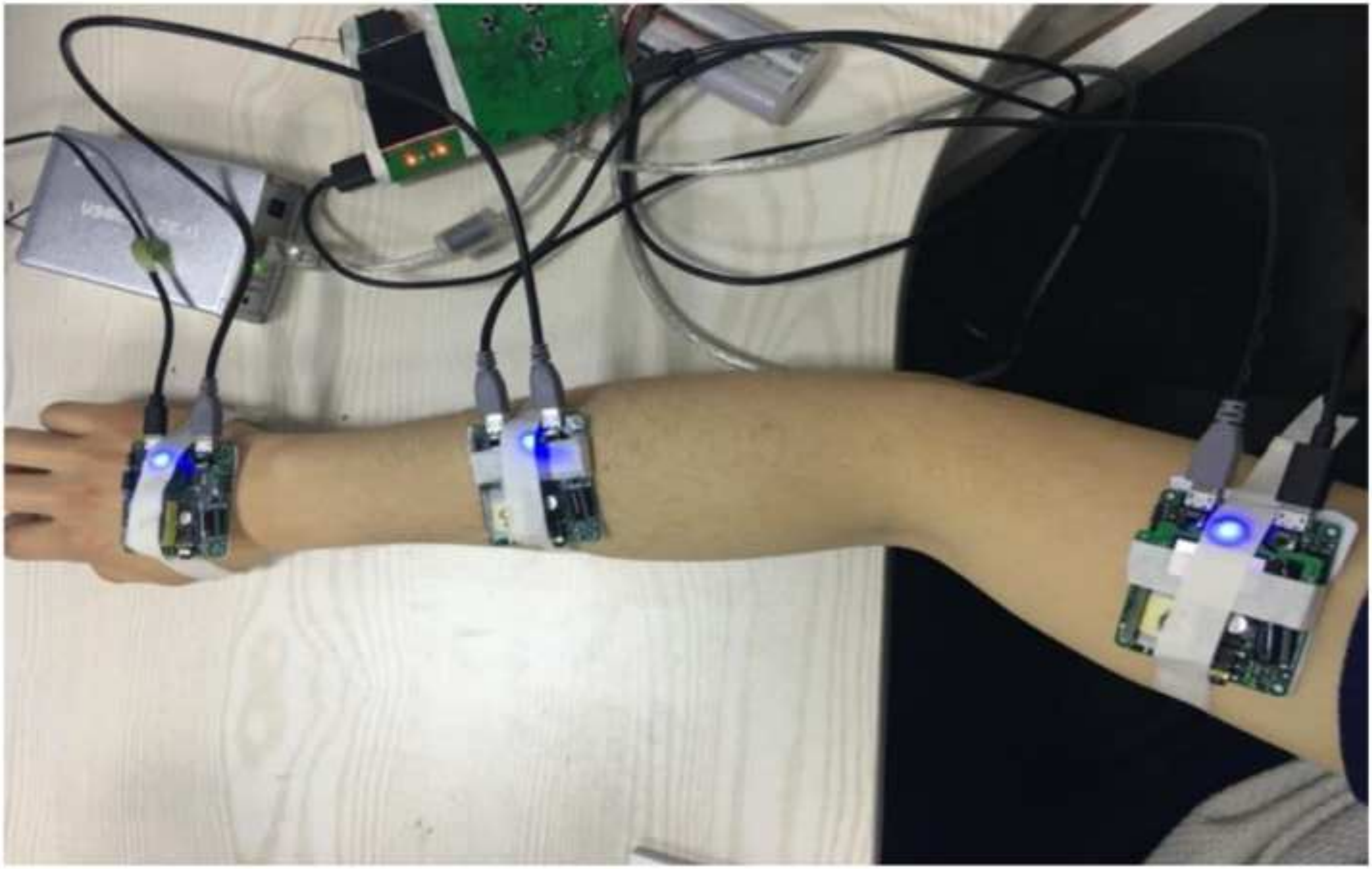

Start
Class Label Hot-codes
Input Layer Module
Hidden Layer Module
Output Layer Module
Reverse Output Layer Module
Inverted Hidden Layer Module
Parameter Update
Error as Expected
N
Y
End

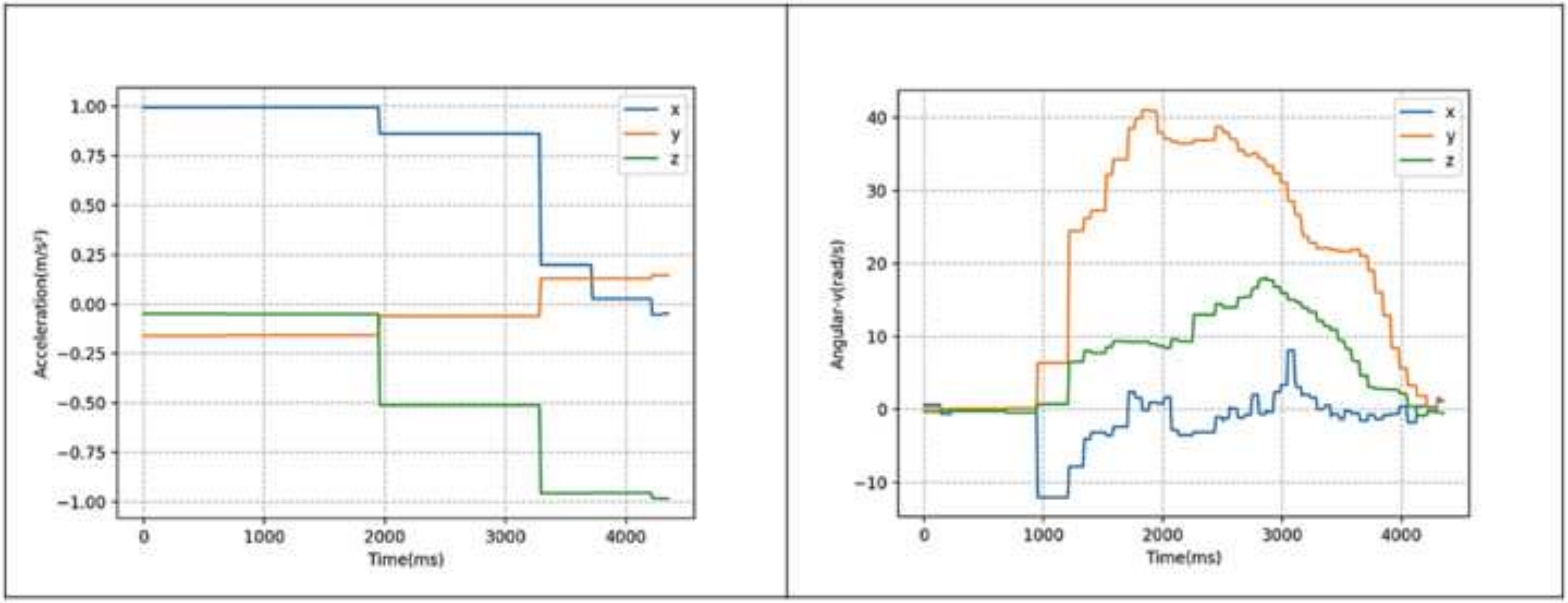

Start
Parameters initialization
Treatment mode selection
FES and sensors synchronization
Stimulation > Threshold?
N
Y
Initiate stimulation and record real-time data
Dangerous movement detected
Y
Emergency stop
N
End of treatment
Y
Save data and initialize treatment parameters
End

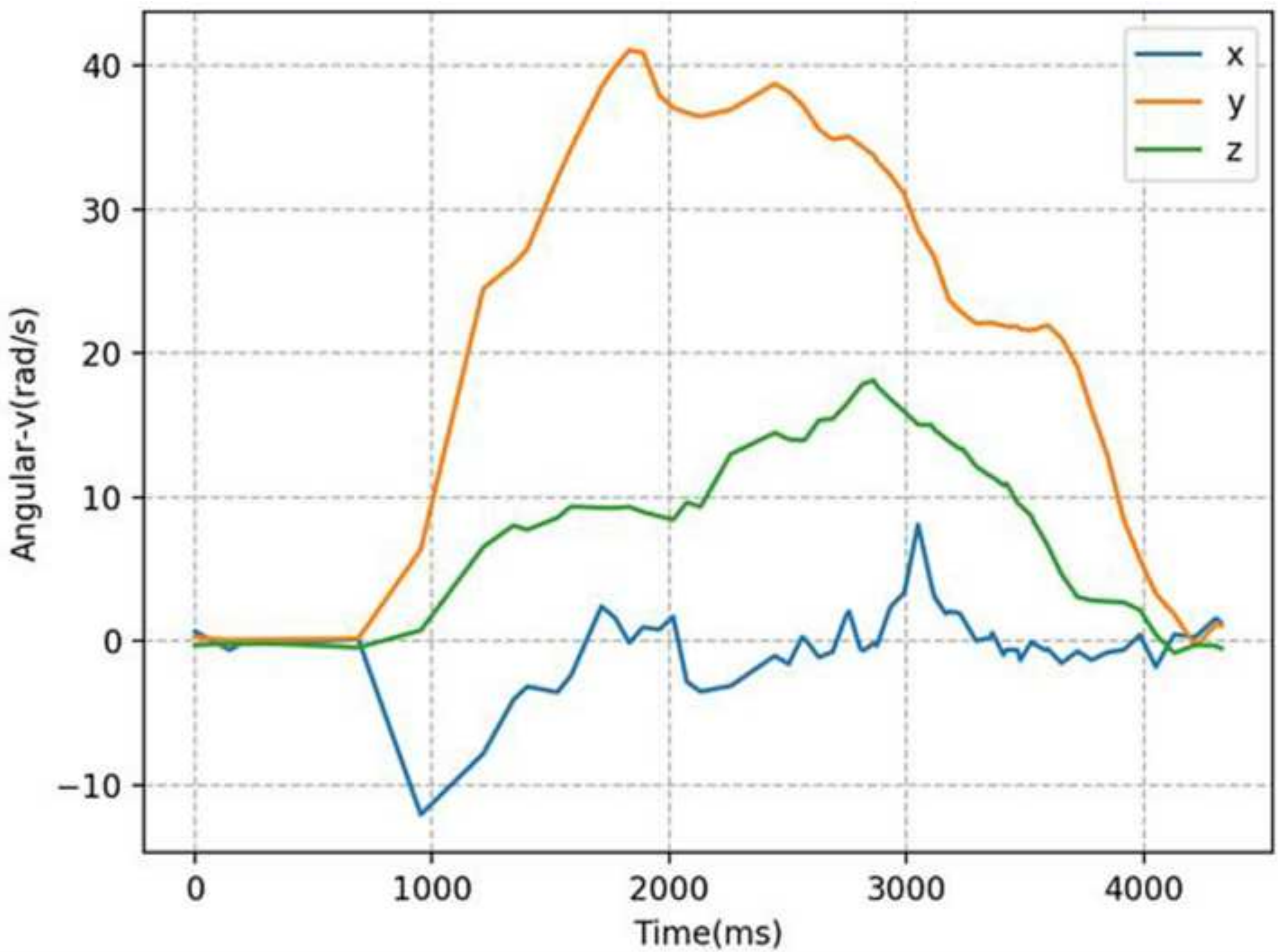

64 bytes
64 bytes
64 bytes
Y_0
Y_1
...
Y_{i-1}
512
512
512
MD5
MD5
MD5
128
128
128
128
128
ABCD


+Z
+Z
+Y
+Y
MPU-6000
MPU-6050
+X
+X


A
B
C
Sensor
Microcontroller
Data conversion
Data formatting
Data separation
Computer
Gesture Data

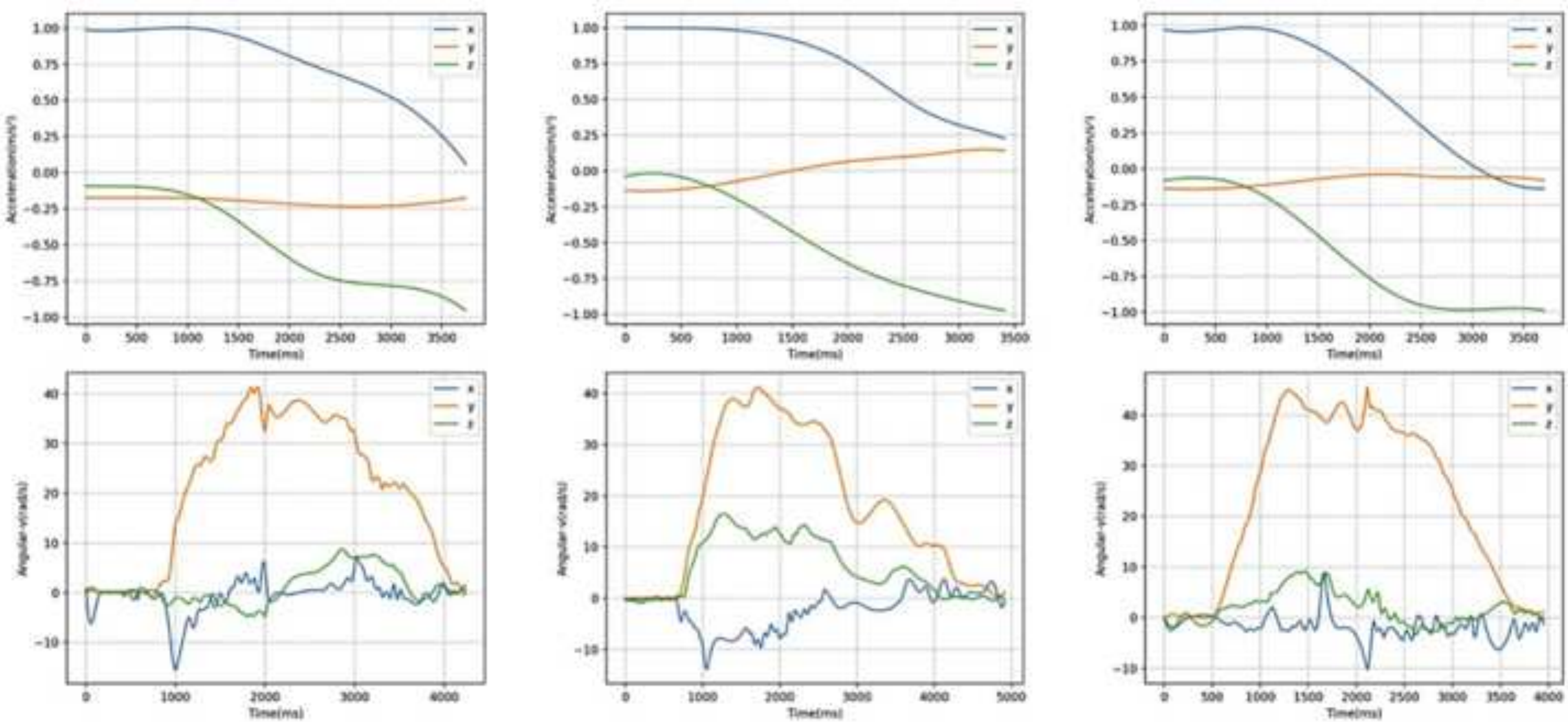


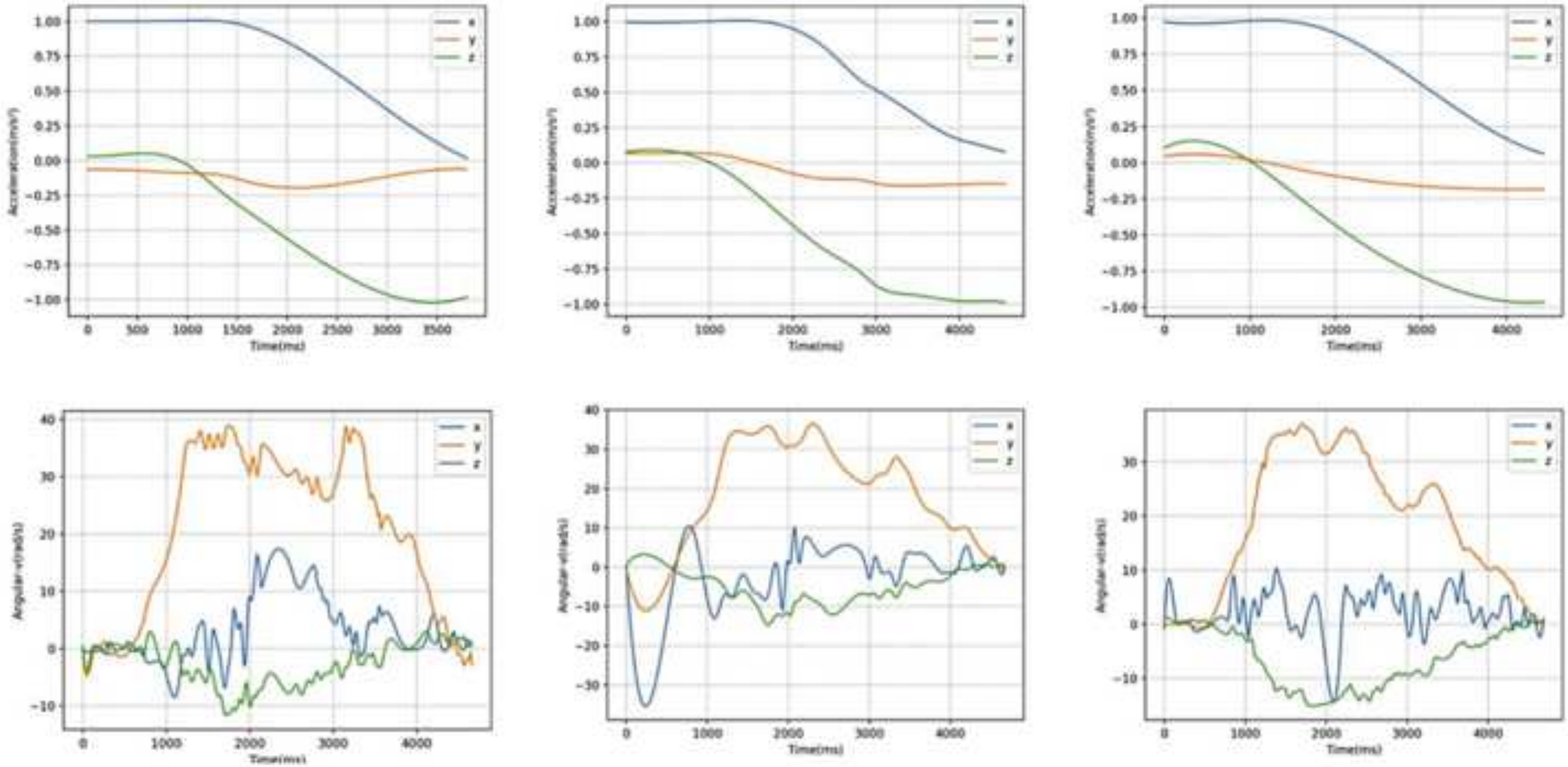

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Acceleration(m/s²)
0
1000
2000
3000
4000
Time(ms)
x
y
z
40
30
20
10
0
−10
Angular-v(rad/s)
0
1000
2000
3000
4000
Time(ms)
x
y
z

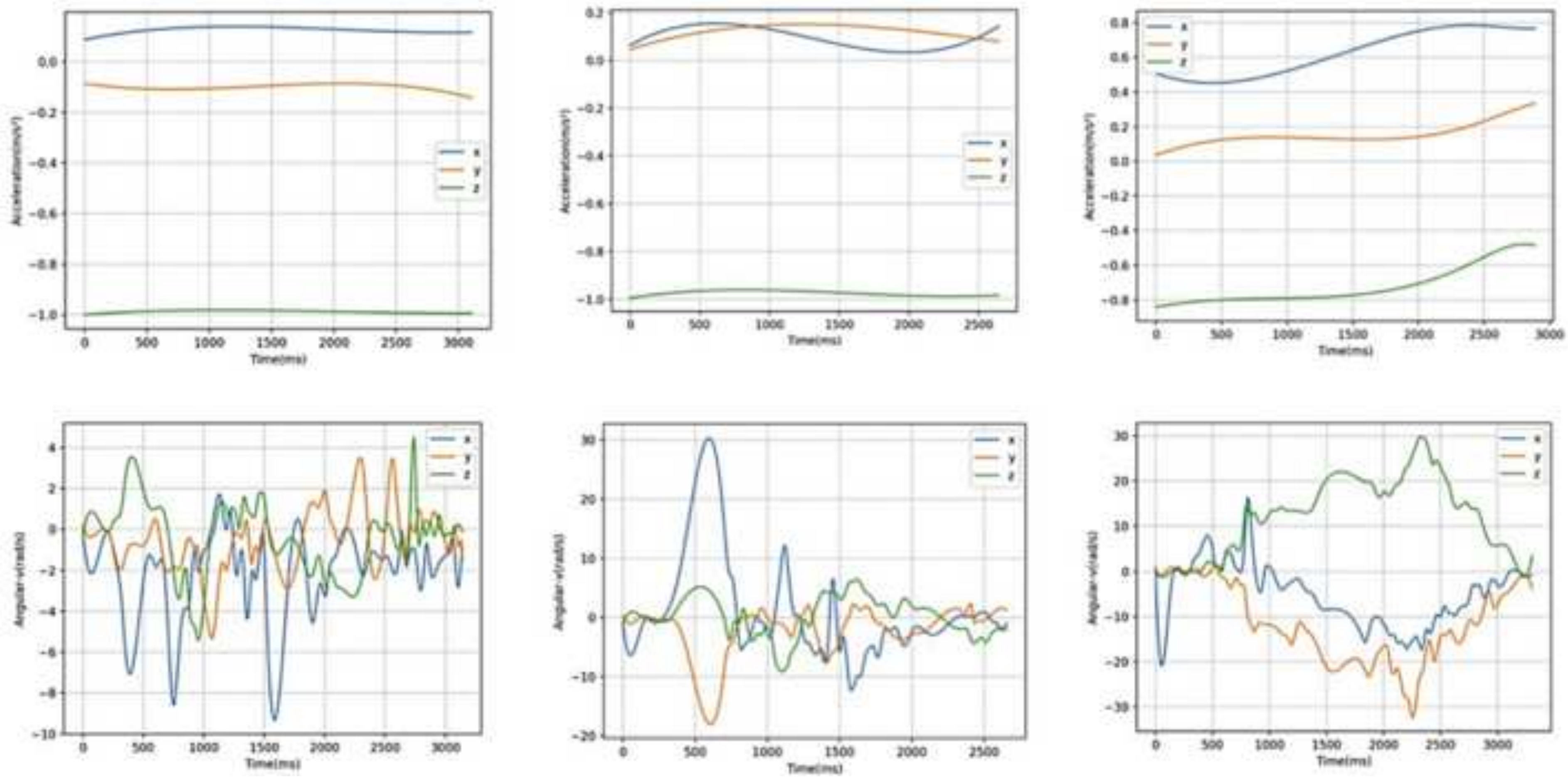


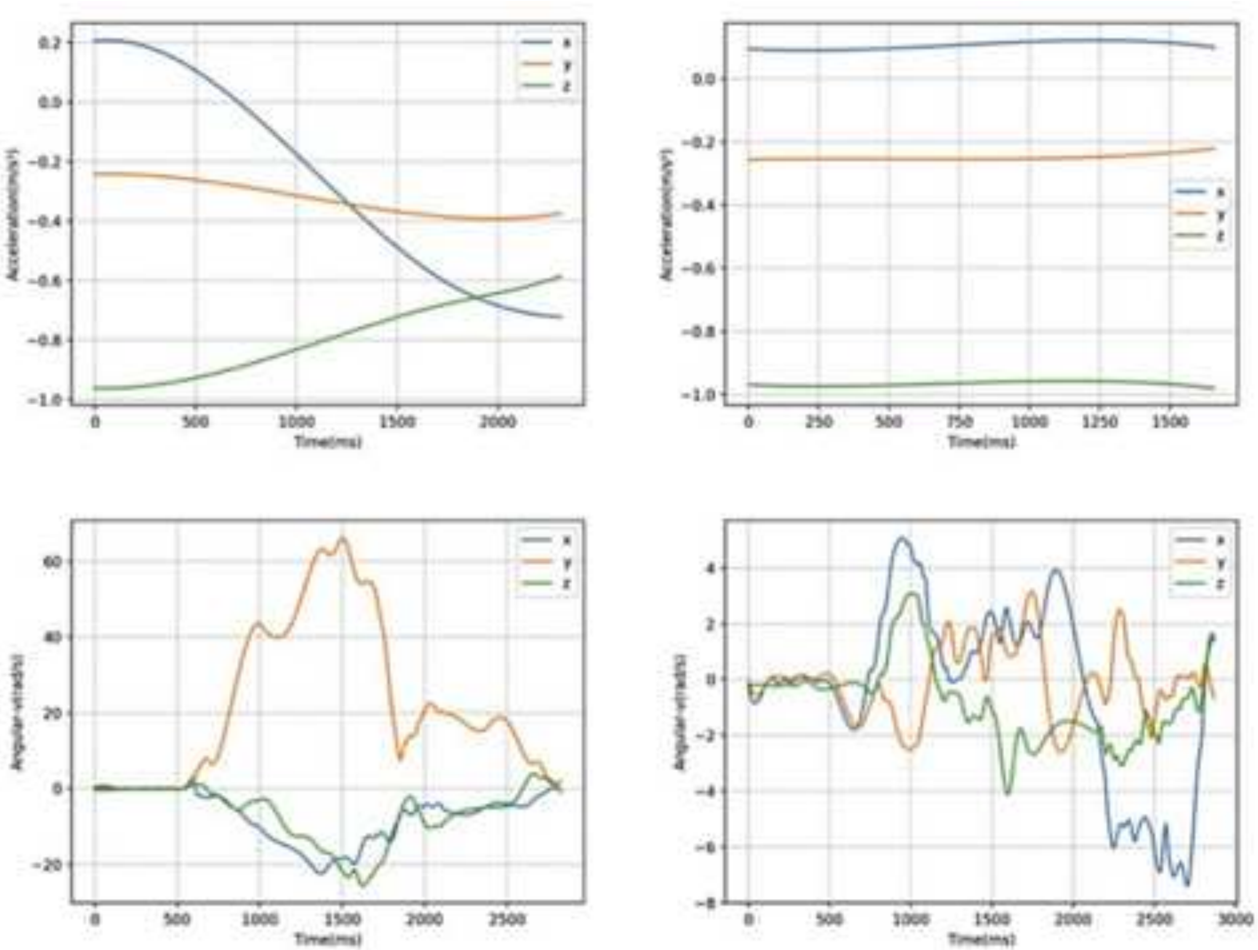
40
30
20
10
0
−10
Angular-v(rad/s)
0
1000
2000
3000
4000
Time(ms)
x
y
z

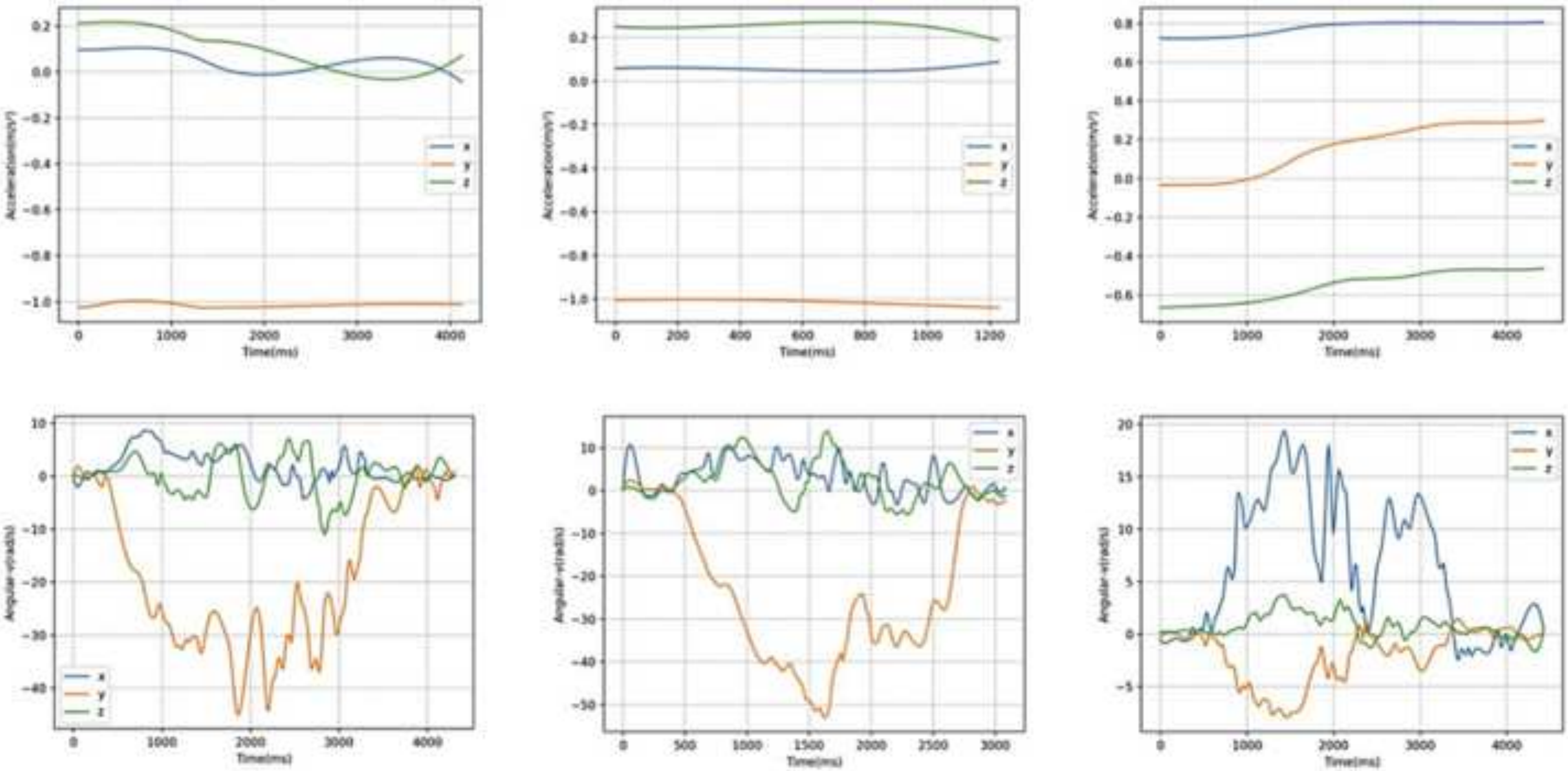
Click here to access/download;Figure;11.png

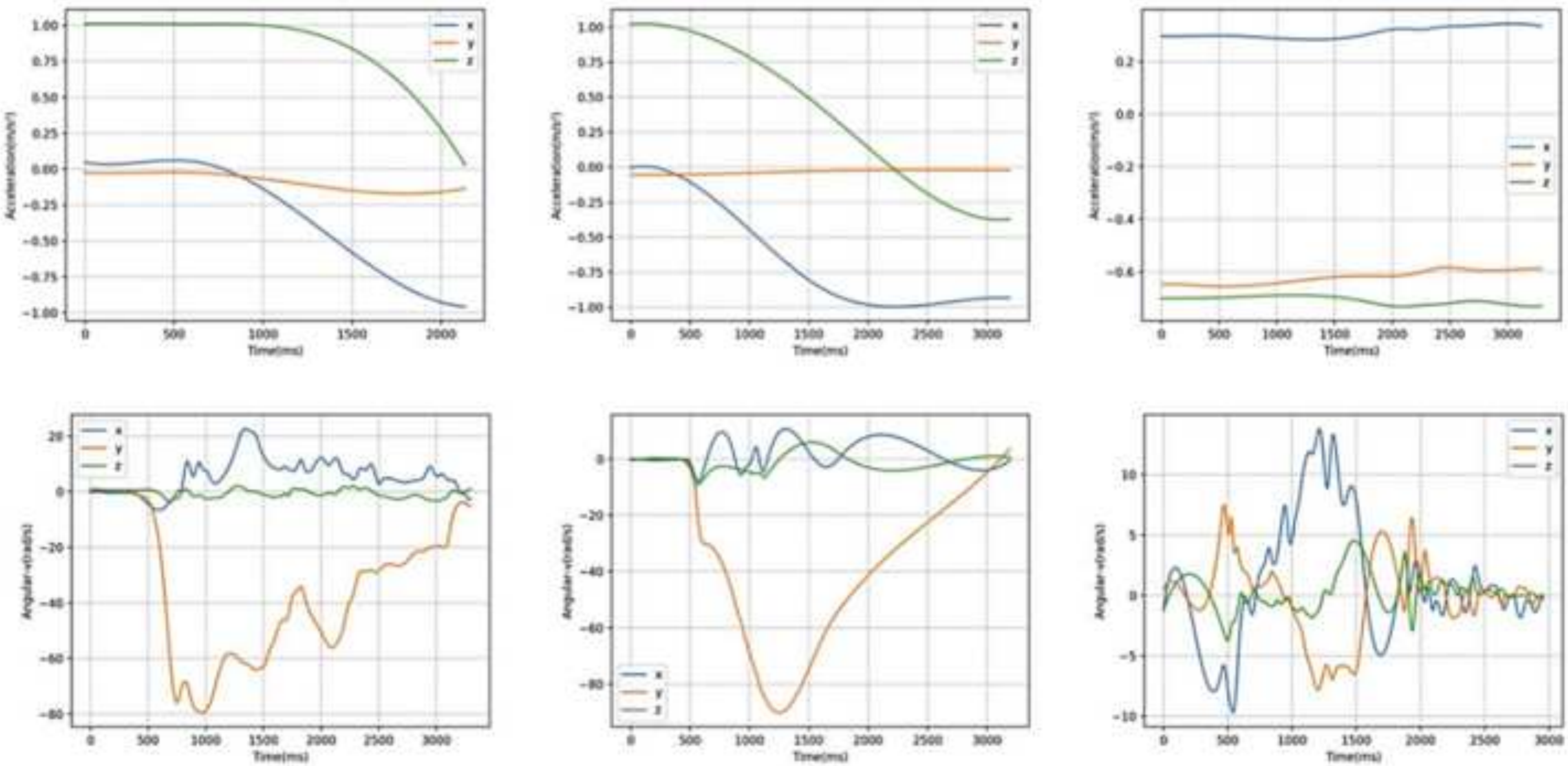



Acceleration(m/s²)
Time(ms)
Angular-v(rad/s)
x
y
z


Acceleration(m/s²)
Time(ms)
Angular-v(rad/s)
x
y
z

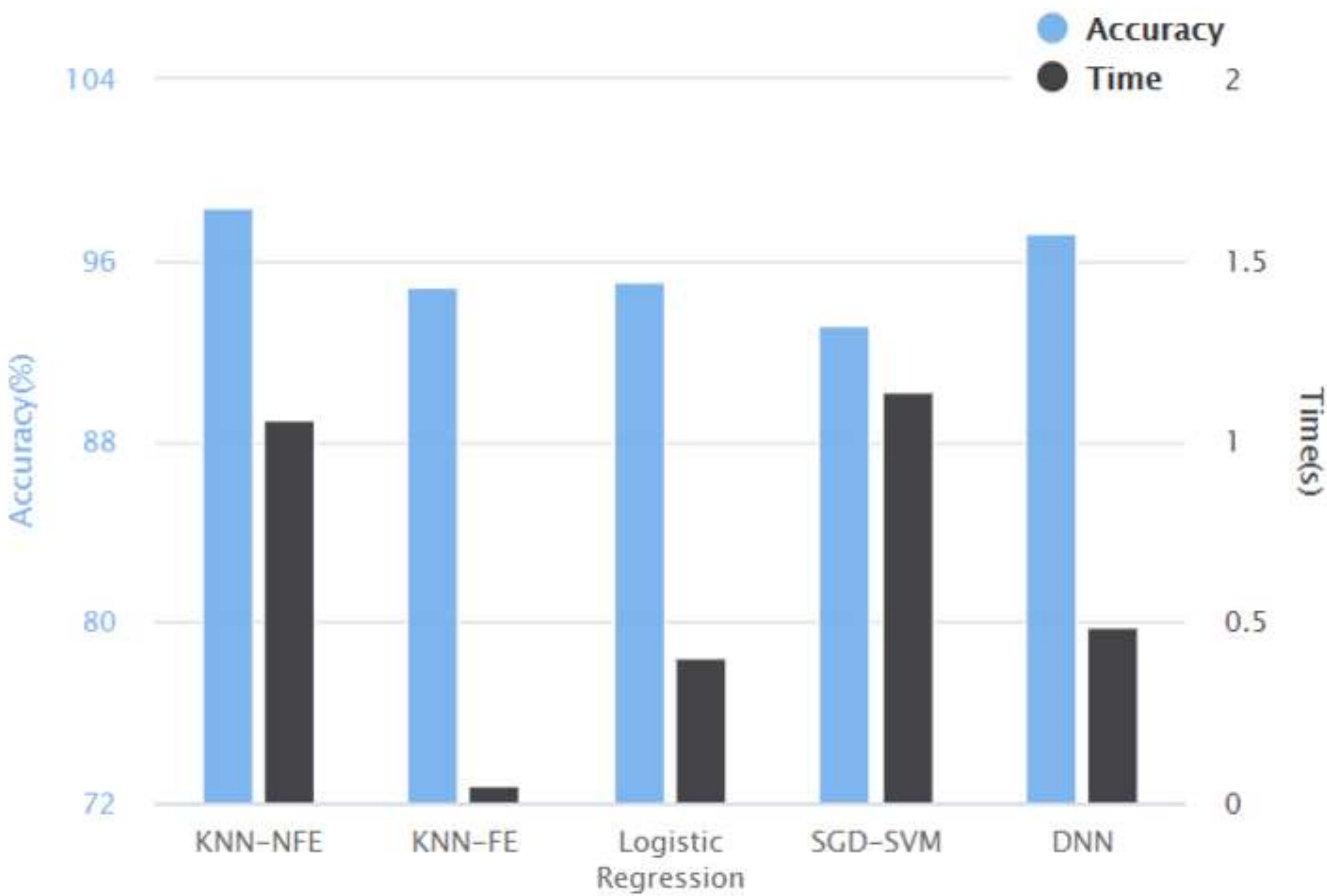

Acceleration(m/s²)
Time(ms)
Angular-v(rad/s)
x
y
z





Acceleration(m/s²)
Time(ms)
Angular-v(rad/s)
x
y
z

Click here to access/download;Figure;17.png



Response to reviewers

Click here to access/download

Supporting Information

A Fully-Connected DeepLearning_RtC.0129.docx