Received October 24, 2019, accepted November 9, 2019, date of publication November 15, 2019,
date of current version November 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953772

Recognizing Ping-Pong Motions Using Inertial Data Based on Machine Learning Classification Algorithms

HENG ZHANG, ZENGJUN FU, AND KUANG-I SHU
School of Computer and Information Science, Southwest University, Chongqing 400715, China

Corresponding author: Heng Zhang (dahaizhangheng@163.com)

This work was supported by the Major Transverse Project, China, under Grant SWU41015718 and Grant SWU20710953.

ABSTRACT With the development of Internet of Things (IoT) technology and various sensing technologies, some newer ways of perceiving people and the environment have emerged. Commercial wearable sensing devices integrate a variety of sensors that can play a significant role in motion capture and behavioral analysis. This paper proposes a solution for recognizing human motion in ping-pong using a commercial smart watch. We developed a data acquisition system based on the IoT architecture to obtain data relating to areas such as acceleration, angular velocity, and magnetic induction of the watch. Based on the features of the extracted data, experiments were performed using major machine learning classification algorithms including k-nearest neighbor, support vector machine, Naive Bayes, logistic regression, decision tree, and random forest. The results show that the random forest has the best performance, reaching a recognition rate of 97.80%. In addition, we designed a simple convolutional neural network to compare its performance in this problem. The network consists of two convolutional layers, two pooling layers, and two fully connected layers, and it uses data with no extracted features. The results show that it achieves an accuracy of 87.55%. This research can provide training assistance for amateur ping-pong players.

INDEX TERMS Smart watch, inertial sensor, motion recognition, table tennis, machine learning.

I. INTRODUCTION

The Internet of Things (IOT) refers to a network formed by combining various information sensing devices with the Internet. The purpose is to enable all objects or people to be remotely perceived or controlled. It is combined with the Internet to create a more intelligent system of production and life. As a new generation of information technology, the core of the IoT is people-oriented, providing people with more convenient and comfortable services. We can use a variety of sensing devices to perceive people's movements, locations, and environmental information, and through the construction of models we can complete the processing and analysis of data to help people make informed decisions. As a typical IoT device, smart watches integrate many sensors and have strong communication capabilities.

In recent years, some human-computer interaction techniques for somatosensory games [1] and assisted training [2] have appeared in sports. Participants can exercise by carrying out certain moves or jumping. Regardless of ball sports or other sports, accurate recognition of human motion is a key technology for virtual reality and human-computer interaction. There are several ways to recognize human motion, and the recognition method based on inertial sensors is an effective approach.

Some researchers have recognized human activity by placing sensors on parts of the human body. Zappi *et al.* [3] identified the worker's basic repair actions by placing 19 three-axis accelerometers on his arms, reaching a recognition rate of 98%. Hong *et al.* [4] fixed three-axis accelerometers to the thigh, waist and forearm of the subject, and studied the recognition of 18 kinds of daily movements by 15 subjects; the recognition rate reached 92.58%. Olguin and Pentland [5] fixed accelerometers to the right hand, left hip and chest of three subjects to identify sitting, running, squatting, walking, standing, crawling, and lying down, with the recognition rate reaching 92.13%. They used the sensors to recognize movements by placing them on areas of the human body that put a lot of extra burden on the users. In addition, there exist additional studies which have used accelerometers to recognize human motion, but their use of gyroscopes and magnetic field sensors is insufficient. It is of great practical

The associate editor coordinating the review of this manuscript and approving it for publication was Mu-Yen Chen.

significance to be able to comprehensively utilize a variety of sensors through a pervasive product.

With the development of MEMS, many high-precision MEMS inertial sensors are now being used in various types of wearable smart terminals [6]. Wearable sensing devices are currently appearing in numerous areas, including motion monitoring, and this has prompted new research.

In Ref. [7], the authors evaluated the accuracy of a host of the latest wearable devices in measuring fitness-related indicators under various semi-natural activities. They found that current mainstream devices are able to reliably measure heart rate, number of steps, distance, and sleep duration, which can be used as effective health evaluation indicators. In Ref. [8], the authors identified, selected and categorized the methodologies for estimating the ground reaction forces from IMUs as proposed across the years. They classified the identified papers as direct modeling-based methods and machine learning-based methods. In Ref. [9], the authors measured walking ground reactions in real-life environments, and found that methods using measured body kinematics exhibited the highest practicality of the three classes of methods reviewed. Wearable devices can use sensors to collect raw data from measurements which are stored and used for the continuous monitoring of health and exercise activity, as well as the assessment of performance, and others [10]. In Ref. [11], a novel wearable device for improving the activity recognition accuracy was proposed based on the different multiple sensors; this device simultaneously collects the muscle activity and motion information. Ref. [12] introduced a system performance analysis of human activity cognition of motion sensor behavior through a smartphone. When participants conducted daily human activities, the authors collected sensor data sequences through a smartphone. The ability to recognize human motion through a common commercial wearable integrated sensing device has some significant advantages, especially in situations with fast motion.

There are many commercial wearable devices available today. In Ref. [10], the authors described the wrist-wearable monitoring devices in relation to three aspects: main functionalities, sensors that are integrated within the device, and IT support. As an increasingly popular wearable product, smart watches have been welcomed by many people. Some researchers have used smart watches to identify human activities. Ref. [13] used an architecture that included an Apple Watch and an Apple TV remote to identify the user. They performed the classification based on the recognition of four types of human activities through building four databases. Ref. [14] employed the data of the acceleration sensor in the smart watch for behavior recognition. The present paper uses an Android smart watch to recognize human motion in table tennis. The advantages of such a solution are as follows: a. As a commercial device, it can be easily worn on the wrist by the user without being directly attached to the body; b. We can only use this one node, which avoids the trouble of wearing multiple nodes; c. With a wide range of sensors integrated into the watch, we are able to combine multiple sensor information fusions to improve recognition performance; d. Smart watches have a variety of ways to communicate, so they can be better integrated into the information environment and take advantage of the IoT and cloud computing. The present paper uses a smart watch to acquire inertial data related to the movement of the player, and then transmits it to the server for analysis in real time. It is a typical IoT architecture. The IoT technology employs a multi-modal approach to perceive human activity. In particular, this article uses a wearable device to perceive the skill movements of ping-pong players.

Table tennis is a wide-ranging ball game with many amateur table tennis fans around the world. In the training of table tennis, players are prone to change their body shape, which makes their movements deviate from the normative ones. Effective external guidance can play a greater positive role. The present paper aims to use some basic and more standardized skill movements as the training standard to provide assistance and reference for the training of amateur players. By accurately determining whether the skill movements in the player training are accurate, the players are provided with intelligent monitoring and guidance to improve their training performance. Based on the smart watch, the present paper analyzes and discusses the constructed skill recognition model.

In the related motion recognition theory, the machine learning classification algorithms have been widely used and have achieved some good results. The Bayes classifier was used in Ref. [3], and the decision tree classification algorithm was applied in Ref. [4]. In Ref. [15], Artificial the neural network classifier and the nearest neighbor algorithm were used to recognize human activities, and achieved recognition rates of 95.24% and 87.17%. The research in the present paper experiments with k-NN, support vector machine, Naive Bayes, logistic regression, decision tree, random forest and a specially designed convolutional neural network.

The main work of the present paper is summarized as follows:

1. Based on commercial smart watches, we designed a data acquisition system that can accurately collect the acceleration, angular velocity and magnetic induction of the device.

2. We designed the experiment of recognizing table tennis movements, collected the watch data of the players during the exercise, and completed the data processing.

3. We experimented with the main machine learning classification model and used the convolutional neural network model as a comparison to discuss the results and draw conclusions.

II. DATA COLLECTING AND PROCESSING

A. DESIGN OF DATA ACQUISITION SYSTEM

In our experiment, we used a common commercial Android smart watch with a variety of sensors. We employed an accelerometer, gyroscope, and a magnetic field sensor to obtain a total of nine axes of data, including acceleration, angular velocity, and magnetic field strength.

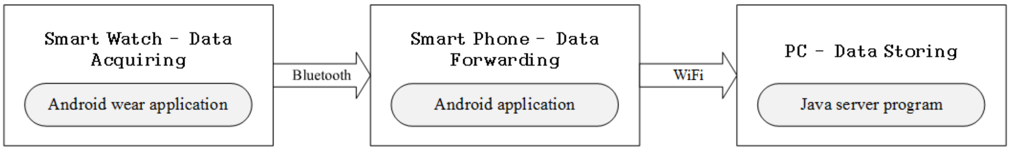


FIGURE 1. Structure of data acquisition system.

TABLE 1. Background information of subjects.

Subject number	Gender	Age(Y)	Height(cm)	Weight(kg)	right-handed
1	male	25	173	71.2	Yes
2	male	25	174	68.9	Yes
3	male	24	169	66.7	Yes
4	male	22	173	67.4	Yes
5	male	21	176	72.3	Yes
6	male	22	184	70.8	Yes
7	female	25	158	50.4	Yes
8	female	25	157	50.2	Yes
9	female	26	172	62.1	Yes
10	female	22	162	53.6	Yes
11	female	22	164	55.4	Yes
12	female	21	161	53.2	Yes

The data acquisition system is designed as follows: first, we developed an Android wear application to acquire real-time data of acceleration, angular velocity and magnetic field strength on the smart watch, and we transmitted this data to a mobile phone via Bluetooth. Following this, the data was received through an android application on the mobile phone, and at the same time it was forwarded to a PC via Wi-Fi. Finally, the data was received and stored on the PC by a specific Java server program, as shown in Figure 1. The sampling frequency can be set in this developed data acquisition system. After several tests, we set the data sampling frequency to 50 Hz, which satisfies our experimental needs and can transmit and receive data steadily. The system was developed using Android Studio, SDK, and JDK.

B. COLLECTING DATA RELATED TO TABLE TENNIS

Our research requires participants, and we recruited volunteers from the author's university to participate in the experiment. We tested a total of 12 college students in a week, including six males and six females. We gave all subjects guidance on ping-pong and trained them. All the subjects were amateur table tennis fans and their background information is shown in Table 1.

The actions we recognize are the eight basic movements in ping-pong. The names of these actions are: Forehand Attack, Forehand Drive, Forehand Chop, Forehand Flick, Backhand Control, Backhand Drive, Backhand Chop, and Backhand Flick.

Our subjects used the right hand to hold the table tennis bat. We asked the participants to wear the smart watch on the right wrist before they started the movement. After the equipment was positioned, the volunteers started to play ping-pong, and we recorded and saved the action data generated by the smart watch for a certain period of time. The button to start sending data is on the watch, which is clicked by the participant and can be initiated at any time. The termination of data acquisition was performed by the experimenter on the PC according to the actual situation.

TABLE 2. The number of samples for each action.

Action name	Male	Female	Sum
Forehand Attack (FA)	142	136	278
Forehand Drive (FD)	144	142	286
Forehand Chop (FC)	143	145	288
Forehand Flick (FF)	140	139	279
Backhand Control (BCL)	148	141	289
Backhand Drive (BDR)	145	143	288
Backhand Chop (BC)	143	137	280
Backhand Flick (BF)	142	145	287
Total	1147	1128	2275

The data we obtained was a continuous inertial signal, and we needed to detect its motion signal segment. A total of 2,275 valid samples were collected, of which 1,147 were from males and 1,128 were from females. Each action and the corresponding number of samples are shown in Table 2.

C. DATA PREPROCESSING

With regard to the raw data, including acceleration, angular velocity, and magnetic induction, we first smooth-filtered it, and then detected and segmented the motion signal segments.

1) SMOOTH FILTERING

In the process of data collection, some noise will inevitably be mixed, which will give rise to some interference with the features of the signal. We used the mean filtering method to smooth the original signal. The process was as follows: for the original signal sequence $(f_1, f_2, f_3, \ldots, f_n)$, in a time window of length, we calculated the average value of the M continuous data points $(f_{s-b}, \ldots, f_{s-1}, f_s, f_{s+1}, f_{s+b})$,

TABLE 3. Features and expressions.

Number	Feature name	Expression
1	Mean	$\frac{1}{L}\sum_{i=w_1}^{w_L} x_i$
2	Var	Variance, $\frac{1}{L}\sum_{i=w_1}^{w_L}(x_i - mean)^2$ 1
3	Std	Standard deviation, $\sqrt{var}$
4	Mode	Number of maximum occurrences in the signal segment
5	Maximum	Maximum value within the signal segment
6	Minimum	Minimum value within the signal
7	N_{oz}	Zero crossings, $\sum_{i=w_1}^{w_L} I(x_i > 0)$
8	range	range = $\mathrm{N_{max}} - \mathrm{N_{min}}$

where $b = (M-1)/2$, following which we took this average value $\sum_{i=s-b}^{s+b} f_i/M$ as the filtered output of point f_s.

2) ACTION DETECTION

The first step of recognition is to segment the action signal accurately. In the period of no action, the signal is stable and has a small variance, and within the action interval, the signal fluctuates greatly and has a large local variance. Therefore, we can set a sliding time window to detect and divide the action signal data by controlling the variance within the window. The specific process was as follows: first, we calculated the variance of each axis for each piece of sensor data in the window (Formula 2), and then compared the sum of each axis's variance (Formula 3) for each sensor with the set threshold (Formula 4). When they satisfied the constraint at the same time, we judged it as the window containing the action signal segment, where the overlap size is $w/3$.

The d-axis average:

$$Avg_d = \frac{1}{w}\sum_{i=1}^{w} d_i, \tag{1}$$

where, $d = \{x, y, z\}$.

The d-axis variance:

$$Var_d = \frac{1}{w}\sum_{i=1}^{w}(d_i - Avg_d)^2, \tag{2}$$

where, $d = \{x, y, z\}$.

The sum of the three axis variances:

$$S = Var_x + Var_y + Var_z, \tag{3}$$

Constraints of signal segmentation:

$$\begin{cases} th1_{acc} < S_{acc} < th2_{acc} \\ th1_{ang} < S_{ang} < th2_{ang} \\ th1_{mag} < S_{mag} < th2_{mag} \end{cases}, \tag{4}$$

where, $S_{acc}, S_{ang}, S_{mag}$ are the sum of the three axis variances corresponding to the accelerometers, gyros and magnetic field sensor. $th1_i, th2_i$ $(i = \{acc, ang, mag\})$ are the lower threshold and the upper threshold using the sum of the three axis variances for each sensor in a certain action signal segment as the reference for the threshold. For example, the sums of the three axis variances of acceleration, angular velocity and magnetic field strength calculated from the r_{th} action signal segment are $S_{acc}r$, $S_{ang}r$ *and* $S_{mag}r$, following which we set the corresponding $th1_i$, $th2_i$ to $a{*}S_ir$, $b * S_ir$, where $i = \{acc, ang, mag\}$, $a, b \in [0, 2]$ are the weight coefficients of upper threshold and lower threshold.

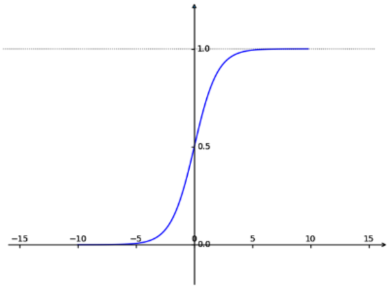


FIGURE 2. Sigmoid function.

D. FEATURE EXTRACTING

Since the time domain features have less computational complexity and can better meet the real-time requirements, we used time domain features to characterize the motion signal segments. For each axis of data for each sensor, we extracted the mean, variance, standard deviation, mode, maximum, minimum, zero crossing, and range as features, following which each skill movement was described by a total of 72 features. The features and descriptions of the movements are shown in Table 3.

To eliminate the effects of different dimensions, we used the sigmoid function to map each value between 0 and 1. The Sigmoid expression is:

$$S(x) = \frac{1}{1+e^{-x}}, \tag{5}$$

The image of its function is as shown in Figure 2.

E. ORGANIZATION OF DATA SETS

This paper applied some supervised machine learning classification algorithms to recognize human motion.

TABLE 4. Label number of each action.

Action name	Label number
Forehand Attack	1
Forehand Drive	2
Forehand Chop	3
Forehand Flick	4
Backhand Control	5
Backhand Drive	6
Backhand Chop	7
Backhand Flick	8

After detecting the signal segment of each action, we added the label corresponding to each action and organized it into a complete ping-pong action data set. We used extracted data features to train and test k-NN, support vector machines, Naive Bayes, logistic regression, decision trees, and random forests, while we employed the filtered data directly to train and test the convolutional neural network. Therefore, this paper organized two action data sets: feature-based and data-based. The label number of each action is shown in Table 4.

III. RECOGNITION MODEL

A. FEATURE-BASED CLASSIFICATION MODEL

1) K-NEAREST NEIGHBOR

The idea of k-nearest neighbor (k-NN) [16] is that "like attracts like'', that is, we can determine the class of a test sample according to the classes of adjacent samples. The so-called proximity sample is the nearest K sample. This can be determined by calculating its distance from all known samples. k-NN's classification of new instances is a vote on the categories of k training instances that are closest to it.

There are many ways to calculate the distance. This paper used the Euclidean distance, which is the L2 norm between the feature vectors. The Euclidean distance between Vector $\mathbf{A} = (x_1^1, x_2^1, \ldots, x_n^1)$ and Vector $\mathbf{B} = (x_1^2, x_2^2, \ldots, x_n^2)$ is:

$$d_{AB} = \sqrt{\sum\nolimits_{i=1}^{n} (x_i^1 - x_i^2)^2}, \tag{6}$$

Thus, for the training set

$$\mathrm{T} = \{(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \ldots, (\boldsymbol{x_N}, y_N)\}$$

where x_i is the feature vector, and $y_i \in \{c_1, c_2, \ldots, c_K$ is the class of the instance, $i = 1, 2, \ldots, N$.

According to the method of measuring distance, we found k points closest to the test sample X in training set T, and we recorded the neighborhood of x covering these k points as $N_k(x)$.

Then, according to the classification decision rules (such as the majority vote), the classifier is expressed as

$$\mathrm{y} = \underset{c_j}{\mathrm{argmin}} \sum\nolimits_{x_i \in N_k(x)} I\left(y_i = c_j\right), \quad \mathrm{i} = 1, 2, \ldots, \mathrm{N}; \quad \mathrm{j} = 1, 2, \ldots, \mathrm{K}, \tag{7}$$

where, I is the indicator function. That is, I is 1 when $y_i = c_j$, otherwise I is 0.

In this paper, the K value was 3, which is to judge the class of samples according to the three nearest neighbors.

2) SUPPORT VECTOR MACHINE

Support vector machine is the linear classifier with the largest interval in feature space [17]. For non-linear problems, the SVM introduces the kernel techniques. It maps the input space to a feature space through a non-linear transformation, so that the hypersurface model in the input space is transformed into a hyperplane model in the feature space.

Let $\Phi(x)$ denote the feature vector from x in input space. Thus, the model of the hyperplane in the feature space can be expressed as

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \Phi(\boldsymbol{x}) + b, \tag{8}$$

In the process of solving parameters, we need to introduce an appropriate kernel function. Common kernel functions include the linear kernel, polynomial kernel, gaussian kernel and Sigmoid kernel. Gaussian kernels are used in this paper and their expression is as follows

$$\mathrm{K}(\mathrm{x}, \mathrm{z}) = \exp(-\frac{\|\mathrm{x\text{-}z}\|^2}{2\sigma^2}), \tag{9}$$

In this case, the classification decision function becomes

$$\mathrm{f}(\mathrm{x}) = \mathrm{sign}(\sum\nolimits_{i=1}^{N_s} a_i^* y_i \exp\left(-\frac{\|\mathrm{x\text{-}z}\|^2}{2\sigma^2}\right) + b^*), \tag{10}$$

3) NAIVE BAYES

The main principle of the Naive Bayes classification algorithm is that it makes a conditional independence assumption based on Bayes' theorem. For input $\mathrm{X} = (x_1, x_2, \ldots, x_n)$,, it calculates the posterior probability $\mathrm{P}(\mathrm{y} = c_k | \mathrm{X} = (x_1, x_2, \ldots, x_n))$ of the data item for each class. And then it takes the class with the greatest posteriori probability as the output of X.

According to Bayes' theorem, the posterior probability is calculated as

$$\mathrm{P}(\mathrm{Y} = c_k \mid x_1, x_2, \ldots, x_n) = \frac{P(Y = c_k) \cdot P(x_1, x_2, \ldots, x_n | Y = c_k)}{P(x_1, x_2, \ldots, x_n)}, \tag{11}$$

The denominator can be expanded from the full probability formula to

$$\sum\nolimits_k P(Y = c_k) \cdot P(x_1, x_2, \ldots, x_n | Y = c_k), \tag{12}$$

Molecules are complex multiplications of conditional probabilities, so the Naive Bayes algorithm makes an independent hypothesis. That is, it considers that the dimensional characteristics of the sample are independent. So, the molecule can be turned into

$$\mathrm{P}(\mathrm{Y} = c_k) \prod\nolimits_{i=1}^{n} P(x_i | Y = c_k), \tag{13}$$

Thus, the basic classification formula of the Naive Bayes is

$$\mathrm{P}(\mathrm{Y} = c_k \mid x_1, x_2, \ldots, x_n) = \frac{P(Y = c_k) \cdot \prod_{i=1}^{n} P(x_i | Y = c_k)}{\sum_k P(Y = c_k) \cdot P(x_1, x_2, \ldots, x_n | Y = c_k)}, \tag{14}$$

Note that the denominator is the same for all classes. We want to find the class with the highest probability, and so only considered the numerator. Thus, the classifier is represented as

$$y = \underset{c_k}{\operatorname{argmin}} P(Y = c_k) \cdot \prod_{i=1}^{n} P(x_i | Y = c_k), \tag{15}$$

where, k = 1,2,…,K.

In this paper, we assumed that the feature observations for each category are Gaussian-distributed, and so we used the Gaussian Naïve Bayes classification algorithm.

4) LOGISTIC REGRESSION

Logistic regression is a classic classification method in statistical learning [17]. It compares the values of the conditional probabilities of different classes and then assigns the instance x to the class with a higher probability value.

For binary classification, the logistic regression model is the conditional probability distribution as follows

$$P(Y = 1 \,|\, x) = \frac{exp(\boldsymbol{w} \cdot \boldsymbol{x} + b)}{1 + exp(\boldsymbol{w} \cdot \boldsymbol{x} + b)}, \tag{16}$$

$$P(Y = 0 \,|\, x) = \frac{1}{1 + exp(\boldsymbol{w} \cdot \boldsymbol{x} + b)}, \tag{17}$$

where, $x \in \boldsymbol{R}^n$ is input., Y ∈ {0, 1} is the output. w is the weight vector, and b is bias.

For multi-classification problems, the set of values of Y is {1,2,…,K},and the logistic regression model is

$$\mathrm{P}(\mathrm{Y} = \mathrm{k} \,|\, \mathbf{x}) = \frac{\exp(\boldsymbol{w_k} \cdot \boldsymbol{x} + b_k)}{1 + \sum_{k=1}^{K-1} \exp(\boldsymbol{w_k} \cdot \boldsymbol{x} + b_k)}, \tag{18}$$

$$\mathrm{P}(\mathrm{Y} = \mathrm{K} \,|\, \mathrm{x}) = \frac{1}{1 + \sum_{k=1}^{K-1} \exp(\boldsymbol{w_k} \cdot \boldsymbol{x} + b_k)}, \tag{19}$$

where, w_k and b_k are the weight vector and bias of the kth class.

In this paper, L2 was used as a regularization term; the error range for iteration termination was set to 1e-4, and the maximum number of iterations was set to 100.

5) DECISION TREE

The structure of the decision tree model is tree-shaped. In the classification problem, it represents the process of classifying instances based on features. Today, there are many decision tree algorithms. ID3 (Iterative Dichotomiser 3), C4.5 and CART (Classification and Regression Tree) are well-known algorithms in this regard. A decision tree algorithm usually consists of three steps: feature selection, spanning tree, and pruning tree [21].

Formally, the decision tree is a tree that is gradually built according to the features. In the different order of the selected features, the shape of the tree we get will also be different. What we want is a simple but effective structure, so it needs to choose features based on some appropriate methods. There are three methods for feature selection: information gain, information gain rate, and Gini index.

Information gain: the information gain of feature A on training data set D is defined as the difference between the information entropy of D and the conditional entropy of A.

$$\mathrm{g}(\mathrm{D}, \mathrm{A}) = \mathrm{H}(\mathrm{D}) - \mathrm{H}(\mathrm{D}|\mathrm{A}), \tag{20}$$

where, $\mathrm{H}(\mathrm{D}) = -\sum_{i=1}^{k} \frac{|D^i|}{|D|} log \frac{|D^i|}{|D|}$, k is the number of classes. $\mathrm{H}(\mathrm{D} \,|\, \mathrm{A}) = \sum_{i=1}^{n} \frac{|D^i|}{|D|} H(D^i)$, and n is the number of values of feature A.

Information gain rate: the information gain rate of feature A for training data set D is defined as the ratio of its information gain to the entropy of the value of feature A corresponding to data set D.

$$g_R(D, A) = \frac{g(D, A)}{H_A(D)}, \tag{21}$$

where, $H_A(D) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} log \frac{|D_i|}{|D|}$, n is the number of values of feature A.

c. Gini index: for k classes, if the probability that the sample points belong to the i-th class is p_i, then the Gini index is

$$Gini(p) = \sum_{i=1}^{k} p_i(1-p_i) = 1 - \sum_{i=1}^{k} p_i^2, \tag{22}$$

For a sample set D, its Gini index is

$$\mathrm{Gini}(\mathrm{D}) = 1 - \sum_{i=1}^{k} \left(\frac{|D_i|}{|D|}\right)^2, \tag{23}$$

where, D_i is a collection of samples belonging to the i-th class in D, k is the number of classes.

The pruning operations of the decision tree include pre-pruning and post-pruning:

Pre-pruning: estimate each node before partitioning in decision tree generation. If the partitioning of the current node does not improve the generalization performance, the partitioning is stopped and the current node is marked as a leaf node.

Post pruning: first, we generate a complete decision tree from the training set, and then observe the non-leaf nodes from the bottom up. If the node's subtree is replaced with a leaf node to improve generalization performance, then the subtree is replaced with a leaf node.

In this paper, the Gini coefficient was selected as the division criterion of the tree; the maximum depth of the tree was not limited, and the minimum number of samples required to distinguish the internal nodes was set to 2.

6) RANDOM FOREST

Random forest is a decision tree-based machine learning algorithm proposed by Breiman in 1995 [18]. As an integrated learning method, its idea is to brainstorm. It obtains multiple training sets from the original sample through the Bootstrap resampling method, and then builds decision trees to form a random forest. The sample to be tested votes on multiple results generated in the random forest, and the result with the highest number of votes is taken as its class.

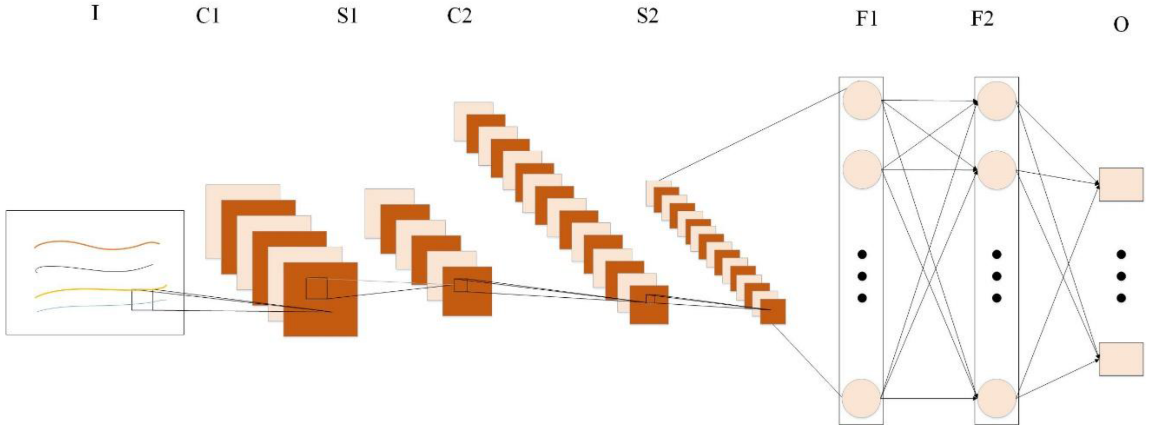


FIGURE 3. CNN network structure.

The feature selection of random forests in the decision tree training process is random. The traditional decision tree selects an optimal attribute in the current attribute set. In RF, a subset of k attributes is randomly selected from the attribute set of the node, and then an optimal attribute is chosen from the subset for division. For a random forest model containing T decision trees, its classifier can be expressed as:

$$\mathrm{y} = \underset{c_j}{\operatorname{argmin}} \sum\nolimits_{i=1}^{T} I(f_i\,(\mathbf{x}) = c_j) \quad \mathrm{j} = 1, 2, \ldots, \mathrm{K}, \quad (24)$$

where, I is the indicator function, $f_i\,(\boldsymbol{x})$ is the i-th decision tree model.

In this paper, the number of trees was set to 10; the Gini coefficient was used as the criterion for division.

B. CONVOLUTIONAL NEURAL NETWORK

Neural networks are the main form of deep learning [19], and the greater the number of network layers, the better the performance of data fitting. In the field of pattern recognition, Convolutional Neural Network (CNN) [20] is an efficient deep learning model. It can automatically learn data features through multi-layer non-linear transformations, and has strong expressive ability and learning ability. It has achieved good results in many aspects [21]–[22]. CNN has the characteristics of local connection, weight sharing and pooling operation [23], which can effectively reduce the network complexity and make it easy to train and optimize. Many researchers have already proposed improved algorithms for CNN [24]–[29].

The structure of the CNN used in this paper is shown in Figure 3, including one input layer (Input), two convolutional layers (C1,C2), two downsampling layers (S1,S2), two fully connected layers (F1,F2) and one output layer (Output).

where $C_i(i = 1, 2)$ represents convolutional layers; $S_i(i = 1, 2)$ represents downsampling layers;$F_i(i = 1, 2)$ represents fully connected layers.

a. Input layer is the original data of the action signal segment $X = (\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n})$, where $x_i = \left(x_{i_{11}}, x_{i_{12}}, \ldots, x_{i_{1w}}, z_{i_{31}}, z_{i_{32}} \ldots, z_{i_{3w}}\right)$ or $x_i = (x_{i_{11}}, y_{i_{11}}, \ldots, z_{i_{31}} x_{i_{1w}}, y_{i_{1w}}, \ldots, z_{i_{3w}})$, n is the number of input samples; $i = 1, 2, \ldots, n$.

b. The formula for convolutional layer is

$$x_j^l = f(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l), \quad (25)$$

where x_j^l represents the j_{th} feature map of the l_{th} layer; M_j is the set of input feature maps; k_{ij}^l is the j_{th} convolution kernel of layer l; b_j^l is the bias; $f\,(\cdot)$ is the activation function. In this paper, the ReLU (Rectified linear unit) [30] function was used as the activation function.

c. Downsampling layer follows the convolutional layer and corresponds to the feature map in the previous layer, with spatially invariant features [31]. Its general formula is

$$x_j^l = f(w_j^l down(x_j^{l-1}) + b_j^l), \quad (26)$$

where, w_j^l is the weight; b_j^l is the bias; $down(\cdot)$ is the downsampling function.

In this paper, we used the maximum pooling method

$$S_{ij} = \max\nolimits_{i=1,j=1}^{c}(H_{ij}), \quad (27)$$

That is, the largest element is extracted from the pooled region of size $c \times c$ in the input feature map H.

d. After two convolutional-pooling layers, two fully connected layers are connected, each neuron of which is connected to all neurons in the previous layer. The fully connected layer can integrate the local information with class discrimination among the convolutional layer or pooling layer [28]. The activation function is still the ReLU function.

To effectively avoid overfitting of the network, we used dropout [33] technology at the first fully connected layer. This approach can randomly inactivate some neurons during the training process, so as to improve the generalization ability of the network.

TABLE 5. Test results of k-NN.

	FA	FD	FC	FF	BCL	BDR	BC	BF
FA	78	1	0	0	0	0	0	0
FD	1	79	1	0	0	1	0	1
FC	0	0	78	0	0	0	0	0
FF	0	0	2	85	0	0	0	0
BCL	0	1	0	1	95	4	0	6
BDR	0	1	0	0	5	75	0	5
BC	0	0	0	0	0	0	81	0
BF	0	0	0	0	2	2	0	78

TABLE 6. Test results of SVM.

	FA	FD	FC	FF	BCL	BDR	BC	BF
FA	75	0	4	0	0	0	0	0
FD	12	60	9	0	0	1	0	1
FC	0	0	78	0	0	0	0	0
FF	5	0	0	81	0	0	0	1
BCL	0	0	0	6	78	7	0	16
BDR	3	2	0	0	9	62	0	10
BC	0	0	0	0	0	0	81	0
BF	1	0	0	3	4	11	0	63

TABLE 7. Test results of Naive Bayes.

	FA	FD	FC	FF	BCL	BDR	BC	BF
FA	61	4	0	14	0	0	0	0
FD	0	63	5	0	1	12	0	2
FC	1	1	76	0	0	0	0	0
FF	3	0	0	84	0	0	0	0
BCL	2	0	0	0	97	5	0	3
BDR	1	4	0	0	3	78	0	0
BC	2	1	0	0	0	0	78	0
BF	0	1	0	1	29	8	1	42

TABLE 8. Test results of logistic regression.

	FA	FD	FC	FF	BCL	BDR	BC	BF
FA	76	2	0	0	0	0	1	0
FD	13	61	6	0	0	3	0	0
FC	0	0	78	0	0	0	0	0
FF	0	0	0	87	0	0	0	0
BCL	1	0	0	1	88	5	0	12
BDR	2	2	0	0	8	68	0	6
BC	0	0	0	0	0	0	81	0
BF	0	0	0	2	7	6	0	67

TABLE 9. Test results of decision tree.

	FA	FD	FC	FF	BCL	BDR	BC	BF
FA	75	2	0	0	1	0	0	1
FD	2	77	0	0	1	1	0	2
FC	0	0	78	0	0	0	0	0
FF	4	0	0	83	0	0	0	0
BCL	1	0	0	0	100	0	0	6
BDR	2	2	0	0	4	76	0	2
BC	1	1	0	0	1	0	78	0
BF	1	1	2	0	2	1	0	75

TABLE 10. Test results of random forest.

	FA	FD	FC	FF	BCL	BDR	BC	BF
FA	78	0	1	0	0	0	0	0
FD	1	82	0	0	0	0	0	0
FC	1	0	77	0	0	0	0	0
FF	2	0	0	85	0	0	0	0
BCL	0	0	0	0	105	0	0	2
BDR	0	0	0	0	0	86	0	0
BC	1	0	0	0	0	0	80	0
BF	0	0	0	0	7	0	0	75

e. Finally, at output layer, the sample is classified by a softmax function

$$S_j = \frac{e^{a_j}}{\sum_{k=1}^{T} e^{a_k}}, \quad (j \in \{1, 2, \ldots, T\}), \tag{28}$$

In this paper, T = 8 was the number of label categories.

IV. EXPERIMENTAL RESULT

For the training and testing of k-NN, support vector machines, Naive Bayes, logistic regression, decision trees, and random forests, we used the features described in Section 2.4. However, for CNN, we employed the data directly after smoothing, without using any features. We randomly selected 30% of the total sample to test, including 683 samples; the remaining 70% of the data was used to train the model, including 1,592 samples. Our experiment was based on Python. The software we used was PyCharm, and we employed scikit-learn and TensorFlow to build the model.

The confusion matrix of k-NN, support vector machine, Naive Bayes, logistic regression, decision tree, and random forest is shown in Table 5 to Table 10.

In the experiments on CNNs, we set the learning rate to 3e-4, the batch size to 50, and we terminated the training after iterating 200 times. The recognition accuracy changes with the iteration times, as shown in Figure 5.

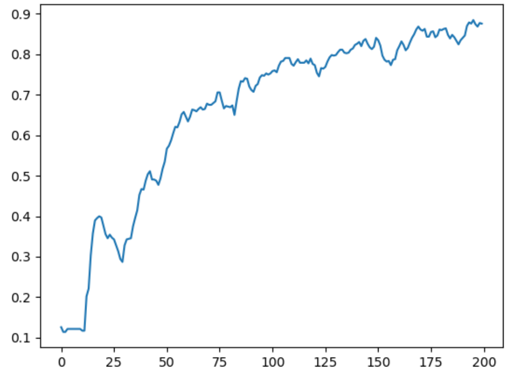


FIGURE 4. Accuracy varies with the number of iterations in CNN training.

The elements of the diagonal in the confusion matrix are the number of correctly recognized samples, and we can obtain the recognition accuracy by calculating the ratio of the sum of the diagonal elements and comparing it to the total number of test samples. In addition, we used precision, recall, and F1-score as indicators of performance evaluation, as shown in Table 11.

From the experimental results, we can see that the random forest has the highest recognition accuracy. It shows better

TABLE 11. Recognition rate of each model.

Model name	Accuracy(%)	precision(%)	recall(%)	F1-score(%)
k-NN	95.02	95.18	95.34	95.22
SVM	84.63	85.26	85.27	84.77
Naive Bayes	84.77	86.17	84.50	84.38
Logistic Regression	88.73	88.99	89.09	88.79
Decision tree	94.00	94.23	94.09	94.09
Random forest	97.80	97.98	97.79	97.86
CNN	87.55	87.57	87.73	87.58

performance than other models in this problem. This result also shows that this integrated learning method has stronger generalization ability in table tennis. Decision tree and k-NN have achieved 95.02% and 94.00% accuracy, both of which have shown good performance. The accuracy of the CNN we designed is similar to that of k-NN and decision trees, and does not exceed that of random forest. However, our CNN uses the action data directly without any features. It also shows a strong learning ability in ping-pong. In addition, these models have different recognition rates for the overall sample, while they also have different effects on specific actions. Comparing Tables 5 and 10 we can easily find that although the overall accuracy of k-NN is lower than that of random forest, the recognition accuracy of FC, BC and BF is higher than that of random forest.

V. CONCLUSION

In the IoT environment, integrating into smart watches, this paper proposes a solution to recognize ping-pong skill movements to assist with the training of amateur athletes. We use the inertial sensing data of smart watches and experiment by building machine learning models. The results show that a high recognition rate is achieved, which can effectively help the amateur ping pong player's action stereotypes. At the same time, there are still some shortcomings in our work. First of all, we only employ a single smart watch, which can effectively capture the upper limbs of the players, but the movements of the whole body in ping-pong are still very important, especially the training of the footwork. Moreover, the amount of data we use for model training is not large enough, and generalization performance may not be good enough.

Currently, we use multiple sensors to study the capture and recognition of whole body movements. In future work, for the use of data, we will increase the number of participants and employ the rule database to constrain the action. In terms of multi-modal fusion, we will combine the smart watch with multi-inertial sensors to explore a more effective approach to sport health computing.

REFERENCES

[1] J. C. Lee, "Hacking the nintendo wii remote," *IEEE Pervasive Comput.*, vol. 7, no. 3, pp. 39–45, Jul. 2008.

[2] Z. Q. Wang, S. H. Xia, X. J. Qiu, Y. Wei, L. Liu, and H. Huang, "Digital 3D trampoline simulating system: VHTrampoline," *Chin. J. Comput.*, vol. 30, no. 3, pp. 498–504, Mar. 2007, doi: 10.3321/j.issn:0254-4164.2007.03.018.

[3] P. Zappi, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Troster, "Activity recognition from on-body sensors by classifier fusion: Sensor scalability and robustness," in *Proc. 3rd Int. Conf. Intell. Sensors, Sensor Netw. Inf.*, Melbourne, QLD, Australia, Dec. 2007, pp. 281–286, doi: 10.1109/ISSNIP.2007.4496857.

[4] Y.-J. Hong, I.-J. Kim, S. C. Ahn, and H.-G. Kim, "Mobile health monitoring system based on activity recognition using accelerometer," *Simul. Model. Pract. Theory*, vol. 18, no. 4, pp. 446–455, Apr. 2010.

[5] D. O. Olguin and A. Pentland, "Human activity recognition: Accuracy across common locations for wearable sensors," in *Proc. Int. Symp. Wearable Comput.*, Oct. 2006, pp. 11–14.

[6] A. E. Halabi and H. Artail, "Integrating pressure and accelerometer sensing for improved activity recognition on smartphones," in *Proc. 3rd Int. Conf. Commun. Inf. Technol. (ICCIT)*, Jun. 2013, pp. 121–125, doi: 10.1109/ICCITechnology.2013.6579534.

[7] J. Xie, D. Wen, and L. Liang, "Evaluating the validity of current mainstream wearable devices in fitness tracking under various physical activities: Comparative study," *JMIR Health Uhealth*, vol. 6, no. 4, p. e94, 2018.

[8] A. Ancillao, S. Tedesco, J. Barton, and B. O'Flynn, "Indirect measurement of ground reaction forces and moments by means of wearable inertial sensors: A systematic review," *Sensors*, vol. 18, no. 8, p. 2564, Aug. 2018.

[9] E. Shahabpoor and A. Pavic, "Measurement of walking ground reactions in real-life environments: A systematic review of techniques and technologies," *Sensors*, vol. 17, p. 2085, Sep. 2017.

[10] A. Kamišalic, I. Fister, M. Turkanovic, and S. Karakatic, "Sensors and functionalities of non-invasive wrist-wearable devices: A review," *Sensors*, vol. 18, no. 6, p. 1714, May 2018.

[11] D. Yang, J. Huang, X. Tu, G. Ding, T. Shen, and X. Xiao, "A wearable activity recognition device using air-pressure and IMU sensors," *IEEE Access*, vol. 7, pp. 6611–6621, 2019, doi: 10.1109/ACCESS.2018.2890004.

[12] Y. Chen and C. Shen, "Performance analysis of smartphone-sensor behavior for human activity recognition," *IEEE Access*, vol. 5, pp. 3095–3110, 2017, doi: 10.1109/ACCESS.2017.2676168.

[13] H. Amroun and M. Ammi, "Who used my smart object? a flexible approach for the recognition of users," *IEEE Access*, vol. 6, pp. 7112–7122, 2018, doi: 10.1109/ACCESS.2017.2776098.

[14] D. Tian, X. Xu, Y. Tao, and X. Wang, "An improved activity recognition method based on smart watch data," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE)*, Guangzhou, China, Jul. 2017, pp. 756–759, doi: 10.1109/CSE-EUC.2017.148.

[15] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," *Pattern Recognit. Lett.*, vol. 29, no. 16, pp. 2213–2220, Dec. 2008.

[16] T. M. Cover, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[17] L. Hang, *Statistical Learning Method*. Beijing, China: Tsinghua Univ. Press, 2012.

[18] L. Breiman, "Random forest," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[19] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

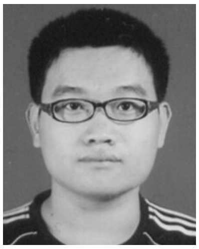
[20] Y. LeCun, B. Boser, and J. S. Denker, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[21] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.

[22] C. Nebauer, "Evaluation of convolutional neural networks for visual recognition," *IEEE Trans. Neural Netw.*, vol. 9, no. 4, pp. 685–696, Jul. 1998.
[23] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[24] M. Lin, Q. Chen, S. Yan, "Network in network," Dec. 2013, *arXiv:1312.4400*. [Online]. Available: https://arxiv.org/abs/1312.4400
[25] C. Xu, C. Lu, X. Liang, J. Gao, W. Zheng, T. Wang, and S. Yan, "Multi-loss Regularized Deep Neural Network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 12, pp. 2273–2283, Dec. 2016, doi: 10.1109/TCSVT.2015.2477937.
[26] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," Jun. 2015, *arXiv:1506.02025*. [Online]. Available: https://arxiv.org/abs/1506.02025
[27] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2528–2535, doi: 10.1109/CVPR.2010.5539957.
[28] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*. Zurich, Switzerland, 2014, pp. 818–833.
[29] J. B. Zhao, M. Mathieu, R. Goroshin, and Y. LeCun, "Stacked what-where auto-encoders," Jun. 2015, *arXiv:1506.02351*. [Online]. Available: https://arxiv.org/abs/1506.02351
[30] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
[31] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, L. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," Dec. 2015, *arXiv:1512.07108*. [Online]. Available: https://arxiv.org/abs/1512.07108
[32] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 8614–8618, doi: 10.1109/ICASSP.2013.6639347.
[33] S. Nitish, H. Geoffrey, and K. Alex, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jan. 2014.

HENG ZHANG received the bachelor's degree in computer science from Chongqing University, Chongqing, China, in 1999, the master's degree, in 2005, and the Ph.D. degree in computer science from the University of Electronic Science and Technology, Chengdu, Sichuan, China, in 2012.

From 1999 to 2002, he worked with Foxconn, Dike, and other companies in technology research and development and project management in Shenzhen, China. He has been teaching at Southwest University, Chongqing, since 2005. He has been serving as a master's tutor with the School of Computer and Information Science, since 2012. He worked in Government, Nanjing, Jiangsu, China, from 2013 to 2014, mainly responsible for the transformation of scientific and technological achievements. He was a Visiting Scholar with the Digital Media Research Center and Artificial Intelligence Research Center, Universität Bremen, Bremen, Germany, from 2016 to 2017. His research interests include artificial intelligence and universal health, novel human motion capture and VR, pattern recognition and machine learning in medical health and other aspects, and the Internet of Things application systems. He is on the special committee of the Human–Computer Interaction Committee and Pervasive Computing Committee, Chinese Computer Society, and the guidance and evaluation expert of the transformation of science and technology industry.



ZENGJUN FU was born in Cangzhou, Hebei, China, in 1991. He received the bachelor's degree from the Tianjin University of Technology, China. He is currently with the School of Computer and Information Science, Southwest University. He is familiar with the basic knowledge of computer science and has practical experience in software development and computer vision. He is skilled at using a variety of software and tools. He has participated in many competitions and achieved good results. His research interests include machine learning, the IoT and wearable computing, and human–computer interaction.

KUANG-I SHU was born in Taiwan, in 1952. He received the master's and Ph.D. degrees in electrical engineering from Polytechnic University, New York, NY, USA.

He has been engaged in communication equipment design in USA for more than 30 years, has made great achievements in the fields of communication and universal health. Successively, he has served as a Researcher for Bell laboratories, a Senior Engineer of other companies, the Group Leader, a Research and Development Manager, the Director, and the Vice President. He has created three start-ups, and obtained two U.S. patents and another one is under review. In March 1999, he became the Founding Member of Santera (later incorporated into Genband), where he has served as the Hardware Director of the company. He was engaged in the industry pioneer design of large VoATM and VoIP media gateway of the core networks. He returned to China at the end of 2012, as the Hardware Director of Aisino Corporation, responsible for new product planning and hardware checking. In November 2015, he was hired as a Professor with the School of Computer and Information Science, Southwest University, Chongqing, China, to prepare for the IoT Intelligent Innovation and Industrialization Center, which is equipped with vehicle-mounted systems and equipment laboratory and intelligent handheld terminal laboratory, conducting fruitful research in the areas of car networking, artificial intelligence, and the Internet of Things application systems.

Dr. Shu received the Research and Development Award from Dr. Penzias, in 1991, the Vice President of Research at Bell Laboratories (Nobel Laureate in physics), and the First Genband Award for Best Leadership, in 2008.