Motion Analysis Based on Inertial Measurement Unit Sensors: Guidance to Novice Ping-Pong Players' Training

Athletics & Sensing Devices

Project Report by Team P³

Xinlei Zhang Ze`an He Haokang Hu

202030101256 202030020342 202030020359

SHIEN-MING WU SCHOOL OF INTELLIGENT ENGINEERING

Abstract

With the development of wearable technology and inertial sensor technology, the application of wearable sensors in the field of sports is becoming more extensive. IMU (Inertial Measurement Unit) is a sensor that measures triaxial accelerations and triaxial angular velocities and it has been integrated in many devices such as smart phone, smart watch and other integrated sensors. In this project, we use two devices containing IMU sensor fixed with the subject to collect inertial characteristic data while the subject is playing table tennis. In the data preprocessing, we conduct time-series segmentation, features selection and dimension reduction using PCA (Principal Component Analysis). Then, we directly choose several important principal components as new features feeding into machine learning models. We use ANN (Artificial Neuron Network) and DT (Decision Tree) to realize binary motion recognition and compare their experimental results. In addition, based on OCSVM (One Class Support Vector Machine) learning model, we also achieve the faulty motion detection which may cause damage to players' wrist.

Keywords: Wearable technology; Inertial Measurement Unit; Table Tennis; Motion Classification; Faulty Motion Detection; Data Mining.

1. Introduction

As a China 'national sport', table tennis is one of the most common sports in Chinese Society. According to surveys, there are millions of people in China playing Ping-Pong from teenagers to elders. It's feasible for table tennis players to apply such a device containing an IMU sensor during playing to recognize motions and detect strokes and mark faulty motion, so as to record the inertial characteristic data and improve player's skills.

In literature, on the topic of table tennis, many researches have been conducted using wearable sensors to monitor players during sports and provide some useful information, such as scoring the forehand loop training [1], motion recognition [2], and shot-detection [3]. That information can assist physical education teachers, improving the effectiveness of trainers' learning and allowing them to practice sports-related skills more easily.

In addition to athletes and experts, there's also lots of crucial guidance could be

suggested by such sensors system for novices, such as detecting and noticing some faulty, harmful movements, which is beneficial to their long-term training. In this project, aiming at improving the quality of training for novice ping-pong players, we plan to use the data from IMU sensors to achieve the basic motion analysis of novices and mark faulty movement patterns during their training.

2. System Design & Modeling

In this section, we describe the complete procedure of our project, from problem formulization to each step of designing the whole system.

2.1 Problem Modeling

Table tennis is a vigorous competitive sport, which requires the cooperation of whole-body muscle and skillful techniques. The basic technique motions to stroke the ball can be categorized as forehand stroke and backhand push. In intuition, the technical movements of players while they're playing table tennis should dynamically impact the inertial characteristic of their bodies and the patterns of these inertial characteristic should be quite various in different kinds of motions. Based on the recorded characteristic, we preprocess the inertial measurement data and extract every single technical movement from the raw signals. Then, we use two classification algorithm, ANN and DT to recognize the forehand stroke motion and backhand push motion. At last, we use OCSVM algorithm to mark the faulty patterns in one stroke which means players rotate their wrist during stroking.

2.2 Data Collection

The subjects in this project were four novice male table tennis players. The IMU integrated devices used are a smart phone and BWT901CL. Before data collection, subjects were required to attach the smart phone and the BWT901CL to the upper arm and the back of the hand, respectively. These two devices were fixed to a certain position relative to the subject every time to ensure the data consistency. The place of collection was chosen at the table tennis court at GZIC. The subjects were informed of the purpose and content of the project. The collection was conducted using a table tennis serving machine, which could serve at a constant frequency and quality. Figure1 below shows the IMU devices used in the project, the placement of the devices and the serving machine.

Figure 1(a) The subject is playing table tennis and the inertial devices are attached to his body

Figure 1(b): The screenshot of smart phone and BWT901CL

Figure 1

2.3 System Overview

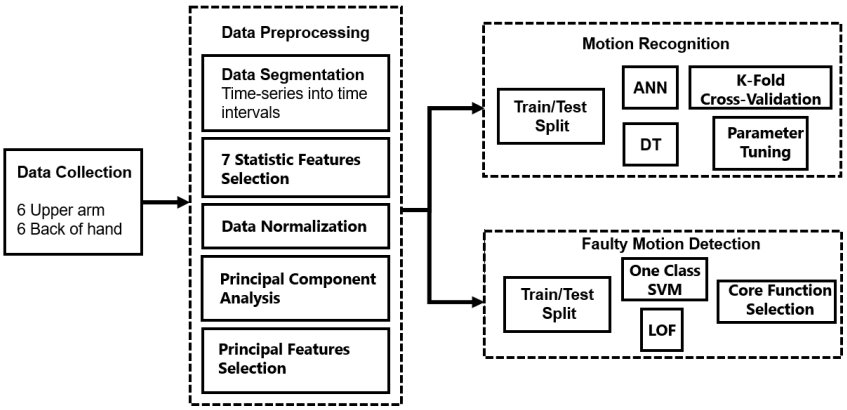

Figure 2: The Flowchart of Our Whole System Design

2.4 Data Preprocessing

In this section, we segment the time-series data into lots of time intervals and each time interval is corresponding to each complete motion, including swing phase and stroking phase. We use seven kinetic statistic parameters of each time interval to represent this motion. Then we conduct PCA (Principal Component Analysis) to achieve the dimension reduction and directly choose important principal components as new features feeding into machine learning model.

2.4.1 Data Segmentation

Raw data is time-series data, which starts before the first stroking and still records the data after the last stroking until the devices are paused. At first, we manually remove the beginning and ending parts which are obvious to distinguish when the data are visualized on the screen. Then we choose the most stably and uniformly changing inertial features from the total 12 inertial features.

Table 1: The 12 inertial characteristic data

	Inertial Features					
Upper Arm	Accel_x	Accel_y	Accel_z	W_x	W_y	W_z
Back of Hand	Accel_x	Accel_y	Accel_z	W_x	W_y	W_z
Accel_x, Accel_y, Accel_z are triaxial accelerations; W_x, W_y, W_z are triaxial angular velocity						

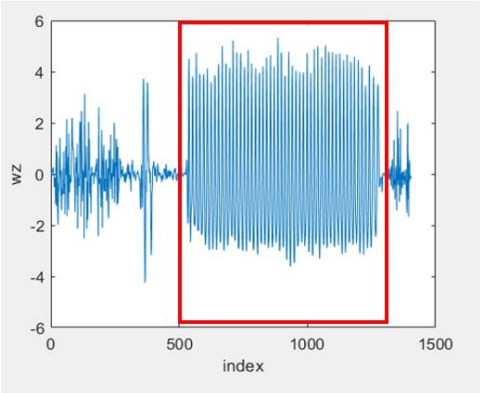

Figure 3(a) The Visualization of removing the beginning and ending part

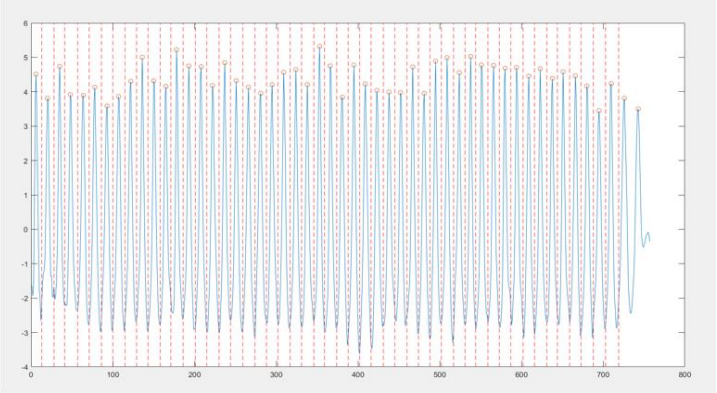

Figure 3(b) The Visualization of peaks-widths method dividing each motion

Figure 3

After that, we divide each motion from the whole interval according to the peaks and the average width. We find the locations of peaks and then calculate the average distance of these peaks, regarding the average distance as the width of each motion.

$$Width\ of\ motion = average\big(difference(locations)\big) \tag{1}$$

2.4.2 Seven Statistic Kinematic Parameters Selection

Statistic kinematic parameters have been used to analyze the motion of table tennis [4]. After data segmentation, the collected raw data needs to be converted into a numerical matrix for subsequent recognition and detection processing. Because the length of each motion sample varies from sample to sample, we carry out statistic features extraction. We use all the data in each motion sample to calculate the mean value, variance, kurtosis, skewness, maximum value, minimum value and energy, as shown in table 2, for the total 12 inertial features, as shown in table 1.

Table 2: Seven Statistic Kinematic Features extracted from inertial data

Statistical Characteristics	Computational Formula
Mean Value	$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$
Variance	$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$
Kurtosis	$K = \frac{\sum_{i=1}^{n}(x_i - \bar{x})\, f_i}{ns^4}$
Skewness	$SK = \frac{n\sum_{i=1}^{n}(x_i - \bar{x})}{(n-1)(n-2)s^3}$
Maximum Value	$Max(X_i)$
Minimum Value	$Min(X_i)$
Energy	$E = \sum_{i=1}^{n} X_i^2$

2.4.3 Data Normalization

Feature normalization is normally necessary, which can restrict the characteristic values to a certain range with little numerical difference, and the characteristics of the sample data represented by each feature are not changed. Besides, data normalization also improves the efficiency of subsequent machine learning model. Small scalar input values always make the model converges faster than the raw data. Common methods of feature normalization have their own metrics. In this project, we choose the linear normalization method, as follows:

$$\dot{X}_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2}$$

2.4.4 Principal Component Analysis

PCA method is conducted in this project. PCA is an unsupervised learning of information measured by variance, which is not subjected to sample label limit. And the principal components are orthogonal to each other after dimensionality reduction, which can eliminate the dependence between the components of the original features [4]. Also, unusual variation, not apparent from the original features will often show up in low-variance components, which are highly informatively in our subsequent faulty motion detection.

$$Principal\ Component = [w_1\ w_2\ w_3\ \dots w_{84}]\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \dots \\ X_{84} \end{bmatrix} \tag{3}$$

Formula 3 shows that each principal component is the linear combination of all original features.

Figure 4 below shows the compute explained variance of each principal component and the cumulative variance of all 84 principal components.

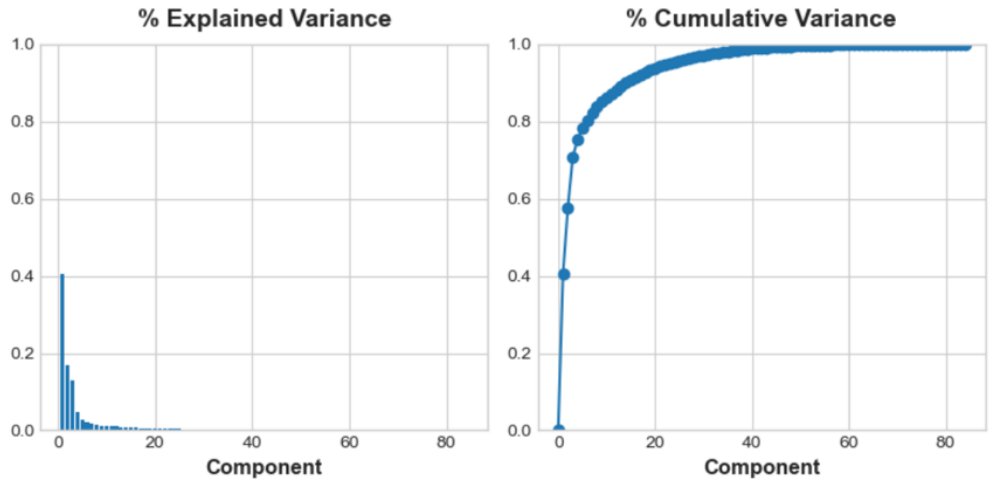

Figure 4(a) The Explained Variance of each Principal Component

Figure 4(b) The Cumulative Variance of 84 Principal Components

Figure 4

2.4.5 Principal Features Selection

After principal component analysis, we can get principal components which are equal in amount to preceding statistical kinematic features. Due to the total number of 84 (2 devices×6 inertial features×7 Statistical Kinematic parameters) features, we score each features according to mutual information and set a threshold to select more informative features, so as to reduce the input dimensions of machine model.

Mutual information describes relationships in terms of uncertainty. The mutual information between two quantities is a measure of the extent to which knowledge of one quantity reduces uncertainty about the other.

$$Mutual\ Information = H(X) - H(X|Y) \tag{4}$$

$$H(X) = -\sum_{i=1}^{n} P(X = i)\log_2 P(X = i) \tag{5}$$

$$H(X|Y = v) = -\sum_{i=1}^{n} P(X = i|Y = v)\log_2 P(X = i|Y = v) \tag{6}$$

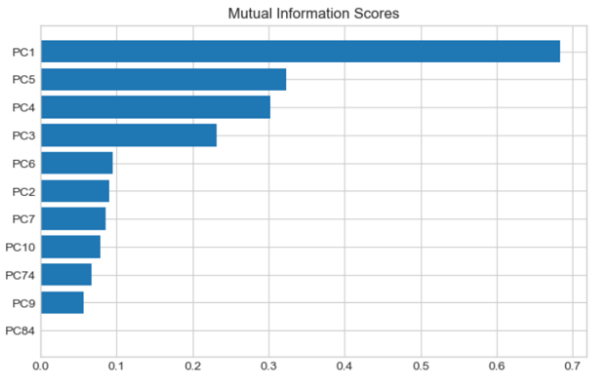

Figure 5: The Mutual Information of All 84 Principal Component features

Table 3: Mutual Information of Principal Components' Features

PC1	0.6841	PC5	0.3147
PC4	0.3011	PC3	0.2257
PC6	0.0983	PC2	0.091
PC7	0.0892	PC10	0.0811
PC74	0.0698	PC9	0.0554

Figure 5 shows that the mutual information scores of the first ten principal component are relatively large than others. Thus, we set the threshold of mutual information score is equal to 0.5.

2.5 Motion Classification

2.5.1 Artificial Neural Network

2.5.1.1 Algorithm Structure

Due to the preprocessed data having 11 effective feature values, we build an input layer with 11 input number. Besides, we build a hidden layer with 6 nodes. Also, owing to the 2-classes classification problem, we determine the number of output layer is 2. The whole structure is the following figure 6. The natural network is a full connected network.

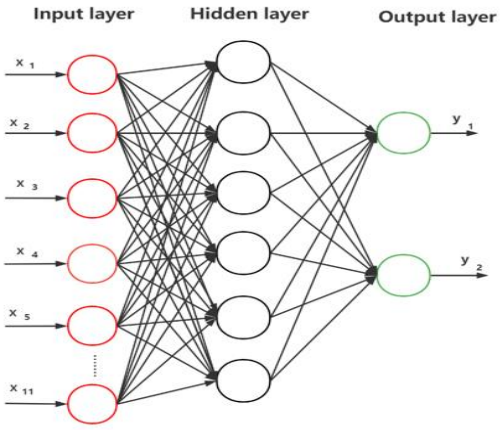

Figure 6: The completely connected neural network

Besides, we use K-fold Cross-Validation to take full advantage of our data set (800 data is not fully enough). Firstly, we separate data set as train set called train set_1 (account for 70%) and test set (account for 30%). Then, we divide train set_1 into 4 parts averagely, use 3 parts as train set to train ANN model and the other set as verification set, every time. The operation will run 4 times, which will choose different train sets and verification set every time. As for the difference of train set, the model of ANN is also different. So, we use every model to predict test set and will gain 4 results. We consider the final result as the mean of the 4 results. In addition, we choose cross entropy as the function of loss calculation [5], during the operational process. The function is:

$$loss(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) = -x[class] + \log\left(\sum_j exp(x[j])\right) \quad (7)$$

We calculate the loss 70 times and update the weight of every input by using BP (Back Propagation) algorithm to find the optimal solution. Also, we use sigmoid function as activation function,

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \quad (8)$$

To use more information of feature value and not determined by one or two feature value, L2 regularization are used, which will punish large weight more severely than small weight. As a result, the generalization ability of the program will promote. The complete process of the ANN algorithm is,

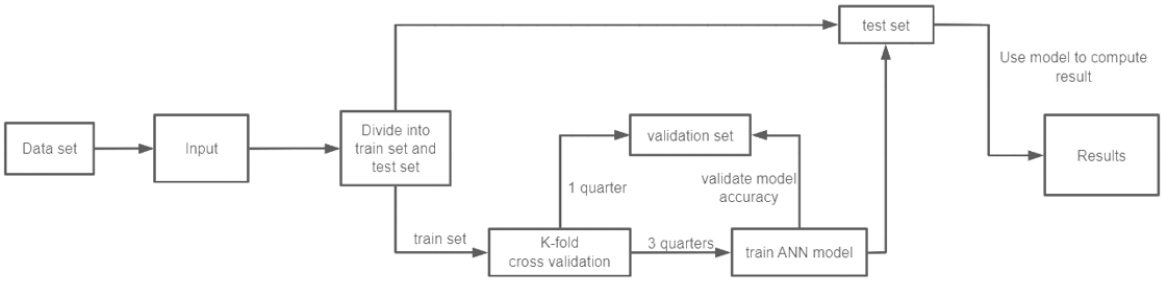

Figure 7: Flowchart of the ANN Learning Model

2.5.1.2 Hyper-Parameter Determination

The number of hidden layer nodes are confirmed by the equation [6],

$$N_h = \frac{N_S}{\left(a \times (N_i + N_O)\right)} \quad (9)$$

$$where\ N_i: the\ number\ of\ input\ layer\ nodes$$
$$N_0: the\ number\ of\ output\ layer\ nodes$$
$$N_s: the\ number\ of\ samples$$
$$N_h: the\ number\ of\ hidden\ layer\ nodes$$
$$a: a\ number\ between\ 2\ and\ 10$$

Calculated according to the equation, N_h is between 4 and 15. In addition, due to the principle of build ANN (choose the simplest structure when two structure have same effect), we choose 6 nodes as we hidden layer.

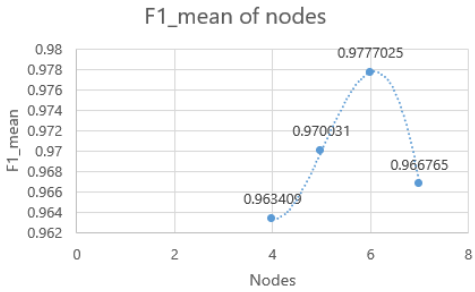

Figure 8(a): The F_1mean of two classes

```
The mean train accuracy is 0.985611510791367
The mean test accuracy is 0.9728033472803348
The precision of forehard is  0.9704433497536946
The precision of backhand is  0.9745454545454545
The recall of forehand is  0.9656862745098039
The recall of backhand is  0.9781021897810219
```

Figure 8(b): The Performance of 2 hidden layers with 6 nodes in each layer

Figure 8

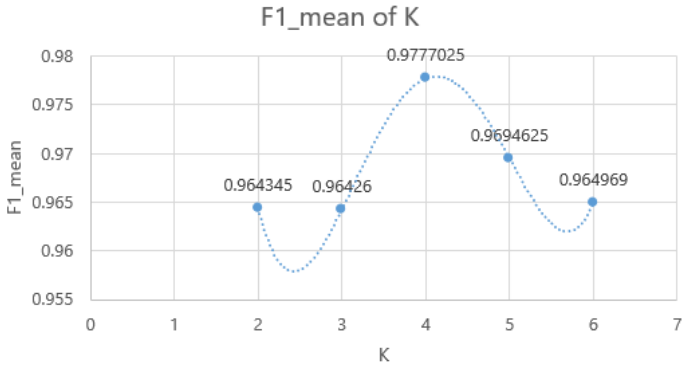

Figure 9: The F_1 mean of the value of k

We use the mean F1 of two classes as the stander of classifier. The larger mean F1 classifier gains, the better classifier is. So, from the figure 8 (a), we can know that when the number of nodes equal to 6, the mean F1 is largest. When N_h less than 6, the model is underfitting. When N_h larger than 6, the model is overfitting. The number of hidden layers is also a hyper-parameter. In this algorithm, we build 1 hidden layer. Actually, for this problem, one hidden layer is enough. We had built a 2 hidden layers structure model with 6 nodes every layer. The result is shown in figure 8 (b).

Compared with 1 hidden layer structure model, the mean train accuracy is higher, but the mean test accuracy and mean F1 is lower, which is typical overfitting. More than 2 hidden layers structure is no need to consider [7].

The k in K-fold Cross-Validation is another parameter. Usually, the number of k is larger than 2. we have tried 3, 4, 5, 6 four cases. According to figure 9, it's easy to know, when k equal to 4, the result is best. So far, all the parameters in algorithm are determined and best.

2.5.2 Decision Tree

The other algorithm we used to classify is DT. The 11 feature values are the input. The output are 2 classes, forehand and backhand motion. We divide data set into train set (account for 50%) and test set (account for 50%). Use Gini coefficient to classify. Set the maximum depth is 10, for a large maximum depth will result to overfitting. When Gini coefficient of all nodes equal to 0 or the maximum depth is larger than 10, the program will stop.

2.6 Faulty Motion Detection

Unlike the preceding models, which distinguish forehand and backhand motions, this model is used to find the faulty motion. To better solve the problem, we choose algorithms used in Anomaly Detection. In figure 10, we can get three classes of data: forehand, backhand and known fault, including both forehand and backhand. But there are many data that we have not obtained: Unknown-Fault, which includes all other types. Thus, the Binary Classification Algorithm may not work here and anomaly detection can do a better job.

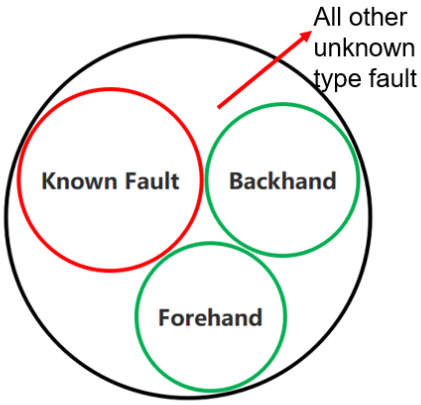

Figure 10: The Venn Diagram of data

2.6.1 Algorithm Introduction

Algorithms we choose for this model are: One-Class SVM and Local Outliers Factor (LOF). The following sections introduce these two algorithms.

2.6.2 One-Class Support Vector Machine

One-Class SVM is an unsupervised learning algorithm, which based on the characteristics of normal data. Data with similar characteristics to normal data is considered as normal data, otherwise it is considered as abnormal data. One-Class SVM do a good job in high dimensional space.

The Core Code:

one_svm = OneClassSVM (nu=0.12, kernel='rbf')

where nu is an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors. (nu∈(0,1)) When nu=0.12, it means it tries to find about 12% faulty data in the test set.

For kernel in this algorithm, we choose the most suitable one: rbf (Radial basis function).

$$\kappa(\boldsymbol{X_1}, \boldsymbol{X_2}) = \big(\phi(\boldsymbol{X_1}), \phi(\boldsymbol{X_2})\big) = \exp\left(-\frac{\big||\boldsymbol{X_1} - \boldsymbol{X_2}|\big|^2}{2\sigma^2}\right) \tag{10}$$

Main idea of kernel function is mapping the vector from ***p*** in the low dimension to ***h*** in the high dimension, so that the case of linear non separability in the low dimension can become linearly separable in the high dimension.

2.6.3 Local Outliers Factor

LOF algorithm is an unsupervised algorithm, which is based on density. LOF algorithm is suitable for anomaly detection of data with different density.

The Core Code：

```
LOF = sklearn.neighbors.LocalOutlierFactor(
    contamination=0.08, n_neighbors=20, novelty=True)
```

where, n: If n_neighbors is larger than n, all samples will be used.

Contamination: The amount of contamination of the data set, like the proportion of outliers in the data set, which is Similar to nu in One-Class SVM.

Novelty: If use LocalOutlierFactor for novelty detection, novelty = True.

Main idea：

LOF algorithm judges whether each point P is an anomaly by comparing the density of each point P and its neighborhood points: the lower the density of point P, the more likely it is to be an anomaly. The density of points is calculated by the distance between points. The farther the distance between points, the lower the density; The closer the distance, the higher the density. In other words, the density of points in LOF algorithm is calculated by the K neighborhood of points, not by global calculation.

3. Experimental Results & Evaluation

Due to the use of K-fold Cross-Validation, the model has trained 4 kinds of ANN model using 4 different train set. Thus, we calculate the change of loss and accuracy of validation set. After many experiments, we find that when the iterations are between 50 and 70 the loss and the accuracy will converge. Then, we use the final model to predict the test set and compute the accuracy and confusion matrix each time. we print when epochs larger than 30, every 3 epochs. Besides, we compute the mean confusion matrix and mean accuracy of 4 model to reduce the error. Using mean confusion matrix, we can calculate the precision and recall score of forehand and backhand. (0 means forehand and 1 means backhand in thermodynamic diagram of mean confusion matrix) The following figures shows the final result.

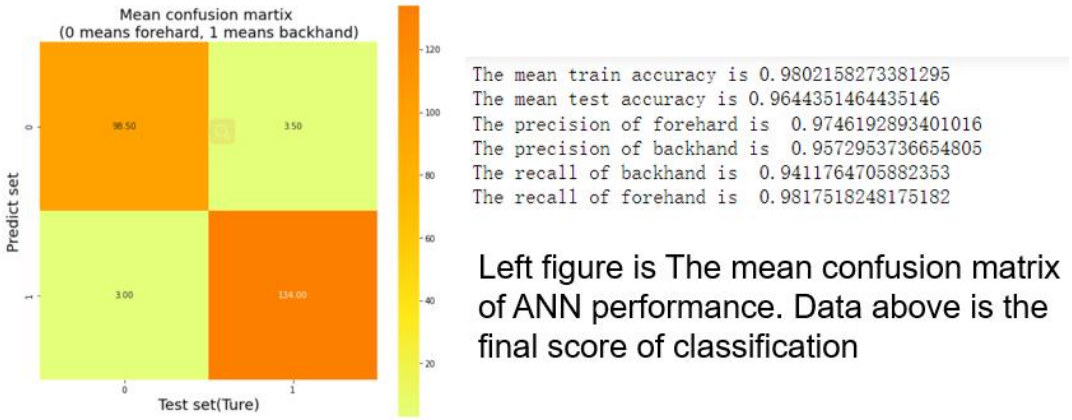

Figure 11

As for Decision Tree algorithm, we visualize the results. Every step of classification is visible. Also, we compute the confusion matrix, accuracy, precision and recall. The results are as follow,

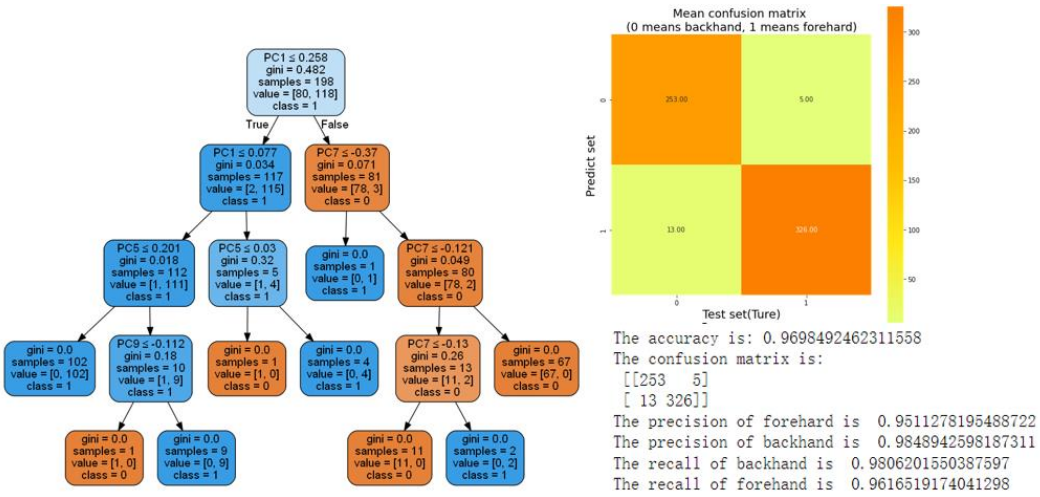

Figure 12: The model structure and final performance of Decision Tree Mode

Table 4: The Comparison of ANN model and DT model

	ANN	DT	Better
Accuracy	0.972322	0.969849	ANN
F1 forehand	0.957605	0.965648	DT
F2 backhand	0.972370	0.973134	Same

The accuracy is an import metric of classifier. Thus, we conclude that ANN performs better than DT. Nevertheless, ANN is vague. It is hard to know the change of every input weight and bias during updating. An ANN model with complex structure is considered as a “black box”. But the DT can do an excellent visualization. You can know clearly every step of classification, such as Gini coefficient, classification condition, the number 2 classes divided by last class. So, we conclude DT is better than ANN in terms of process.

As for the One-Class SVM model, dividing all data into two set: train set (66%) and test set (34%). When the value of nu=0.12, in the figure13 (a), the number of True Negative is 9 and False Positive is 0. All faulty data are detected. But the total 255 normal data, 9 are judged as faulty data. In the judgment of wrong data, the accuracy rate reaches 100%, but in the judgment of normal data, the accuracy rate is 96.47%.

As for LOF model, dividing all data into two set: train set (66%) and test set (34%). When the value of contamination is equal to 0.08, n_neighbors equal to 20, in the figure 13(b), the number of True Negative is 32 and False Positive is 0. All faulty data are detected. But the total 255 normal data, 32 are judged as faulty data. In the judgment of wrong data, the accuracy rate reaches 100%, but in the judgment of normal data, the accuracy rate is 87.45%.

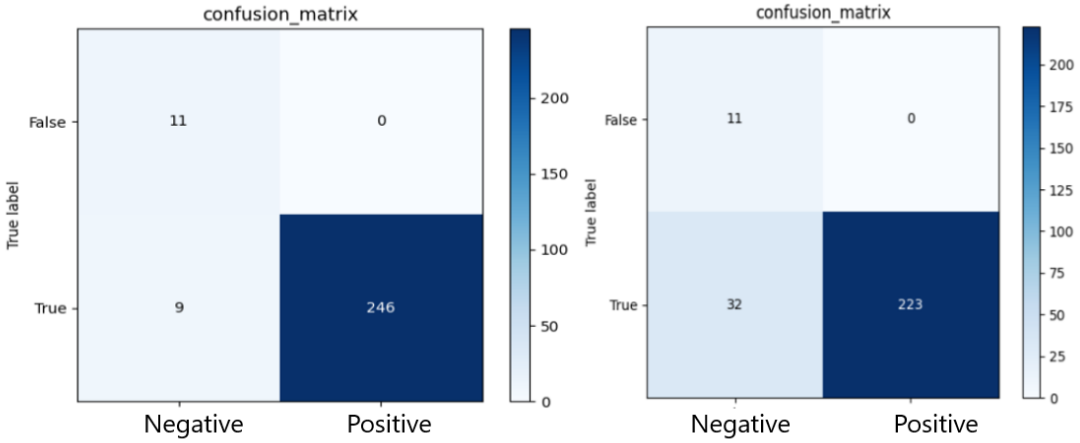

Figure13(a): Result of One-Class SVM Figure13(b): Result of LOF

Figure13: The Confusion Matrix of Experimental Results.

Table 5: The Comparison of One-Class SVM model and LOF model

	Proportion of FP	Proportion of NT	Accuracy
One-Class SVM	0/11=0	9/225=0.0353	0.9647
LOF	0/11=0	32/255=0.1255	0.8745

Compared with these two algorithms, it is not difficult to find that although the models trained by the two algorithms can find out the wrong data, One-Class SVM has fewer errors and is more accurate in judging the normal data.

4. Discussion

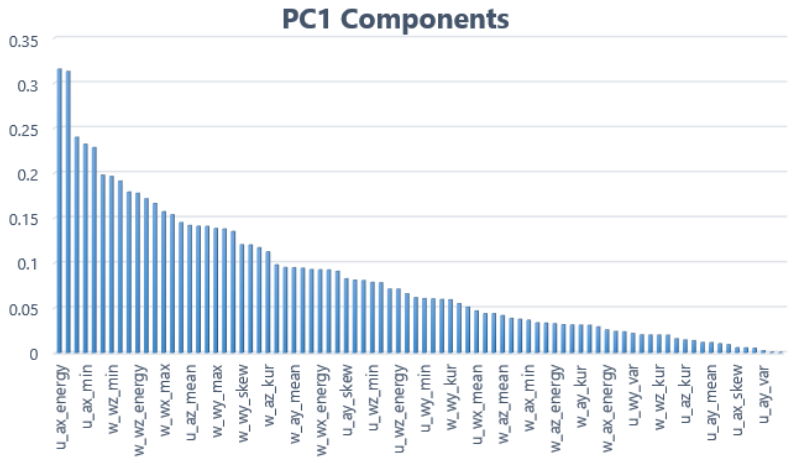

Figure 14: The Absolute Value of Weights of all 84 Statistical Features in Principal Component 1

After principal component analysis, it can be found that in the principal component 1, which gets highest mutual information score among all 84 principal components, the statistics, energy and minimum value of the x-axis linear acceleration of upper arm accounts for the most. It shows that the x-axis linear acceleration of upper arm becomes the most important factor in distinguishing forehand motion and backhand motion, which is quite beyond our expectations. Besides, although the data

segmentation may perform not very well due to the fuzziness in the boundary of each motion, the results of classification and faulty motion detection both perform quite satisfying.

The parameter of hidden layer in ANN determination is based on empirical formula, which always considers the worst condition. So that the range of hidden layer and hidden layer nodes will be wide, which need cost heavy calculated amount to find the appropriate parameter. Besides, it is hard to judge whether the model is overfitting or underfitting for the stander is not clear. Actually, the two algorithms cannot make full use of 11 feature values, for example: the DT just use the PC5, PC9, PC1 and PC7 four feature values to classify.

By adjusting the algorithm parameters, it is easy to find all abnormal data (FP = 0). However, because the existence of TN is difficult to eliminate, there is always a bottleneck in the accuracy of our algorithm. Imagining a situation: when a test set is full of normal data, due to the existence of TN, the data set will still be detected and considered to have a lot of abnormal data. Therefore, further reducing the value of TN will be the key problem. Therefore, we try to prove whether expanding the training set can effectively reduce TN.

Take the average value through multiple experiments, this figure uses the existing data to describe the relationship between the number of training set data and proportion of TN. Through MATLAB, the fitting function is:

$$F(x) = 21.62x^{-0.9942}$$

x is the number of training set data and F(x) is predicted value of proportion of TN.

Table 6: The Prediction Results According to the Fitting Equation

Accuracy	99%	99.90%	99.99%
Proportion of TN	0.01	0.001	0.0001
Training Set Data	2292.95	23337.4	237525.97

5. Conclusions & Future Work

In conclusion, we've gained a lot and progressed a lot. We formulate a real-world problem into data mining task and use what we've learnt in this class to solve this problem. To our surprise, the final result is quite satisfied, not only the classification model but also the faulty motion detection model. Besides, we think what we've done is meaningful and useful for our reality. Table tennis players can record their performance during a game or a training. Through the analysis from our model, he or she can get how many times he or she uses forehand stroke and backhand push, and whether his or her motion is correct in terms of the guidance of novices. We think the information above definitely will improve table tennis player's training quality.

If more time allowed, we will continue to improve our model, not only will we learn and try more efficient and appropriate algorithm, but also consider more complicate situations. All of our team are ping-pong enthusiastic fan, we hope to make contributions to what we love.

6. References

[1] Wu, W.-L.; Liang, J.-M.; Chen, C.-F.; Tsai, K.-L.; Chen, N.-S.; Lin, K.-C.; Huang, I.-J. Creating a Scoring System with an Armband Wearable Device for Table Tennis Forehand Loop Training: Combined Use of the Principal Component Analysis and Artificial Neural Network. *Sensors* 2021, *21*, 3870. https://doi.org/10.3390/s21113870

[2] Wang, Huihui & Li, Lianfu & Chen, Hao & Li, Yi & Qiu, Sen & Gravina, Raffaele. (2019). Motion Recognition for Smart Sports Based on Wearable Inertial Sensors. 10.1007/978-3-030-34833-5_10.

[3] Sharma, M., Anand, A., Srivastava, R., & Kaligounder, L. (2018). Wearable Audio and IMU Based Shot Detection in Racquet Sports. *ArXiv, abs/1805.05456*.

[4] Ba´nkosz, Z.; Winiarski, S. Using Wearable Inertial Sensors to Estimate Kinematic Parameters and Variability in the Table Tennis Topspin Forehand Stroke. *Appl. Bionics Biomech.* 2020, *2020*, 1–10. [CrossRef]

[5] CrossEntropyLoss — PyTorch 1.10.0 documentation

[6] https://zhuanlan.zhihu.com/p/100419971

[7] The Number of Hidden Layers | Heaton Research