Complex & Intelligent Systems (2023) 9:4657–4671
https://doi.org/10.1007/s40747-023-00979-8

ORIGINAL ARTICLE

# RaiseAuth: a novel bio-behavioral authentication method based on ultra-low-complexity movement

Shuo Zhao[1] · Zhongwen Guo[1] · Xu Cheng[2] · Sining Jiang[1] · Hao Wang[3]

Received: 29 August 2022 / Accepted: 16 January 2023 / Published online: 7 February 2023

## Abstract

Authentication plays an important role in maintaining social security. Modern authentication methods often relies on mass data datasets to implement authentication by data-driven. However, an essential question still remains unclear at data level. To what extent can the authentication movement be simplified? We theoretically explain the rationality of authentication through arm movements by mathematical modeling and design the simplest scheme of the authentication movement. At the same time, we collect a small-sample multi-category dataset that compresses the authentication movement as much as possible according to the model function. On this basis, we propose a method which consists of five different cells. Each cell is matched with a custom data preprocessing module according to the structure. Four cells are composed of neural network modules based on residual blocks, and the last cell is composed of traditional machine learning algorithms. The experimental results show that arm movements can also maintain high-accuracy authentication on small-sample multi-class datasets with very simple authentication movement.

**Keywords** Sensor · Biometric authentication · Behavioral authentication · Neural network · Machine learning

## Introduction

Human authentication has always been the focus of attention in security field. Human authentication and identity camouflage are like two armies that confront each other and upgrade their equipment, and constantly propose corresponding solutions according to each other's technological development. The backwardness of the authentication often leads to serious economic problems, the lack of credibility, and even more fatal consequences.

In recent years, the improvement of computer computing power and the development of related methods have promoted the diversity and accuracy of human authentication, and gradually met various personal privacy protection needs and organizational privacy protection needs.

The knowledge-based authentication method based on "people knows information" [1] has been widely used for several decades. Including common access card [2,3], username and password, etc. The 4- or 6-digit pin codes authentication method or the common access card authentication method makes it difficult for the perpetrator to imitate others in a short period of time without preparation. However, several researches [4] have shown that knowledge-based authentication methods such as pin codes are difficult to remember and vulnerable to attack by perpetrators [5,6]. Common access cards are easily stolen or lost [3], and usernames and passwords are easily disclosed and embezzled. Hence, about one-fifth [4] of people prefer to store all of the personal information in one device, leading to a significantly increased risk and harm of information leakage.

The biometric-based authentication method based on "people is something" has greatly alleviated the above prob-

✉ Zhongwen Guo
guozhw@ouc.edu.cn

Shuo Zhao
zhaoshuo@stu.ouc.edu.cn

Xu Cheng
xu.cheng@ieee.org

Sining Jiang
jsn@stu.ouc.edu.cn

Hao Wang
hawa@ntnu.no

[1] Department of Computer Science, Ocean University of China, Songling Road, Qingdao 266100, Shandong, China

[2] Smart Innovation Norway, 1783 Halden, Norway

[3] Department of Computer Science, Norwegian University of Science and Technology, Gjøvik, 2815 Innlandet, Norway

lems. The biometric is usually divided into bio-physiological and bio-behavioral due to the nature of the human feature. Bio-physiological authentication achieves the purpose of recognizing a person's identity by measuring the physical features of the human body, including hand region features [7,8], facial region features [9,10], ocular features [11,12], etc.. Bio-physiological authentication does not require people to remember anything, but requires cooperation to provide a sufficient amount of features for authentication. Therefore, bio-physiological authentication is usually used in situations where the person voluntarily provides personal information or is compelled to provide personal information under supervision, such as police stations and banks. The bio-behavioral authentication is well adapted to the authentication needs under unsupervised conditions. Bio-behavioral authentication identifies people by detecting and learning the biological behaviors such as the personal habitual movements. Including voice recognition [13,14], gait recognition [15,16], keystroke dynamics [17] and signature [18,19]. Because each person's body characteristics (such as: height, weight, arm length, muscle development, throat development, etc.) and behavioral habits (such as: keystroke speed and strength, step size, etc.) are different, bio-behavioral authentication can identify people with acceptable accuracy. Compared with knowledge-based authentication, bio-behavioral authentication does not have the disadvantages of being easily lost or stolen. Compared to bio-physiological authentication, bio-behavioral authentication requires less hardware and can be operated by low-cost sensors. At the same time, bio-behavioral authentication has higher concealment [20,21] and is not easy to be discovered.

However, no matter in the collection of dataset or the application of the methods, there is a fact that cannot be ignored, that is, in most real life scenarios, the person being measured will not provide such ample authentication information. For the most common example, stealing is often a quick and precise movement completed within 1–5 s. The thief's face is not completely exposed to the camera, and there is no interaction with the phone screen or buttons. In addition, such as the movement of the intruder opening the door, etc., people in these scenarios often do not provide sufficient data and rich variety of features for authentication.

Therefore, this leads to two important questions and challenges:

1. Can a simple movement that is completed in a short time and only generates a small amount of data be used for authentication?
2. Can simple and commonly used sensors capture these very short-term movements?

In response to the above challenges, we compress the time and complexity of movements to an unprecedented degree. At the same time, we design a variety of method structures to process data, and make a detailed analysis.

In this work, we first model the authentication movement based on human joints and bones from a mathematical point of view and construct the model function. According to the relevant parameters of the function, the authentication movement is analyzed in detail and the sensor selection method is designed.

Then, because the existing public dataset could not support our work, we collected a 110-persons movement dataset for the designed authentication movement. The participants have no prior knowledge, that is, the participants who are collected data do not know the purpose of the data before the collection, and only inform them of data use after collection and obtain data use authorization. In this way, it is ensured that the collected data conform to the daily habits of the collected participants.

Finally, since there has never been a previous authentication method based on data of the same order of magnitude as our movement data, we designed different deep neural network structures based on a variety of common backbones and combined with a variety of machine learning classification methods for the results comparison and analysis.

We note that a shorter conference version [22] of this paper appeared in ACM Turing Celebration Conference (2020). Our previous work did not analyze the authentication movement mathematically, and did not clarify the connection between movement, mathematical model and sensor. Compared with previous work, we greatly compress the movement complexity, reduce the number of sensors, add four additional cells and add a cell evaluation module.

The remainder of the paper is organized as follows: In the next section, we describe related work in bio-behavioral authentication. In the subsequent section, we analyze the mathematical model and design the authentication movement followed by which we introduce our proposed method named RaiseAuth. In the penultimate section, we analyze the model performance and the resistance to attack. We conclude this work in the final section.

## Related work

In this section, we summarize the efforts of recent research community in bio-behavioral authentication and corresponding measurement studies.

A number of bio-behavioral authentication methods have been reported during the last decade [1,23]. Hong [24] collected the sensor data generated by the human hand when writing through a special watch containing an acceleration sensor, and used this as the basis for authentication. Each participant is required to write 20 words and approximately 120 strokes, which takes approximately three minutes to generate one training sample. Langyue [25] proposed a novel feature extractor which achieved 97% accuracy in authentication based on people's gait behavior, where each person needed to provide data generated by walking for more than 33 min for training. Timothy [26] authenticated people through the coherent movements of people hitting the keyboard continuously, and designed the method based on a training dataset of over 5000 keyboard strokes per person. The above methods can often achieve an accuracy of more than 95%, and have good application prospects in some specific situations. However, it is obvious that the amount of dataset used by these methods is huge, and it takes a long cost to collect data from one person to train the model and improve the accuracy, which makes these methods unsuitable for short-term simple movement data.

Reducing the complexity and data scale of the authentication movement is important because each order of magnitude decrease in authentication movement ushers in a new set of unforeseen challenges. Several researches [27–30] are also trying to use simple movements to achieve authentication. In [31], the author used the combined movement when the user answers the phone, that is, the user unlocked the mobile phone and took the mobile phone to the ear as an authentication movement. A 5% Equal Error Rate (EER) was achieved on a self-built dataset of 48 people based on a training set where each person performed ten movements within 6000 ms. Jakub [32] collected the phone-holding behavior data of participants within 40 min, and proposed a method based on Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN) machine learning models. The trained model is able to achieve 8% EER on 20 s level test samples. Although the above methods and systems have made certain contributions in compressing the amount of data, there is still room for improvement due to the following two reasons. First, the above methods are all data-driven, and do not analyze and explain the authentication movement itself, which makes the correlation between the authentication movement and authentication method unclear. Therefore, our method fills in this gap by analyzing in detail the mathematical correlation between movements, sensors, and authentication. Secondly, even if the above methods simplify the authentication movement, the amount of data required for single-sample training is still more than 2.5 times ours.

In contrast, RaiseAuth uses a simpler authentication movement, establishes a mathematical model to explain the rationality of the movement, and comprehensively analyzes the model structure and authentication results.

## Movement and sensors

In this section, we build a mathematical model of human joints and bones. By analyzing the relevant parameters of the model functions, the authentication movement is designed and the reasons for the selection are explained. Also the sensor selection method is designed according to the analysis results.

### Mathematical analysis

Most of the recognizable behaviors of the human body depend on the operation of the limbs. Compared with the trunk, the limbs have fewer bones but have higher flexibility. The muscles can ensure the normal operation of the limbs, and bones and joints provide a natural entry point for the establishment of mathematical models.

We chose the right arm among the limbs as the model building template (Fig. 1), where point $O$ is the shoulder joint, point $M$ is the elbow and point $N$ means the wrist. We take the point $O$ as the coordinate origin, and put the bone and joint models into the space coordinate system. At the same time, we assume that the initial state of the arm is $O-A_1-B_1$, the end state of the arm movement is $O-A_2-B_2$, and the arm state $O-M-N$ is intermediate state of the arm at time $t (t \in [0, T])$ during the movement. Note that the movement trajectory of the arm's change from $O-A_1-B_1$ state to $O-A_2-B_2$ state is not necessarily a straight line change, but more likely an irregular trajectory change.

Set point $A_1(a, 0, 0)$,point $B_1(a+b, 0, 0)$, at time $t$, elbow is located at point $M(x_1, y_1, z_1)$, the angle of rotation of the upper arm $O-M$ is:

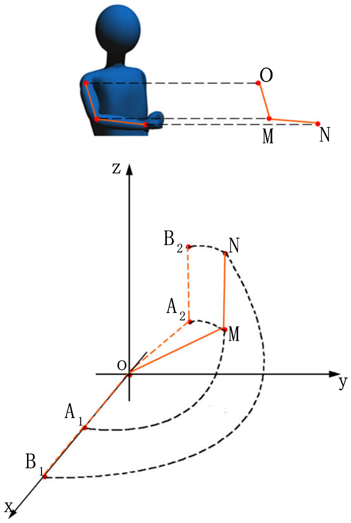


**Fig. 1** Human arm joints and bones mathematical model

$$\overrightarrow{\theta_1} = \overrightarrow{\theta_1}(t) = (\theta_{1x}(t), \theta_{1y}(t), \theta_{1z}(t)), \tag{1}$$

the angular velocity is:

$$\overrightarrow{\omega_1} = \overrightarrow{\omega_1}(t) = (\omega_{1x}(t), \omega_{1y}(t), \omega_{1z}(t)). \tag{2}$$

For the forearm $M - N$, set point $N(x_2, y_2, z_2)$, relative to the upper arm the angle of rotation is:

$$\overrightarrow{\theta_2} = \overrightarrow{\theta_2}(t) = (\theta_{2x}(t), \theta_{2y}(t), \theta_{2z}(t)), \tag{3}$$

the angular velocity is:

$$\overrightarrow{\omega_2} = \overrightarrow{\omega_2}(t) = (\omega_{2x}(t), \omega_{2y}(t), \omega_{2z}(t)). \tag{4}$$

Then

$$\overrightarrow{\theta_1} = \int_0^t \overrightarrow{\omega_1(t)}\mathrm{d}t = \left(\int_0^t \omega_{1x}(t)\mathrm{d}t, \int_0^t \omega_{1y}(t)\mathrm{d}t, \int_0^t \omega_{1z}(t)\mathrm{d}t\right), \tag{5}$$

$$\overrightarrow{\theta_2} = \int_0^t \overrightarrow{\omega_2(t)}\mathrm{d}t = \left(\int_0^t \omega_{2x}(t)\mathrm{d}t, \int_0^t \omega_{2y}(t)\mathrm{d}t, \int_0^t \omega_{2z}(t)\mathrm{d}t\right). \tag{6}$$

When $t = T$, set point $A_2(c_1, d_1, e_1)$, point$B_2(c_2, d_2, e_2)$, in the state change of the arm from $O - A_1 - B_1$ to $O - A_2 - B_2$, the trajectory traversed by the elbow $M$ is $L_1 = A_1A_2$, the trajectory traversed by the wrist $N$ is $L_2 = B_1B_2$. Since the upper arm $O - M$ and the forearm $M - N$ are two inflexible bones, the trajectory traversed by the points on the line $OM$ and $MN$ are actually the concentric motion with the same angular velocity as the end points $M$ and $N$, respectively. Therefore, the movement state function of the arm can be simplified to be represented by the motion trajectories of points $M$ and $N$, so we have the movement state function

$$Y = L_1 + L_2. \tag{7}$$

Then

$$\begin{cases} x_1 = a\cos\theta_{1x} \\ y_1 = a\cos\theta_{1y} \\ z_1 = a\cos\theta_{1z} \end{cases}, \tag{8}$$

$$\begin{cases} x_2 = x_1 + b\cos(\theta_{1x} + \theta_{2x}) \\ y_2 = y_1 + b\cos(\theta_{1y} + \theta_{2y}) \\ z_2 = z_1 + b\cos(\theta_{1z} + \theta_{2z}) \end{cases}, \tag{9}$$

calculate the differential:

$$\begin{cases} dx_1 = -a\sin\theta_{1x}\cdot\omega_{1x}(t)\mathrm{d}t \\ dy_1 = -a\sin\theta_{1y}\cdot\omega_{1y}(t)\mathrm{d}t \\ dz_1 = -a\sin\theta_{1z}\cdot\omega_{1z}(t)\mathrm{d}t \end{cases}, \tag{10}$$

$$\begin{cases} dx_2 = [-a\sin\theta_{1x}\cdot\omega_{1x} - b\sin(\theta_{1x} + \theta_{2x})(\omega_{1x} + \omega_{2x})]\mathrm{d}t \\ dy_2 = [-a\sin\theta_{1y}\cdot\omega_{1y} - b\sin(\theta_{1y} + \theta_{2y})(\omega_{1y} + \omega_{2y})]\mathrm{d}t \\ dz_2 = [-a\sin\theta_{1z}\cdot\omega_{1z} - b\sin(\theta_{1z} + \theta_{2z})(\omega_{1z} + \omega_{2z})]\mathrm{d}t \end{cases}, \tag{11}$$

then

$$\begin{aligned} L_1 &= \int_0^T \sqrt{(dx_1)^2 + (dy_1)^2 + (dz_1)^2} \\ &= a\int_0^T \sqrt{(\sin\theta_{1x}\cdot\omega_{1x})^2 + (\sin\theta_{1y}\cdot\omega_{1y})^2 + (\sin\theta_{1z}\cdot\omega_{1z})^2}\mathrm{d}t \end{aligned} \tag{12}$$

$$L_2 = \int_0^T \sqrt{(dx_2)^2 + (dy_2)^2 + (dz_2)^2} = \int_0^T \sqrt{f(t)}\mathrm{d}t, \tag{13}$$

where

$$\begin{aligned} f(t) = a^2[&(\omega_{1x}\sin\theta_{1x})^2 + (\omega_{1y}\sin\theta_{1y})^2 + (\omega_{1z}\sin\theta_{1z})^2] \\ &+ b^2[((\omega_{1x} + \omega_{2x})\sin(\theta_{1x} \\ &+ \theta_{2x}))^2 + ((\omega_{1y} + \omega_{2y})\sin(\theta_{1y} + \theta_{2y}))^2 \\ &+ ((\omega_{1z} + \omega_{2z})\sin(\theta_{1z} + \theta_{2z}))^2] \\ &+ 2ab[\omega_{1x}(\omega_{1x} + \omega_{2x})\sin\theta_{1x}\sin(\theta_{1x} + \theta_{2x}) \\ &+ \omega_{1y}(\omega_{1y} + \omega_{2y})\sin\theta_{1y}\sin(\theta_{1y} + \theta_{2y}) \\ &+ ((\omega_{1z} + \omega_{2z})\sin(\theta_{1z} + \theta_{2z}))^2] \\ &+ \omega_{1z}(\omega_{1z} + \omega_{2z})\sin\theta_{1z}\sin(\theta_{1z} + \theta_{2z})]. \end{aligned} \tag{14}$$

The derivation of the above formulas show that the movement state function is composed of two parts, $L_1$ and $L_2$, which means that the most comprehensive data acquisition scheme should be to wear sensors at the elbow $M$ and wrist $N$ for measurement, and use the data to authentication. However, through Formulas (12) and (13), it can be found that the parameters required to calculate $L_2$ cover all the parameters required by $L_1$, and the parameter richness of $L_2$ is much higher than that of $L_1$, that means the weight of $L_2$ in the movement state function should be much higher than $L_1$. So it can be concluded that only adding sensors at wrist $N$ to measure $L_2$ should also be able to achieve authentication with good performance.

## Authentication movement

In Formula (14), it can be found that the main parameters that affect the movement state function are the upper arm length $a$, the forearm length $b$, the angular velocity $\omega$ with the three components of the space coordinate system, and the rotation angle $\theta$, which means that the authentication movement only needs to meet these conditions to be able to authenticate. So the movement we designed is very simple (Fig. 2):

We asked participants to place their arms below their waist and then raise them 15–20 cm according to their natural strength and speed. This movement is typically completed within 2 s. Putting the hands below the waist is only to ensure that the participants have enough room to raise the hand, and the reason for the raise is that we need to ensure that the arm movement is from the participant's subjective decision, which makes the participants to mobilize more muscle power to perform the movement, which is the key to affecting angular velocity and rotation angle. At the same time, since the lengths of the upper arms and forearms of each participant are different, this will also provide more parameter variables for the movement state function. Therefore, with such a simple movement, all the requirements of the movement state function are satisfied. At the same time, we provide two scenarios, that is, to complete this movement in the scenario of sitting and standing, respectively.

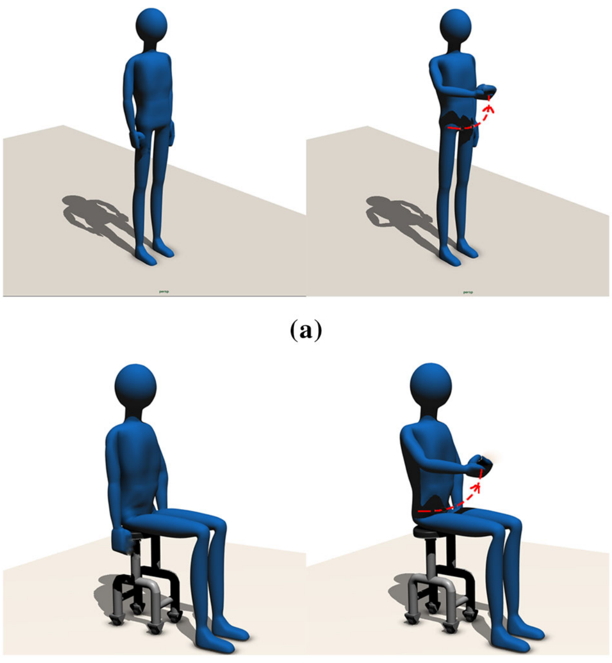


**Fig. 2** Authentication movement. **a** In standing posture. **b** In sitting posture

## Sensors selection

Accelerometer, magnetometer and gyroscope sensors are chosen as the sensor for data acquisition for the following two reasons. First, after mathematical analysis of arm model, the movement function we obtained reveals the parameter composition that affects authentication movement. The accelerometer sensor can provide good data support for calculating the trajectory traveled by the wrist, the gyroscope sensor can intuitively reflect the change of angular velocity, and the magnetometer sensors provides convenience for confirming the instantaneous direction of movement. Second, We aim to increase the applicability of the method by collecting data using simple and commonly used sensors. At present, most smartphones have built-in accelerometer, magnetometer and gyroscope sensors to meet the different needs of different mobile applications. Therefore, choosing these three sensors can make our method have a wide range of application scenarios, rather than just stay in theory.

### Accelerometer

The acceleration sensor can collect the acceleration of the sensor itself. Due to the gravity of the earth, when the device is stationary, the value of the accelerometer sensor will always generate 9.81 m/s$^2$ interference. To eliminate this interference, high-pass filtering is used to process the acceleration sensor data. The low-pass filter is also used to process the raw data of acceleration sensor. The low-pass filter can eliminate the noise generated in the process of data collection. The accelerometer data is used to reflect the user's habits of the arm strength and direction of the arm force. At the same time, the accelerometer data will be used for auxiliary calculation trajectory.

The data processed by high-pass filter and low-pass filter are added to the feature subset select together with the raw data to ensure that the important feature information will not be lost.

### Gyroscope

Due to the rigidity and precession of the gyroscope, the gyroscope data can provide an important basis for calculating the change of the angular velocity in three-dimensional direction.

### Magnetometer

Magnetometer sensor can measure the strength and direction of the magnetic field of the sensor. The data of magnetometer sensor can reflect the instantaneous direction change of sensor. In our method process, magnetometer data are used to reflect the direction during the participant moving his arm.

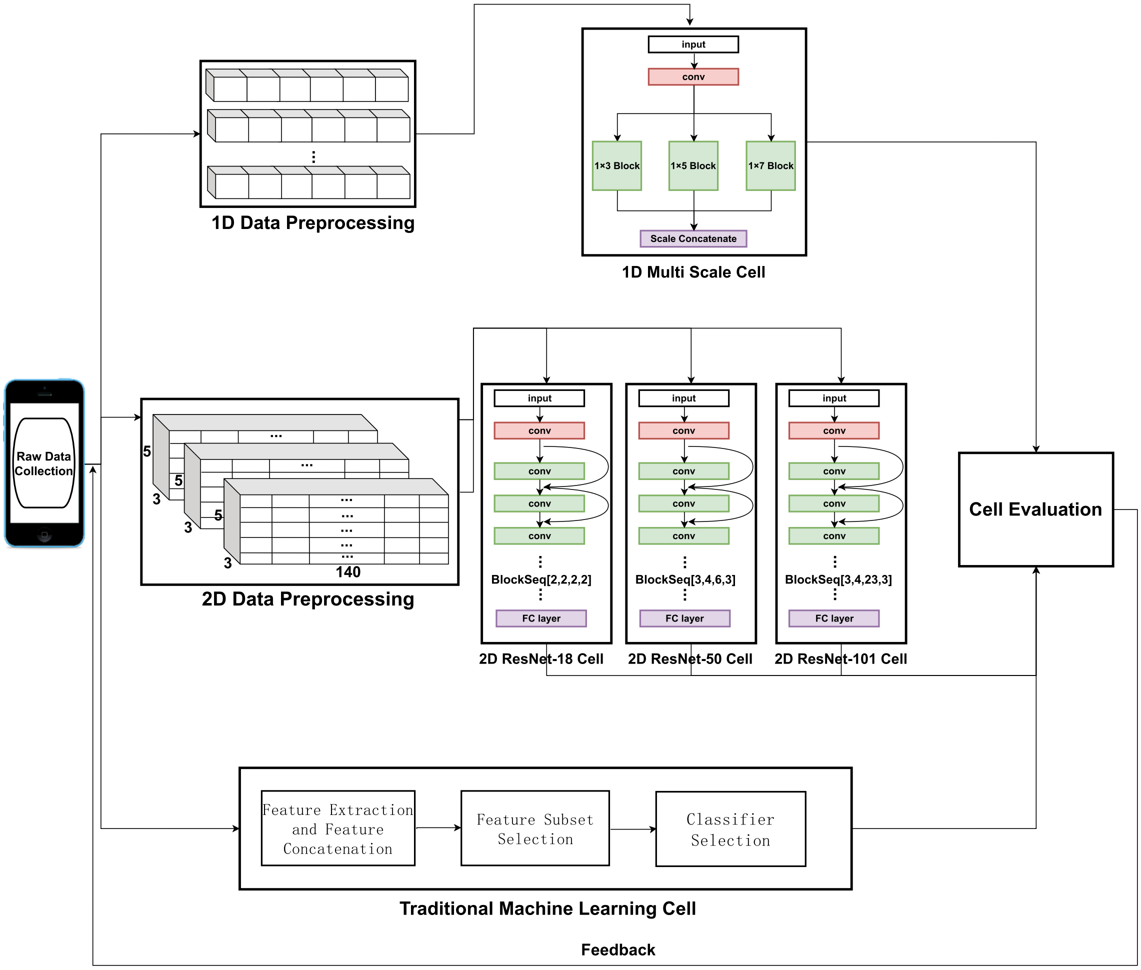


**Fig. 3** RaiseAuth method architecture

## Method

For better evaluation, we build a variety of architectures suitable for us based on the existing popular backbones, and put them into different cells to select the most suitable one and do further analysis. The architecture of the RaiseAuth method is shown in Fig. 3, which contains 5 cells, each cell is configured with the corresponding data preprocessing method. Cell evaluation will calculate the score each cell gets on the dataset during training, and provide feedback that will cause RaiseAuth to close paths to the cells with low scores.

The data of accelerometer, magnetometer and gyroscope were, respectively, expressed as $(AX, AY, AZ)$, $(MX, MY, MZ)$, $(GX, GY, GZ)$, and the experimental data obtained by the accelerometer were processed by low-pass filtering (LPF) and high-pass filtering (HPF) respectively, and the obtained data were, respectively, expressed as $(LX, LY, LZ)$ and $(HX, HY, HZ)$.

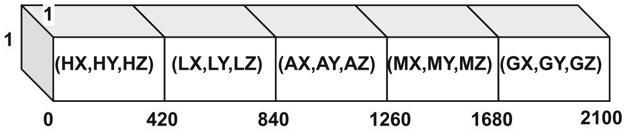


**Fig. 4** 1D multi-scale cell data preprocessing

## 1D multi-scale cell

What we first consider is to process and analyze the sensor data from a one-dimensional perspective. Due to the limited receptive field of the one-dimensional convolution kernel, we extract features from the input data from three different scales based on the bottleneck of ResNet [33] to achieve multiple classification task.

### 1D data preprocessing

As shown in Fig. 4, we stitch together the sensor data in the order of HPF, LPF, accelerometer, magnetometer and gyroscope. The length of data generated by each sensor on each axis is fixed at 140, the insufficient part is filled with zero, and the excess part will be discarded, forming a one-dimensional data input with a length of 2100.

### Cell architecture details

The cell structure and details are shown in Fig. 5. Where BN means batch normalization, "/2" means stride is 2. Taking "1×7conv, 32, /2" as an example, it means to perform a convolution calculation with a convolution kernel of 1×7 size, the number of output channels is 32, and the stride is 2. Taking "1×3 maxpool, /2" as an example, it means to perform a max pooling calculation with a convolution kernel of 1×3 size and the stride is 2. Taking "1×6 avgpool, /1" as an example, it means to perform a average pooling calculation with a convolution kernel of 1×6 size and the stride is 1.

When the preprocessed data are input into the cell, it will first go through two convolution layers to increase the number of channels. We choose to use a large convolution kernel to increase the receptive field, thereby increasing the breadth and depth of the feature map. We adopt batch normalization (BN) [34] right after each convolution and before activation, following [34]. We choose to add max pooling to mix features and adjust output resolution after the activation function. Then, considering the restricted receptive field of one-dimensional convolution, we perform residual convolution operations on the input from three scales of 1×3, 1×5, and 1×7, and finally concatenate the feature vector by the output of the average pool. And input the vector into the fully connected layer, and combine with softmax to achieve classification.

During training, we use a batch size of 64 and use SGD with a momentum of 0.9 to initialize the weight. The learning rate is initialized to 0.01 and exponentially descends with a descending rate of 0.95. When the error plateaus, the learning rate will be divided by 10. Dropout is not used, following the practice in [33]. The models are trained for up to 150 epochs.

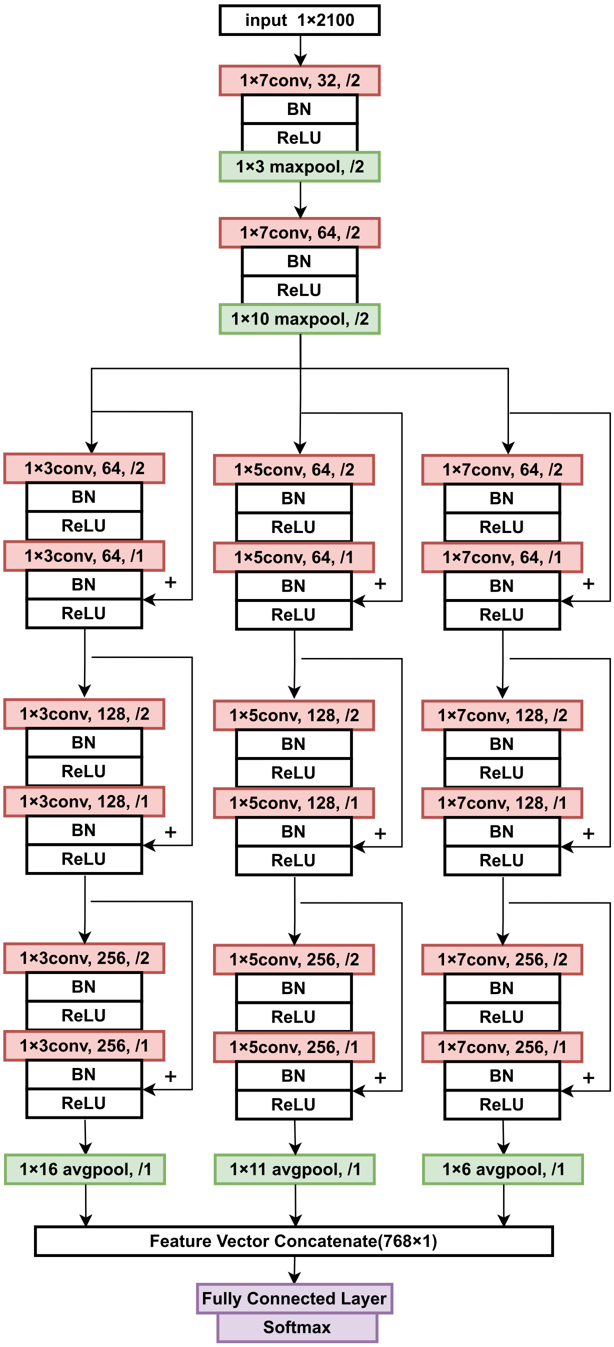


**Fig. 5** 1D multi-scale cell architecture

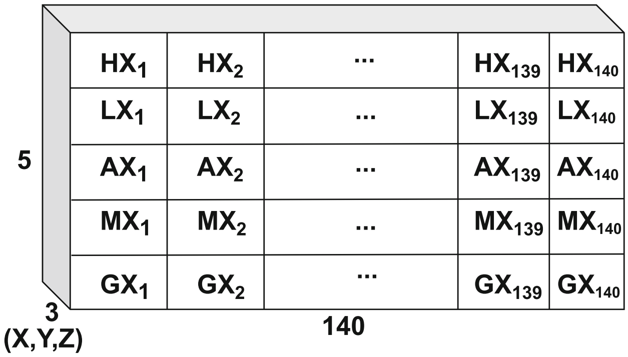
## 2D residual cell

In this cell, we preprocess the signal into a 2D structure and combine ResNet to achieve multi-classification tasks. We choose models of different depths to construct cells.

### 2D data preprocessing

As shown in Fig. 6, we construct the sensor data into an input format of 3 (channel) × 5 (height) × 140 (width). The length of one training sample data generated by each sensor on each axis is fixed at 140, the insufficient part is filled with zero, and the excess part will be discarded. Among them, 5 height corresponds to the data generated by 5 sensors, and 3 channels correspond to the data generated by 5 sensors on the three axes of $(x, y, z)$.

The advantage of this structure is that during the convolution operation, the convolution kernel can simultaneously acquire the data generated by the three axes at the same time,



| | | | | |
|---|---|---|---|---|
| $HX_1$ | $HX_2$ | ··· | $HX_{139}$ | $HX_{140}$ |
| $LX_1$ | $LX_2$ | ··· | $LX_{139}$ | $LX_{140}$ |
| $AX_1$ | $AX_2$ | ··· | $AX_{139}$ | $AX_{140}$ |
| $MX_1$ | $MX_2$ | ··· | $MX_{139}$ | $MX_{140}$ |
| $GX_1$ | $GX_2$ | ··· | $GX_{139}$ | $GX_{140}$ |

**Fig. 6** 2D residual cell data preprocessing

**Algorithm 1** Correlation feature maximization feature subset selection algorithm.

```
Input: $N_{\text{feature}}$ , $N_{\text{sensor}}$ , $S_{\text{feature}}$ , $W_{\text{feature}}$ , $M_{\text{subset}}$
Output: Number of features extracted from each feature group, $M_{\text{sensor}}$
function Correlation Feature Maximization Algorithm
$N_{\text{subset}} \leftarrow N_{\text{feature}} \;/\; N_{\text{sensor}}$
for $i \leftarrow 0$ to $N_{\text{sensor}}$ do
    for $j \leftarrow 0$ to $M_{\text{subset}}[i]$ do
        $P_{\text{sensor}}[i] \leftarrow P_{\text{sensor}}[i] + W_{\text{feature}}[i][j] * S_{\text{feature}}[i][j]$
        $P_{sum} \leftarrow P_{sum} + P_{\text{sensor}}[i]$
    end for
end for
for $k \leftarrow 0$ to $N_{\text{sensor}}$ do
    $M_{\text{sensor}}[k] \leftarrow P_{\text{sensor}}[k] \;/\; P_{sum} * N_{\text{subset}}$
end for
return $M_{\text{sensor}}$
```

and with the difference in the size of the convolution kernel, the receptive field can cover the data generated by other sensors in a similar time, too.

### Cell architecture details

Referring to previous work on Occam's Razor and ResNet [33], we choose ResNet-18, ResNet-50 and ResNet-101 to construct cell. Even though it is theoretically shown that the residual block will ensure that the network accuracy will at least not decrease as the network deepens, we still try shallow networks to improve the accuracy comparison. During training we use a batch size of 32 and use SGD with a momentum of 0.9 to initialize the weight. The learning rate is initialized to 0.01 and exponentially descends with a descending rate of 0.94. When the error plateaus, the learning rate will be divided by 10. Dropout is not used. The models are trained for up to 130 epochs.

## Traditional machine learning cell

In [35], it was shown that traditional machine learning methods can often achieve better results than deep neural networks on small sample datasets. For example, on the ORL dataset [36], 400 images are divided into 40 categories, and the Random Forest algorithm achieves better results than the deep neural network in classification, which is also confirmed in some of our previous work [22]. Our dataset is also a small sample dataset, so we built a traditional machine learning cell for experiments and comparisons.

### Feature extraction and concatenation

To increase the optional range and diversity of features, on the basis of the original data, we take $\text{MA} = \sqrt{\text{AX}^2 + \text{AY}^2 + \text{AZ}^2}$, $\text{MM} = \sqrt{\text{MX}^2 + \text{MY}^2 + \text{MZ}^2}$, $\text{MG} = \sqrt{\text{GX}^2 + \text{GY}^2 + \text{GZ}^2}$, $\text{MLA} = \sqrt{\text{LX}^2 + \text{LY}^2 + \text{LZ}^2}$, $\text{MHA} = \sqrt{\text{HX}^2 + \text{HY}^2 + \text{HZ}^2}$. For the above 20 sets of data, we calculated the minimum, maximum, mean, standard deviation, skewness and kurtosis values for each set of data. Therefore, for each training sample, we get a total of 120 features as shown in Table 1. For the convenience of the following, we assign a unique id to each feature in the Table 1. For example, FId.15 is the mean value of the $Z$ component of the acceleration measured by the accelerometer (AZ) and FId.95 is the skewness of the magnitude of the acceleration (MA). The feature id will be used instead of the long name of the feature in the following.

The feature fusion of the bio-behavioral authentication can increase the accuracy and reduce the redundancy of the data. Moreover, the earlier the data fusion, the better the effect [37], but due to the early sensor-level data fusion, a large amount of noise will be brought into the model, so sensor-level fusion often does not yield the best results. Therefore, the feature-level data fusion is considered to be a more effective choice for improving accuracy. We selects feature-level data fusion.

### Feature subset selection

Feature subset selection plays an important role in reducing data dimensions and preventing data overfitting. Based on the correlation between features and classes, a feature subset selection algorithm is proposed, named correlation feature maximization (CFM) feature subset selection algorithm (Algorithm 1).

Suppose the number of features is $N_{\text{feature}}$, CFM evaluates each feature according to the correlation between features and classes [38]. $N_{\text{feature}}$ correlation evaluation scores are obtained, formed array $S_{\text{feature}}$. The higher the score, the greater the correlation between the feature and the class. Suppose $N_{\text{sensor}}$ sensors are involved in the authentication movement, CFM regards features extracted from the same sensor as a set, totally $N_{\text{sensor}}$ sets of features. Suppose $M_{\text{subset}}$ is the number of features in each set of features. At the same time, CFM assigns a weight value to each feature according to the ranking of correlation evaluation scores, So there is the array $W_{\text{feature}}$. The higher the ranking, the bigger

**Table 1** The correspondence between feature name and feature id

| Feature name | AX | AY | AZ | MX | MY | MZ | GX | GY | GZ | LX |
|---|---|---|---|---|---|---|---|---|---|---|
| Minimum | 1 | 7 | 13 | 19 | 25 | 31 | 37 | 43 | 49 | 55 |
| Maximum | 2 | 8 | 14 | 20 | 26 | 32 | 38 | 44 | 50 | 56 |
| Mean | 3 | 9 | 15 | 21 | 27 | 33 | 39 | 45 | 51 | 57 |
| Standard deviation | 4 | 10 | 16 | 22 | 28 | 34 | 40 | 46 | 52 | 58 |
| Skewness | 5 | 11 | 17 | 23 | 29 | 35 | 41 | 47 | 53 | 59 |
| Kurtosis | 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 | 60 |
| Feature name | LY | LZ | HX | HY | HZ | MA | MM | MG | MLA | MHA |
| Minimum | 61 | 67 | 73 | 79 | 85 | 91 | 97 | 103 | 109 | 115 |
| Maximum | 62 | 68 | 74 | 80 | 86 | 92 | 98 | 104 | 110 | 116 |
| Mean | 63 | 69 | 75 | 81 | 87 | 93 | 99 | 105 | 111 | 117 |
| Standard deviation | 64 | 70 | 76 | 82 | 88 | 94 | 100 | 106 | 112 | 118 |
| Skewness | 65 | 71 | 77 | 83 | 89 | 95 | 101 | 107 | 113 | 119 |
| Kurtosis | 66 | 72 | 78 | 84 | 90 | 96 | 102 | 108 | 114 | 120 |

**Table 2** Feature subset in sitting/standing posture

| Sitting | Standing |
|---|---|
| FId | 19, 20, 21, 58, 4, 26, 27, 55, 76, 73, 1, 25, 97, 69, 32, 99, 15, 61, 98, 28, 33, 100, 77, 31, 62, 57, 8, 45, 9, 75, 3, 7, 63, 64, 22, 46, 10, 67, 68, 34, 70, 16, 14, 81, 112, 44, 74, 105, 94, 106, 93, 23, 52, 85, 56, 49, 13, 82, 80, 104 |
| | 102, 97, 103, 75, 109, 81, 91, 115, 57, 19, 49, 25, 31, 87, 63, 105, 33, 117, 21, 111, 27, 51, 69, 9, 3, 15, 45, 39, 43, 37, 13, 44, 38, 50, 8, 20, 14, 2, 7, 86, 80, 56, 68, 62, 26, 32, 61 |

the weight value is. The weight value depends on whether the user values the individual performance of individual sensors or the cooperative performance of all sensors. If users pay more attention to the individual performance of individual sensors, the features with high correlation score have greater weight value difference with those with low ranking. If users pay more attention to the performance of sensor cooperation, the features with high correlation score have smaller weight value difference with those with low ranking. The details of CFM are shown in Algorithm 1.

Finally, from each set of features, the features of the top $M_{\text{sensor}}$ scores are extracted and combined the feature subset (Table 2). This enables the CFM algorithm to filter out the best-performing features produced by each sensor and group them into feature subsets. Among them, in the sitting posture, the proportions of the features related to the high-pass filtered accelerometer, low-pass filtered accelerometer, accelerometer raw data, magnetometer and gyroscope to the total number of features of the feature subset are: 15%, 21.7%, 21.7%, 28.3%, 13.3%. In the standing posture, the proportions are 14.9%, 19.1%, 19.1%, 23.4%, and 23.4%, respectively. Each sensor data occupies an important proportion in the feature subset.

## Classifier selection

According to the feature subsets (Table 2), to find the most suitable classification algorithm, 5 commonly used classification algorithms [39] (Naive Bayes (NB), Bayes Net (BN), J48, Random Forest (RF) and Simple Logistic (SL)) were used to model and authenticate in the sitting and standing postures respectively.

## Cell evaluation

Cell evaluation will calculate the Score each cell gets:

$$Score = \frac{1}{N_{\text{param}} \times \text{Error}}, \tag{15}$$

where $N_{\text{param}}$ is the number of parameters involved in the cell calculation, and Error is the minimum error rate on the valid set during training. This module selects the cell with the highest score at the end of training and keeps its path, while closing the paths to other cells, ensuring that only one cell path will be left in the end.

## Performance evaluation

In this section, we first describe the details of our dataset composition and the collection of training and test samples. Secondly, we evaluate the performance of different cells of the model on the dataset, and conduct random attack experiments and imitation attack experiments.

Data collection used iPhone7 64 GB, with IOS 12.2 system. The calculation is performed on Windows 10, RTX3080Ti 32 GB RAM, 1TB hard disk.

### Data collection

We used the built-in accelerometer, magnetometer and gyroscope sensors to collect the data. The number of participants in the experiment was 138, 110 participants participated in the training set and testing set collection, 20 participants participated in the random attack experiment, and 8 participants participated in the imitation attack. The sampling rate of the accelerometer, magnetometer, and gyroscope was set to 100 Hz.

In response to our previous challenge of using the smallest possible dataset to achieve high-accuracy authentication, to compress the sample size required for each person's training as much as possible, we asked each of 110 participants to perform 5 times authentication movement in the sitting and standing positions. The duration of each movement was determined by the participants' personal habits, but was usually no more than two seconds (in fact, we founded that the longest movement time of the 110 participants was 1.38 s). Therefore, in the 1D data preprocessing section and 2D data preprocessing section, we set the data length of each axis of each sensor to 140, to ensure that all sample data can be included while padding as few zeros as possible. A total of 1100 training sample data were collected. We then asked 110 participants to perform 4 more times authentication movement in the sitting and standing positions to form the test dataset. Therefore, we got a total of 1980 training and testing samples, a total of 110 types of authentication movement datasets. We will perform a 110-class multi-classification task on this dataset.

To test RaiseAuth's resistance to random attacks, each of the 20 random attackers provided 50 random attacks in sitting posture and 50 random attacks in standing posture. Each random attack has the following characteristics: random attacker will try to make authentication movement with random behavior habits without knowing the 110 real user data. A total of 1000 random attack data of standing posture and 1000 random attack data of sitting posture were provided.

To test RaiseAuth's resistance to imitation attack, four pairs of people who were similar in height, weight, forearm length and upper arm length were required to make authentication movement. Each pair of people will have a real user and a imitated attacker. During the process of observing the real user's authentication movement, the imitated attacker tries to imitate the real user 20 times in the standing posture and sitting posture respectively. four pairs of people were provided with 80 attack data of standing posture imitation and 80 attack data of sitting posture imitation.

### Results

Based on the training set and testing set of 110 participants, we evaluated the performance of each Cell, and parts of the results are shown in Table 3. The column Params indicates the number of parameters involved in the calculation process of the Cell. The more the number of parameters, the more computing resources are required to implement training and testing. It can be seen that the network structure parameters in 1D multi-scale cell are significantly lower than 2D ResNet-18 cell, 2D Resnet-50 cell and 2D Resnet-101 cell. In terms of error rate, we selected three ways to evaluate. We took out the data of sitting posture, the data of standing posture, and the data of mixed sitting posture and standing posture for model training and testing respectively. From the results, we can see that even if the network structure includes a 1-dimensional shallow low-parameter neural network to a 2-dimensional 101-layer deep high-parameter network, their errors are basically the same. The depth of the network did not bring about a significant change. This is in agreement with our previous predictions, and with previous work in [35]. We believe that this is due to the order of magnitude of the sample data is too low, and the sample categories are too many, so that the deep neural network structure cannot obtain enough data to train the weight in the network. This makes the network usually overfit very quickly on the training set, but not well performance on the valid set and test set. However, we can still see from the results that compared with the standing posture, the sitting posture has a lower error rate, and the effect achieved by the mixed data set is comparable to the sitting posture. The relatively low error of 1D multi scale cell indicates that, compared with deepening the structure of the network, collecting more information from different scales is better. The richness of scales information can improve the training accuracy of small sample dataset.

Then, we tested the traditional machine learning cell. We selected a total of 5 algorithms (Naive Bayes (NB), Bayes Net (BN), J48, Random Forest (RF) and Simple Logistic (SL) ) as the classifier of the model, train the model under the standing and sitting posture dataset respectively, and select 5-fold cross-validation. It can be seen that the RandomForest classification algorithm has the best performance, achieving an accuracy of 97.49% in sitting posture and an accuracy of 94.68% in standing posture (Fig. 7). The accuracy is

**Table 3** Cell parameters and classification error in sitting posture, standing posture and combined dataset

| Cell | Params | Sit error (%) | Stand error (%) | Mix error (%) |
|---|---|---|---|---|
| 1D Multi Scale | 2,209,999 | 26.53 | 27.46 | 24.31 |
| 2D ResNet-18 | 11,233,455 | 30.41 | 31.55 | 29.28 |
| 2D ResNet-50 | 23,735,471 | 29.76 | 31.28 | 29.79 |
| 2D ResNet-101 | 42,727,599 | 31.33 | 31.39 | 30.82 |

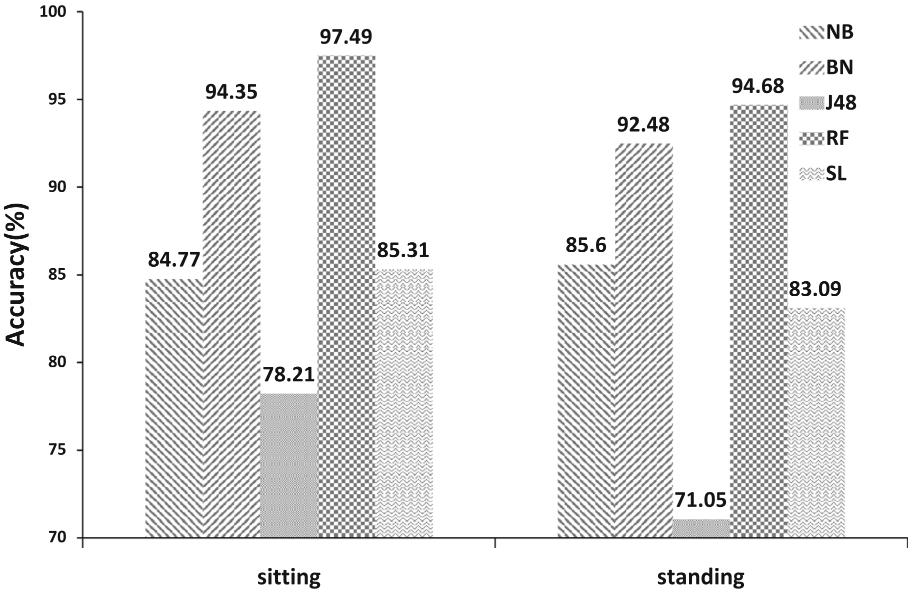


**Fig. 7** The accuracy results of RaiseAuth traditional machine learning cell combined with 5 classification algorithms in sitting/standing posture

$$\text{Accuracy} = \frac{\text{TPR} + \text{TNR}}{\text{TPR} + \text{FPR} + \text{FNR} + \text{TNR}}, \tag{16}$$

where TPR is true positive, FPR is false-positive rate, TNR is true-negative rate and FNR is false-negative rate.

At the same time, we selected five commonly used feature subset selection algorithms to compare with the CFM algorithm. We limit the number of features of the feature subset selection algorithms to have the same number of features as the CFM algorithm, and also use the RandomForest classifier. In the sitting posture, the accuracies of the ReliefF [40], GainRatio [41], InfoGain [42], SymmetricalUncert [43], oneR [44], and CFM (ours) algorithms are 96.98%, 97.21%, 97.21%, 97.35%, 97.38%, and 97.49%, respectively. In the standing posture, they were 93.17%, 93.55%, 93.17%, 94.49%, 93.24%, and 94.68%, respectively. This shows that our CFM algorithm has better performance on this task.

The False Positive Rate (FPR) and True Positive Rate (TPR) of the model are calculated and the Receiver Operating Characteristics (ROC) curve is drawn when the user is sitting posture (Fig. 8) and standing posture (Fig. 9). The ROC curves can visually display the FPR and TPR. The more the curve is "convex" to the upper left corner, the better the classifier effect.

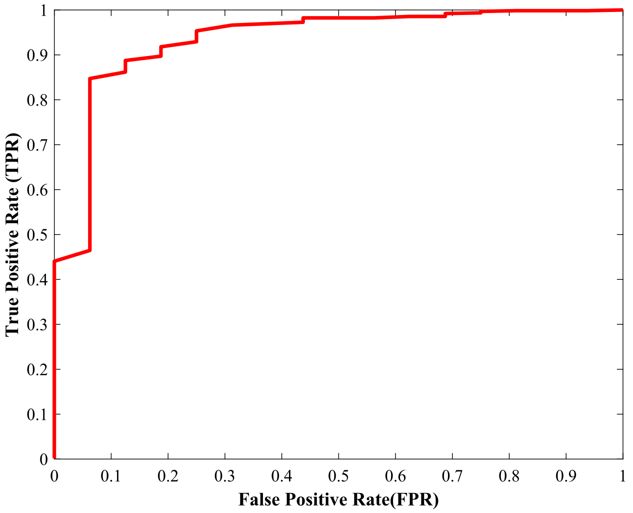


**Fig. 8** The ROC curve in the sitting posture of RaiseAuth

## Random attack test

To test the robustness and security of RaiseAuth, 20 attackers randomly performed 50 authentication movement data collections in both sitting and standing postures without knowledge of the 110 real user's information. This resulted in a total of 1000 sitting attack data and 1000 standing attack

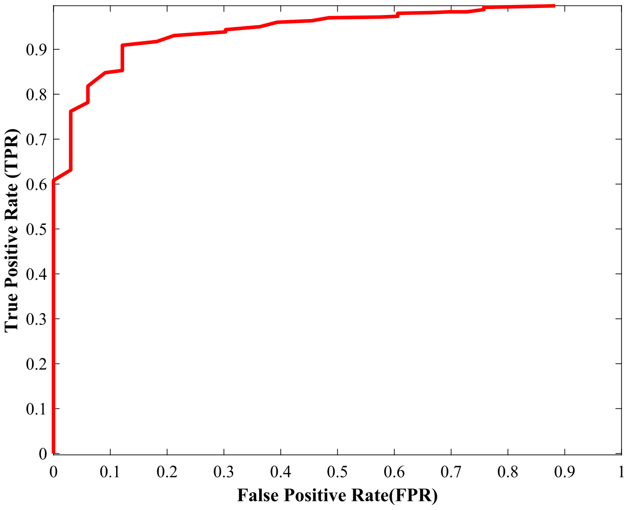


**Fig. 9** The ROC curve in the standing posture of RaiseAuth

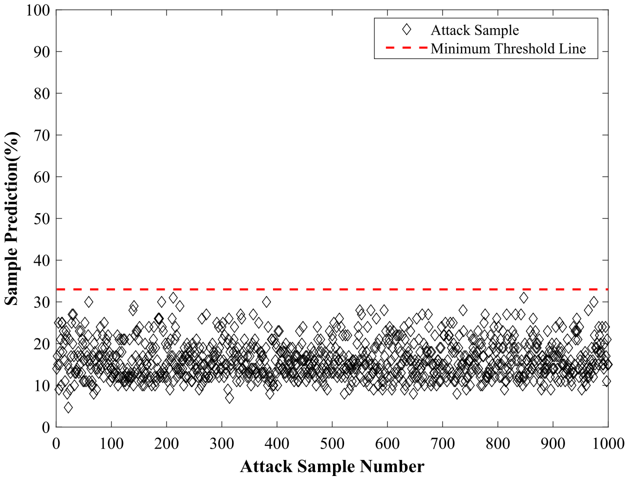


**Fig. 10** The sample predictions of 1000 random attacks calculated by RaiseAuth in sitting posture

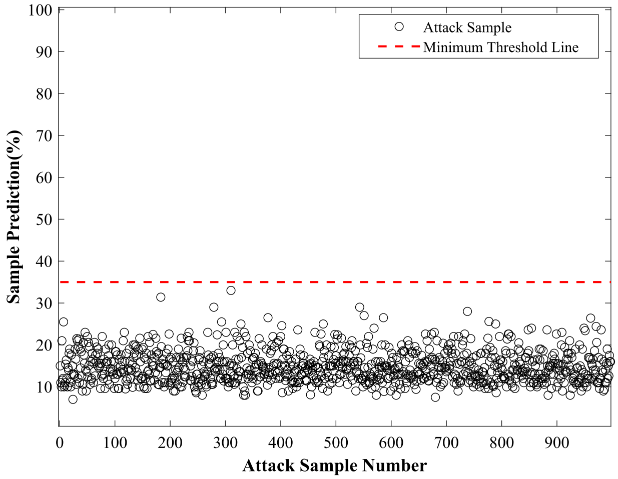


**Fig. 11** The sample predictions of 1000 random attacks calculated by RaiseAuth in standing posture

data. These 2000 attack data was used to test RaiseAuth, and the sample prediction of each random attack data was shown (Figs. 10, 11). The sample prediction represents how much confidence the model has in classifying samples to the current class. For example, 40% prediction means that the current sample matches one classified class by 40%. In the sitting posture, no attack prediction was more than 33%, and only 2 attacks had a prediction of 31%-35%. In the standing posture, no attack prediction was more than 35%, and only 3 attacks had an prediction of 31%-35%. The Minimum Threshold Line indicates how much the model needs to set the sample prediction rate threshold under attack to be able to resist the attack well. Usually, the threshold of sample prediction rate of real users is set to 90%, that is, when the model makes a judgment that the prediction is more than 90%, the judgment will be considered reliable. Our experiments prove that all 2000 random attack samples can be filtered out only by setting the minimum threshold line to 35%, which makes the difference between random attack samples and real user samples very obvious. This means that traditional machine learning cell in RaiseAuth structure perform well in resisting random attacks.

## Imitated attack test

Imitated attacks occur occasionally in real life, where attackers achieve identity obfuscation by observing and imitating real users. We tested the imitated attack resistance of the RaiseAuth. Participants were four pairs of people who were similar in height, weight, forearm length and upper arm length (Table 4). Four of them are real users and four are attackers. Real users normally collected authentication movement data, while attackers observed and imitated the collection process of real users. Next, each attacker imitated 20 attacks by imitating real users of similar body size in sitting (Fig. 12) and standing (Fig. 13) posture. In the results, the predictions of attacks were mostly concentrated in the 40% - 60% range, no one was more than 70%. The minimum threshold line is 70%.

At the same time, we analyzed the experimental data of real users and attackers. We showed the two best features in the standing posture (Fig. 14) as an example. This data comes from a pair of Real user1 and Attacker 1, of which the lower left corner is the value of FId.19 and FId.21 generated after 20 authentication movements by real user1. The upper right corner is the value of FId.19 and FId.21 generated by the imitator Attacker 1 imitating the real user1 to make 20 imitation movements. It could be seen that even though the real user and the attacker were very similar in body size, and the attacker was given 20 opportunities to observe and imitate, the difference between them was still very obvious for RaiseAuth.

**Table 4** Body size of attacker and real user in imitated attack test

| Role | Height (cm) | Weight (kg) | Upper arm length (cm) | Forearm length (cm) | Habitual hand | Gender |
|---|---|---|---|---|---|---|
| Real user 1 | 185 | 82.9 | 23 | 49 | Right | Male |
| Attacker 1 | 185 | 83.5 | 25 | 49 | Right | Male |
| Real user 2 | 188 | 78.2 | 28 | 47 | Right | Male |
| Attacker 2 | 186 | 80.3 | 27 | 47 | Right | Male |
| Real user 3 | 176 | 65 | 30 | 44 | Left | Male |
| Attacker 3 | 175 | 60 | 28 | 47 | Left | Male |
| Real user 4 | 162 | 52 | 26 | 38 | Right | Female |
| Attacker 4 | 161 | 45 | 26 | 38 | Right | Female |

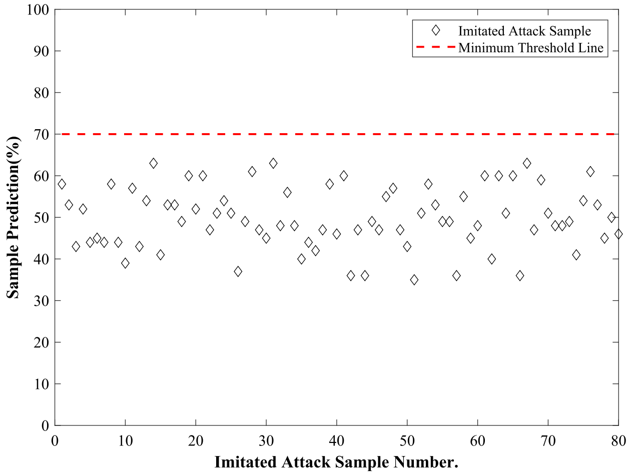


**Fig. 12** The sample predictions of imitated attacks calculated by RaiseAuth in sitting posture

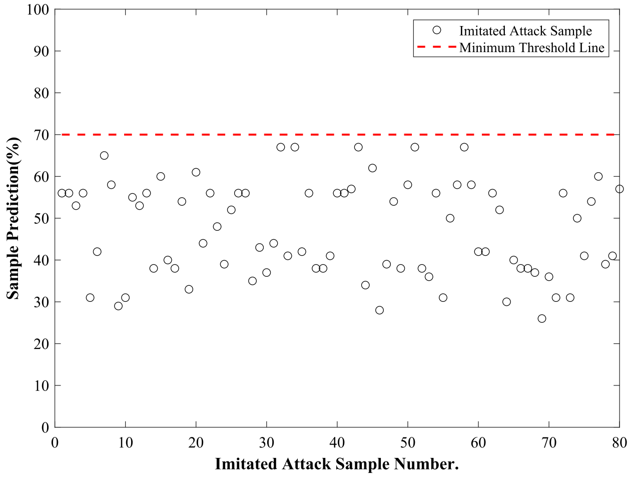


**Fig. 13** The sample predictions of imitated attacks calculated by RaiseAuth in standing posture

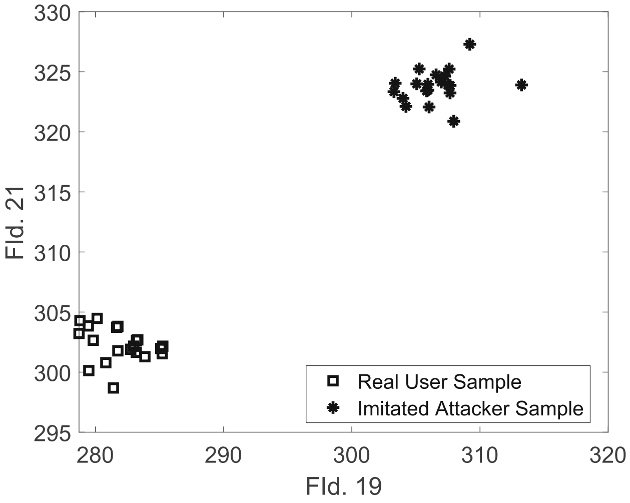


**Fig. 14** Feature Id 19's and Feature Id 21's data visualization of a pair of participants in imitated attack test

## Conclusions

In this work, we aim to achieve high-accuracy authentication while compressing the movement complexity of the authentication movement as much as possible. The contribution of this work is twofold.

First, based on the mathematical modeling of the arm, we constructed the movement state function, and according to the parameters of the movement state function, the complexity of the authentication movement was maximally compressed.

Second, to the best of our knowledge, we are the first to authenticate against an authentication movement with such low movement complexity. We constructed an authentication dataset involving a total of 138 people. At the same time, based on this dataset, we designed an authentication method named RaiseAuth with multiple Cells.

Through the analysis of the experimental results, we can draw the following two points, (1) RaiseAuth is better than deep neural network cell when paired with traditional machine learning cell, which means that in the face of small samples and multi-category dataset, traditional machine learning algorithms are more effective. This is in agreement with previous work in [35]; (2) RaiseAuth performs well against random attacks that do not know the real user information, but it suffers a certain impact when confronting imitators with similar physical parameters that stare at real users and conduct 20 imitation attacks. This shows that under extreme conditions, authentication based on body movements may be affected by imitated attacks.

In summary, RaiseAuth can perform multi-classification tasks with good performance on the 110-class authentication dataset with low movement complexity and small samples.

**Funding** No funding was received to assist with the preparation of this manuscript.

## Declarations

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

## References

1. Stylios I, Kokolakis S, Thanou O, Chatzis S (2021) Behavioral biometrics and continuous user authentication on mobile devices: a survey. Inf Fusion 66:76–99. https://doi.org/10.1016/j.inffus.2020.08.021
2. Shen C, Zhang Y, Guan X, Maxion RA (2016) Performance analysis of touch-interaction behavior for active smartphone authentication. IEEE Trans Inf Forensics Secur 11(3):498–513. https://doi.org/10.1109/TIFS.2015.2503258
3. Matyas V, Riha Z (2003) Toward reliable user authentication through biometrics. IEEE Secur Privacy 1(3):45–49. https://doi.org/10.1109/MSECP.2003.1203221
4. Stylios I, Kokolakis S, Thanou O, Chatzis S (2016) Users' attitudes on mobile devices: can users' practices protect their sensitive data?
5. Muslukhov I, Boshmaf Y, Kuo C, Lester J, Beznosov K (2012) Understanding users' requirements for data protection in smartphones. In: 2012 IEEE 28th international conference on data engineering workshops, pp 228–235. https://doi.org/10.1109/ICDEW.2012.83
6. Lu L, Liu Y (2015) Safeguard: user reauthentication on smartphones via behavioral biometrics. IEEE Trans Comput Soc Syst 2(3):53–64. https://doi.org/10.1109/TCSS.2016.2517648
7. Bharath M, Rao KR (2022) A novel multimodal hand database for biometric authentication. Int J Adv Technol Eng Explor 9(86):127
8. Ahmed MA, Roushdy M, Salem A-BM (2022) Multimodal technique for human authentication using fusion of palm and dorsal hand veins. In: Kountchev R, Mironov R, Nakamatsu K (eds) New Approach Multidimens Signal Process. Springer, Singapore, pp 63–78
9. Ramya S, Sheeba R, Aravind P, Gnanaprakasam S, Gokul M, Santhish S (2022) Face biometric authentication system for atm using deep learning. In: 2022 6th international conference on intelligent computing and control systems (ICICCS). IEEE, pp 1446–1451
10. Dar AS, Palanivel S (2022) Real time face authentication system using stacked deep auto encoder for facial reconstruction. Int J Thin Film Sci Technol 11(1):9
11. Lee YW, Park KR (2022) Recent iris and ocular recognition methods in high-and low-resolution images: a survey. Mathematics 10(12):2063
12. Chen Q, Huang M, Wang H (2021) A feature discretization method for classification of high-resolution remote sensing images in coastal areas. IEEE Trans Geosci Remote Sens 59(10):8584–8598. https://doi.org/10.1109/TGRS.2020.3016526
13. Su Y, Ma K, Zhang X, Liu M (2022) Neural network-enabled flexible pressure and temperature sensor with honeycomb-like architecture for voice recognition. Sensors 22(3):759
14. Alkhammash EH, Hadjouni M, Elshewey AM (2022) A hybrid ensemble stacking model for gender voice recognition approach. Electronics 11(11):1750
15. Shen C, Yu S, Wang J, Huang GQ, Wang L (2022) A comprehensive survey on deep gait recognition: algorithms, datasets and challenges. arXiv preprint arXiv:2206.13732
16. Gonçalves Filipi, dos Santos C, Oliveira DdSA, Passos L, Gonçalves Pires R, Felipe Silva Santos D, Pascotti Valem LP, Moreira T, Cleison S, Santana M, Roder M, Paulo Papa J (2022) Gait recognition based on deep learning: a survey. ACM Comput Surv (CSUR) 55(2):1–34
17. Roy S, Pradhan J, Kumar A, Adhikary DRD, Roy U, Sinha D, Pal RK (2022) A systematic literature review on latest keystroke dynamics based models. IEEE Access
18. Li G, Sato H (2022) Sensing in-air signature motions using smartwatch: a high-precision approach of behavioral authentication. IEEE Access
19. Chen Q, Huang M, Wang H, Xu G (2022) A feature discretization method based on fuzzy rough sets for high-resolution remote sensing big data under linear spectral model. IEEE Trans Fuzzy Syst 30(5):1328–1342. https://doi.org/10.1109/TFUZZ.2021.3058020
20. Yang Y, Guo B, Wang Z, Li M, Yu Z, Zhou X (2019) Behavesense: continuous authentication for security-sensitive mobile apps using behavioral biometrics. Ad Hoc Netw 84:9–18
21. Chen Q, Ding W, Huang X, Wang H (2022) Generalized interval type II fuzzy rough model based feature discretization for mixed pixels. IEEE Trans Fuzzy Syst, 1–15. https://doi.org/10.1109/TFUZZ.2022.3190625
22. Zhao S, Guo Z, Zhong C, Xian L, Liu Y (2020) A novel smartphone identity authentication mechanism. In: Proceedings of the ACM turing celebration conference-China, pp 157–161
23. Dong G, Tang M, Wang Z, Gao J, Guo S, Cai L, Gutierrez R, Campbell B, Barnes LE, Boukhechba M (2022) Graph neural networks in IoT: a survey. arXiv preprint arXiv:2203.15935
24. Hong F, Wei M, You S, Feng Y, Guo Z (2015) Waving authentication: your smartphone authenticate you on motion gesture. In: Proceedings of the 33rd annual ACM conference extended abstracts on human factors in computing systems, pp 263–266
25. He L, Ma C, Tu C, Zhang Y (2022) Gait2vec: continuous authentication of smartphone users based on gait behavior. In: 2022 IEEE 25th international conference on computer supported cooperative work in design (CSCWD). IEEE, pp 280–285

26. Dee T, Richardson I, Tyagi A (2022) Continuous nonintrusive mobile device soft keyboard biometric authentication. Cryptography 6(2):14
27. Huang C, Zhang F, Xu Z, Wei J (2022) The diverse gait dataset: gait segmentation using inertial sensors for pedestrian localization with different genders, heights and walking speeds. Sensors 22(4):1678
28. Li Y, Hu H, Zhu Z, Zhou G (2020) Scanet: sensor-based continuous authentication with two-stream convolutional neural networks. ACM Trans Sens Netw (TOSN) 16(3):1–27
29. Abuhamad M, Abuhmed T, Mohaisen D, Nyang D (2020) Autosen: deep-learning-based implicit continuous authentication using smartphone sensors. IEEE Internet Things J 7(6):5008–5020
30. Li G, Bours P (2018) A novel mobilephone application authentication approach based on accelerometer and gyroscope data. In: 2018 International conference of the Biometrics Special Interest Group (BIOSIG). IEEE, pp 1–4
31. Buriro A, Crispo B, Zhauniarovich Y (2017) Please hold on: unobtrusive user authentication using smartphone's built-in sensors. In: 2017 IEEE international conference on identity, security and behavior analysis (ISBA). IEEE, pp 1–8
32. Dybczak J, Nawrocki P (2022) Continuous authentication on mobile devices using behavioral biometrics. In: 2022 22nd IEEE international symposium on cluster, cloud and internet computing (CCGrid). IEEE, pp 1028–1035
33. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
34. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. PMLR, pp 448–456
35. Bi Y, Xue B, Zhang M (2020) Genetic programming with a new representation to automatically learn features and evolve ensembles for image classification. IEEE Trans Cybern 51(4):1769–1783
36. Samaria FS, Harter AC (1994) Parameterisation of a stochastic model for human face identification. In: Proceedings of 1994 IEEE workshop on applications of computer vision. IEEE, pp 138–142
37. Gofman MI, Mitra S, Cheng T-HK, Smith NT (2016) Multimodal biometrics for enhanced mobile device security. Commun ACM 59(4):58–65
38. Hall MA (2000) Correlation-based feature selection of discrete and numeric class machine learning
39. Choi H-S, Lee B, Yoon S (2016) Biometric authentication using noisy electrocardiograms acquired by mobile sensors. IEEE Access 4:1266–1273
40. Robnik-Sikonja M, Kononenko I (1997) An adaptation of relief for attribute estimation in regression. In: Proceedings of the fourteenth international conference on machine learning. ICML '97. Morgan Kaufmann Publishers Inc., San Francisco, pp 296–304
41. Pehlivan U, Baltaci N, Acartürk C, Baykal N (2014) The analysis of feature selection methods and classification algorithms in permission based android malware detection. In: 2014 IEEE symposium on computational intelligence in cyber security (CICS), pp 1–8 . https://doi.org/10.1109/CICYBS.2014.7013371
42. Zhao Z, Morstatter F, Sharma S, Alelyani S, Anand A, Liu H (2010) Advancing feature selection research. ASU Feature Selection Repository Arizona State University, pp 1–28
43. Witten IH, Frank E, Hall M (2014) A data mining: practical machine learning tools and techniques
44. Holte RC (1993) Very simple classification rules perform well on most commonly used datasets. Mach Learn 11(1):63–90. https://doi.org/10.1023/A:1022631118932