The WSN Localization Based Machine Learning Application to Tennis Game

Xiaoyi Zheng, Xinhui Ding, Zian Chen
University of Waterloo
Ontario, Canada
x94zheng ,x55ding ,z677chen@uwaterloo.ca

***Abstract*—**
Sensors, a new class of devices, can detect, process, and communicate important data for users. Therefore, increasing Wireless Sensor Network (WSN) applications, such as target tracking and emergency response, have been developed in recent years. It is worth mentioning that localization algorithms could be used in many applications,such as localization.
There are various algorithms [1] to localize the object by WSN. They mainly divided in two categories,range-based and range-free. For range-based method,there are various methods such as RSSI, TOA, TDOA and etc. The main principle of these methods is firstly obtaining the distances between the object and at least three anchors and then compute the location.
In our paper, we introduce a machine learning based method which could localize the objective and apply it to the tennis games which could localize the drop point of tennis ball and judge the tennis ball is out of bound or not.

I. Introduction

A. Tennis Games

Tennis games are played on a rectangular court. According to the standard, the size of the court is 23.77 meters long. Since the tennis games could be separated into singles and doubles games. For the singles game, the wide is 8.2 meters and the doubles game is 11 meters. For each half of court, there are baseline, singles sidelines, doubles sidelines, server line and center service line. As shown in the figure 1, the distances are marked on the figure.

The players start on opposite sides of the court [2]. Based on the result of tossing the coin, one side of players in the server and another is the receiver. The server could start from the areas surrounded by the baseline, the center service line and the sideline. The legal server should across the net and should not touch the net, diagonally drop into the service box of opponents. The receiver should hit back the ball before the ball dropping twice. The ball must cross the net and drop into the side of the opponents court.

B. Predict localization of tennis ball

The tennis courts have three categories: grass, clay and hard courts. The clay is the easiest material to judge since the drop point could leave a mark on the surface. But the grass and hard courts are relatively much harder to judge because of the high speed and the low bounce of the balls. Human eyes are not precise enough to capture the drop point of the ball and sometime could make it wrong. When the judgements are different between players and referees, the players are allowed to challenge point-ending line calls. Then the hawk-eye technology would be used on decision making.

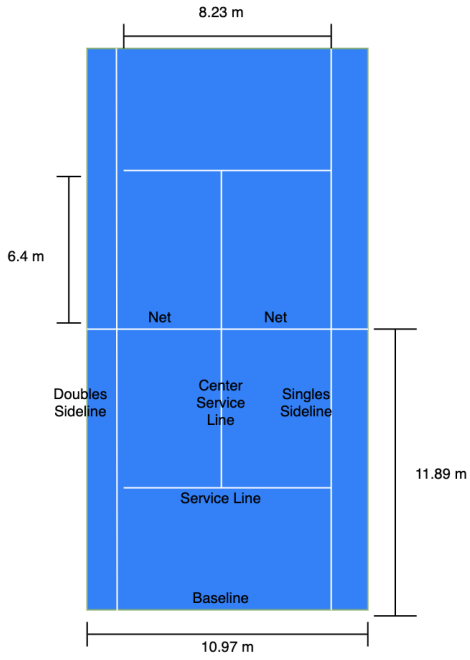

Figure 1: The Schematic Diagram of the Tennis Court

The hawk-eye technology [3] [4] is the system consists of cameras and computers which could visually track the trajectory of the ball and display a record of its statistically most likely path as a moving image. The principle [5] is base on the triangulation that using visual images and timing data provided by the set cameras. Triangulation is the process of finding the location of a point by measuring the angles between the point and other points which are fixed.

Normally, more than 10 cameras [6] which have about 120 MHz frame rate are set around the tennis court to capture

the localization of the tennis, as shown in the figure 2. The signals and data are further sent to the high-speed video processor and update 100 times every second. The video processor would identify each frame from each camera, then transfers information to the computer for it to be turned into a 3D image and for the information to be collected into images such as a grouping corresponding pixels to the image of the ball. Then the computer would predict a ball path from the said 3D ball position as computed in successive frames and map the predicted path on the modeled area so as to identify any interaction with ball and lines.

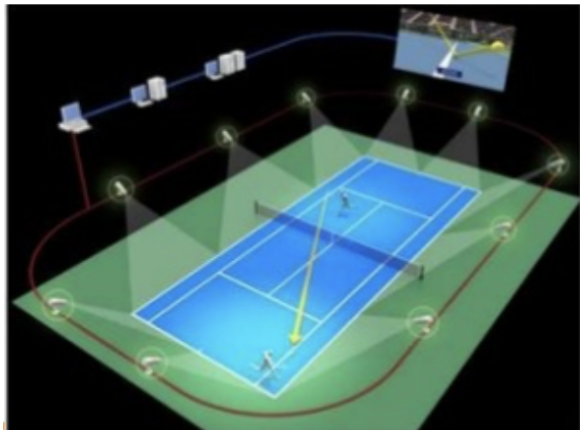

Figure 2: The Schematic Diagram of the Hawk-eye Technology

One of the advantages of hawk-eye technology is the system optimized human error in decision making. The collected data could further help the players to study from past games and improve their skills. However, there are several disadvantages that could not be avoided. Firstly, the lens distortion, the point spread caused by the scattering of the light and the distortion of pixels in digital photos could affect precision. Secondly, since the system should rapidly process the video feeds from the cameras. The quality of the video processor should be extremely high and would cause a high cost. Just imagine that if we would like to set the system in every match and every court.

With the development of sensor technology, the size of the sensor is becoming smaller and smaller. It's easier to set up sensors anywhere you need them. In addition, the cost of sensors is lower than that of professional motion cameras. So,a very tiny sensort could be set up on the tennis balls which could send a signal when the tennis ball drop on the ground which we could consider the tennis ball as a unknown node and then we can use various algorithm to predict the location of its drop point.In term of localization algorithms,they are becoming more and more mature and easy to understand and apply. People can choose different algorithms according to different scenes. Using the data obtained by sensors and machine learning technology, we can process and analyze the data to help the referee judge the game and improve the athletes' sports skills.

II. Realted work

A. Localization Algorithm

In the wireless sensor networ [1], the localization is one of the most important topics, because of the widely used in our daily life. The localization algorithms could be separated into two categories: range-free and range-based. The range-free algorithms are using landmarks to connect the unknown nodes, during this process, the information may need the artificial or the GPS system to require. The principle of range-based algorithms is using distance or angle to estimate the location of the targets. The common algorithms of range-based are: the time of arrival, the time difference of arrival, the angle of arrival and received signal strength indication.

Concerning range-based localization algorithms [7], the time-based algorithms have been widely proposed in the application of different scenarios. The algorithm is using signal propagation time measurement and known propagation velocity to determine the distance between the sensor nodes. The major difference between ToA and TDoA algorithm is the nodes and receivers of ToA should be synchronized. Both the ToA and TDoA algorithms have high accuracy, but the energy-consuming is expensive at the same time. Besides, better performance of hardware is required by the TDoA.

Comparing to ToA and TDoA, the RSSI [8] [9] algorithm could save more energy and cost-effective. The RSSI algorithm relies on the transmitted signal power decays with distance. There is a path loss called free-space propagation loss related to the distance when the nodes sent a signal to the medium. The landmark could calculate the distance between the unknown node and itself by using this relationship. The most typical model between the attenuation of radiofrequency signal and distance in the air is the free space propagation model of radiofrequency signal, based on the Friis transmission equation we could have:

$$\frac{P_r}{P_t} = \frac{A_t A_r}{d^2 \lambda^2} \tag{1}$$

where the P_r represents the power at the receiving antenna, P_t means the output power of transmitting antenna, λ is the wavelength and d is the distance between the antennas.

The gain of the transmitting antenna could be written as:

$$G_t = \frac{4\pi A_t}{\lambda^2} \tag{2}$$

Then the gain of the receiving antenna could be written as:

$$G_r = \frac{4\pi A_r}{\lambda^2} \tag{3}$$

Since the Effective Isotropic Radiated Power of RF tag is:

$$P_{EIRP} = P_t G_t \tag{4}$$

Therefore, the attenuation of the radiofrequency signal and distance in free space propagation model could be expressed as:

$$P_r = P_{EIRP} G_r (\frac{\lambda}{4\pi d})^2 \tag{5}$$

The term $(\frac{\lambda}{4\pi d})^2$ is called free space path loss.
In dB, the expression of (5) becomes:

$$P_r(dBm) = P_t(dBm) + G_t(dB) + G_r(dB) - L_p(dB) \tag{6}$$

To calculate free-space loss by using the above relation and assuming $G_t = G_r = 1$:

$$L_p(dB) = 20log_{10}(\frac{4\pi d}{\lambda}) \tag{7}$$

The terms $G_t(dB)$ and $G_r(dB)$ are clearly gains, when they are positive, the received power increases. And as distance increases, $L_p(dB)$ increases, which because of the negative sign, reduces the received power.

B. Machine Learning Algorithms

For traditional RSSI algorithm, it use the free-space path loss formula to compute the distance between the objective and the known nodes. Then use trilateration to compute the location of the objective which is a range-based method. For machine learning algorithms, we use machine learning method to predict the location of the object which replace the trilateration. Therefore, our input only need the receivied power and feed to the model.Then the model will predict the location of the objective. So our method is a kind of range-free method.

1) RNN based: The neural network could be divided into the biological neural networks and artificial neural networks, the latter is the mathematical model that we need to learn to simulate animal neural networks. A common problem in modern data analysis is neural network, which can be classified as a semi-parametric method. There are 27 kinds of neural network models, such as perceptron, Markov chain, artificial neural network, cyclic neural network and so on.

Recurrent neural network (RNN) [10] is a kind of recurrent neural network that takes sequence data as input, recursion in the evolution direction of sequence and all nodes (cyclic units) are linked in a chain. The traditional neural network only establishes the weight connection between layers. Compared with the traditional neural network, RNN is a special neural network structure. It is proposed according to the view that "people's cognition is based on past experience and memory". It not only considers the input of the previous moment but also creates a "memory" function of the neural network for all the previous contents. The main structure of the RNN neural network model is shown in Figure 3.

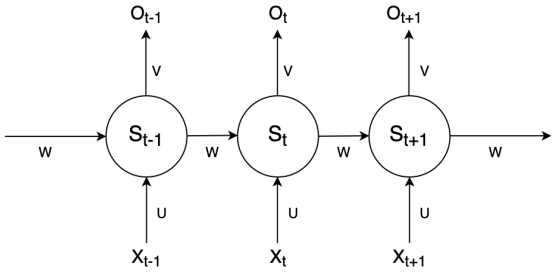

Figure 3: The Structure of RNN Model

In the hidden layer of the RNN network, there is an arrow to indicate the circular update of the data and the reverse of the hidden layer. Feedback not only flows into the output, but also flows into the hidden layer of the next time step, and then affects the weight values of the next time step. This is the method to realize the time memory function. The layer by layer expansion diagram of the neural network with multiple time steps is shown in figure.

As shown in the figure, $t-1$, t, $t+1$ denotes time series, x represents input samples, S_t is the memory unit of the sample at time T. for the time of t, when the sample is input:

$$S_t = f(WS_{t-1} + Ux_t) \tag{8}$$

where W is the weight of the input, u is the weight of the input sample at the moment, and f is the activation function. O_t is the output at the time of t:

$$O_t = g(VS_t) \tag{9}$$

Where V is the sample weight of the output and g is the activation function.

The training method of the RNN neural network model is back-propagation through time (BPTT), which continuously searches for better points along the negative gradient direction of parameters to be optimized until convergence. Because of the RNN processes time series, it needs time backpropagation. The weight parameters need to be updated in RNN are W, U and V. If the error value of each output O_t is e_t, then the total error can be expressed as: $E = \sum_t e_t$.

The cross-entropy loss function or square error loss function could be used as the loss function. Since the output of each step not only depends on the network of the current step, but also needs the network state of the previous steps, the error value of the output end should be transmitted backward, and the gradient descent method is used to update:

$$\Delta U = \frac{\partial E}{\partial U} = \sum_t \frac{\partial e_t}{\partial U} \tag{10}$$

$$\Delta V = \frac{\partial E}{\partial V} = \sum_t \frac{\partial e_t}{\partial V} \tag{11}$$

$$\Delta W = \frac{\partial E}{\partial W} = \sum_t \frac{\partial e_t}{\partial W} \tag{12}$$

The advantage of the RNN neural network is that it could effectively deal with nonlinear time series, which makes the previous time-series directly affect the data of the next time point, and the hidden layer of RNN could realize the effect of self-loop recursive feedback. However, RNN is unstable, and gradient vanishing and gradient explosion may occur at any time. The long-term and long-term memory model (LSTM) proposed by Schmiduber et al. (1997) solved the above problems of recurrent neural networks.

As mentioned above, the long and short term memory neural network (LSTM) [11] [12] is an improvement of RNN. Each layer of LSTM is designed with multiple gate structure neurons, which could memorize any time state of the hidden layer. No matter how long the propagation path of the gradient is, it will not completely disappear or drop to zero.

The LSTM model has three gates to control each neural unit: an input gate determines when to allow the activation state to pass to the memory unit, an output gate determines whether the activation state leaves the memory unit, and a forgetting gate determines whether the state of the last neuron is completely or partially memorized or completely forgotten. When the gradient decreases, these gates can be used to modify the parameters of the error function of selective memory feedback. The network structure is shown in figure 4. Each black node is connected with an activation function. The middle node is used to store the internal state of the memory unit. The number of 1 is used as the weight to cross the time step and then feedback to itself. The LSTM network updates the memory unit by recursive equation and activates the mapping from input x to output y:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1} + b_i) \tag{13}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1} + b_f) \tag{14}$$

$$c_t = f_t \Theta C_{t-1} + i_t \Theta g(W_c x_t + U_c h_{t-1} + b_c) \tag{15}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o c_t + b_o) \tag{16}$$

where σ is the $sigmod$ activation function; g is the tanh activation function; i, f, c, o are the activation vectors of input gate, forgetting gate, cell state and output gate respectively; W, U, V are the weight matrix; b is the partial value matrix; Θ is the element by element product of vectors. The training method of LSTM neural network model also adopts the back propagation algorithm, which is the same as the back-propagation algorithm of RNN. It also updates all parameters by gradient descent method. The difference is that there are two hidden states $h(t)$ and $C(t)$ in LSTM model. Therefore, it is necessary to use gradient descent method for back propagation to update weight parameters.

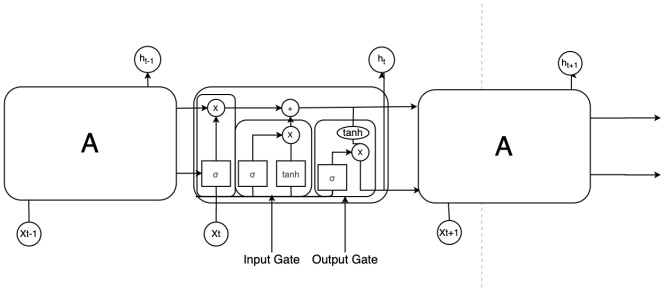

Figure 4: The Structure of LSTM Model

2) MLP based: Another algorithm adopted is Multiple Layers Perceptron (MLP). MLP [13] is the fully connected neural network, including the input layer, the output layer and multiple hidden layers. As shown in the figure 5, the layers of the multilayer perceptron are fully connected with each other, and any neuron in the upper layer is connected with all neurons in the lower layer. The hidden layers could be used to solve the non-linear problem. The most simple MLP model contains only one hidden layer, forming a three-layer structure.

The input layer decided by the number of the dimensions for the vectors. If you have n-dimensional vectors then you have n neurons. Since the neurons in the hidden layer are fully connected to the input layer, assuming that the input layer is represented by the vector x, then the output of the hidden layer would be $f(w_1 x + b_1)$, where w_1 is the weight, and the b represents the bias, and the function f could be the commonly used Sigmoid function or tanh function.

About the output layer, from the hidden layer to the output layer can be seen as a multi-category logistic regression, so the output of the output layer is $softmax(w_2 X_1 + b_2)$, where the X_1 represents the output $f(w_1 X + b_1)$ of the hidden layer.

3) SVM based: SVM is a supervised machine learning method,the main principle of it is that for linear data, it will find a hyper-plane to separate different classes and maximum the geometric spacing of each. It could also efficiently perform a non-linear classification by using kernel which will implicitly inputs into high-dimensional feature spaces in order to make the data linear.In addition, compared with RNN or MLP, it is more lightweight than those neuron networks so it could easily be deployed on a limited-source equipment,such as wireless sensors.

III. METHODOLOGY

A. Setting Scenario

The goal of this paper is to find the location of the tennis ball. Thus, the first step is to set the sensors. According to the

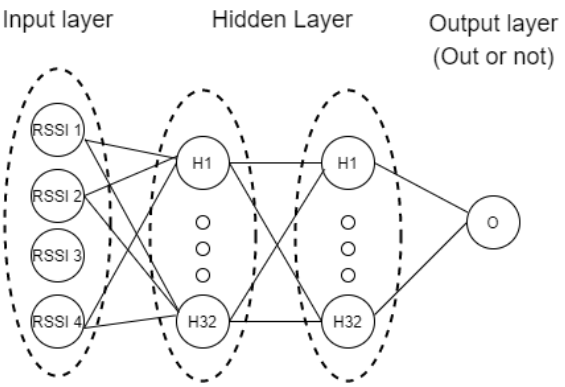

Figure 5: The Structure of MLP Model

symmetric, except the area of the net, the other parts of the tennis court could be treated as the same situation. In order to prevent the settled sensors from affecting the athletes, our plan is to set up sensors around the tennis court so that the transmission range of sensor signals can surround the whole court. As shown in the figure 7, take the 2 meters left of the intersection point of the bottom line and the left doubles sideline as the coordinate origin, and establish the coordinate system. In addition, we considered that place a tiny chip in tennis which could send a signal when the ball drop on the grand.

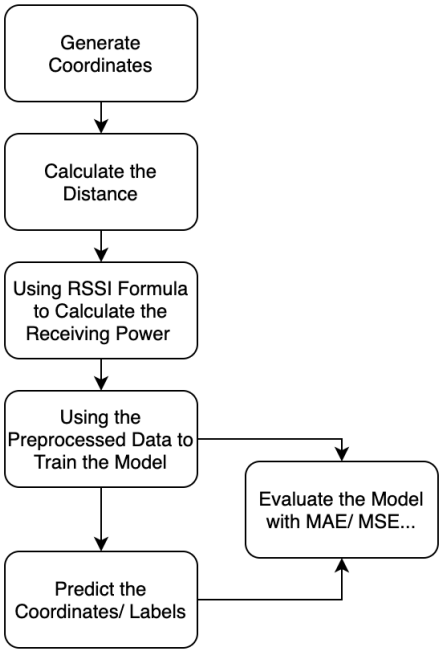

Figure 6: The Flow Chart of Simulation

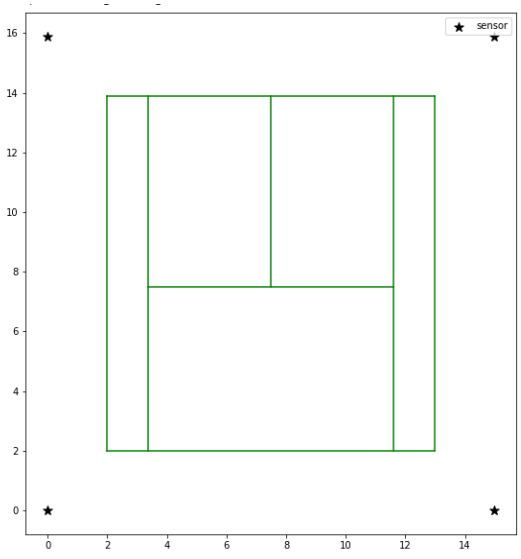

Figure 7: The Schematic Diagram of Setting Sensors

B. Choosing Localization algorithm

We further assume the relationship between a tennis ball and the settled sensors are a relatively static state. Then the method for localization could be narrow down into static landmarks with static nodes. The category of static landmarks with static nodes including range-free and range-based. As mentioned above, the range-based approaches are more proper for our scenario. Considering that the service speed of the tennis ball can reach more than 200km/h, we believe that the time-based measurement method cannot capture such a fast ball. In addition, in wireless sensor network, it is very difficult to synchronize the time of sensors, which increases the error and reduces the accuracy to some extent. Therefore, from the ToA, TDoA, AoA and RSSI methods, we chose the RSSI as our localization algorithms based on the requirements of precision and accuracy.

C. LSTM based Model

The flow chart of the simulation is shown in the figure 6. The first step is to create the dataset. Because the LSTM model has the memory of the previous information, based on the different techniques of each player, in order not to increase redundant information, we just use one match as a unit. On the professional circuit, men play best-of-five-set matches at the Grand Slam tournaments, we assuming that for one match: each set has 12 games, each game has 8 points (just one deuce included), each point has one batting with one double fault serve, and we could have our upper bound of the positions: overall 1440 drop points of balls. With the same principle, except the men's singles of the Grand Slam tournaments, others are best-of-three-set

matches. Suppose for one match: without any double fault for service as well, each set has 6 games, each game has 4 points, we will have 216 drop points of balls. Therefore, the upper bound for our dataset is 1440 and lower bound is 216. Considering that the number of Masters tournaments are much greater than the Grand Slam tournaments, we first used 500 as our dataset size.

We randomly generate 500 two-dimensional coordinates in the established Cartesian coordinate system. These coordinates are used for labels for training the model. And the coordinates are further used with the Euclidean distance formula to calculate the distance between the ball to the sensors.

$$D = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} \tag{17}$$

As mentioned above, the RSSI could be calculated by the equation (6), which is the sum of the transmitting power, the gain of transmitting, the gain of receiving and free-space transmission loss. The transmitting power is fixed value which we assume is 10 dBm. If we chose the fixed double plate directional antenna the range of transmitting could cover at least half of the tennis court. The working frequency of such antenna is from the 2400 MHz to 2525 MHz and the gain of receiving normally is 14 dB. The free-space transmission loss could be calculated by the equation (7). The formula (7) is a generic form of free-space transmission loss in the metric system of units. Since in actual links, the frequency is in MHz or GHz and distance in km, by using $c = 3 \times 10^8$ m/s, then free-space transmission loss is specified by one of the following formula:

$$L_p = 32.4 + 20logf + 20logd \tag{18}$$

And if the sensors are working in the 25 centigrade and 1 atmosphere, we could think the transmission power is 2400 MHz. If the sensors are working in the 25 centigrade and 1 atmosphere, we could think the transmission power is 2400 MHz. Therefore, by using the formula (18), we could transfer the distances of dataset into receiving power which is the input of the model. And by setting the random state as 42, 80% data are taken out as train set.

Before using the created data to feed our model, the data need to preprocess. The method we chose is standardization. As shown in equation (19), the standardization is to scale proportionally and falls into a small interval. To be more specific, standardizing a vector means subtracting a measure of location and dividing it by a measure of scale. After standardization, all attributes of data will have the same weight.

$$x^* = \frac{x - \mu}{\sigma} \tag{19}$$

The structure of the model [10] is shown in the figure 8. It consists of two LSTM layers and a fully-connected layer. The first LSTM layer fed the data of the training set and returns all hidden states obtained according to the input to the next LSTM layer. Since the required output is the two-dimensional coordinates of the placement, the number of neurons in the fully-connected layer is set to 2.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 1, 24)             2784
_________________________________________________________________
lstm_1 (LSTM)                (None, 24)                4704
_________________________________________________________________
dense (Dense)                (None, 2)                 50
=================================================================
Total params: 7,538
Trainable params: 7,538
Non-trainable params: 0
_________________________________________________________________
```

Figure 8: The LSTM Model

During the compilation of the model, in order to make the model converge, we set the number of epochs as 2000. When the entire dataset passes through the neural network once and returns once, this process is called an epoch. If the number of epochs is too small, the model will be underfitting, which will not reach the expected accuracy. If too many epochs were set, the model would overfit and would not perform well in the test set.

The batch size as 10. If the batch size is too small, the training data will be difficult to converge, resulting in underfitting. Since we have a total of 500 samples and the batch size is 10, 50 iterations are needed to train the complete sample set.

For the training set, we extract 20% as the validation set and set validation frequency to 1. The validation set is used to test the state and convergence of the model in the training process. At the same time, the validation set can also be used to monitor whether the model has been overfitted. Generally speaking, after the verification set is stable, if the training continues, the performance of the training set will continue to rise, but the validation set will not rise but fall, so the overfitting generally occurs. So the validation set is also used to determine when to stop training.

The initial learning rate was set to 0.001, and when the model stopped being promoted, the ReduceLROnPlateau function was set as a monitor. The patience is set as 3 which means if 3 epochs with no improvement after which the learning rate will be reduced. The learning rate will be reduced following the equation:

$$lr_{new} = lr \times factor \tag{20}$$

Where the factor is 3. And the minimum learning rate is set as 0.00001 in case of the learning rate is too small.

In addition, we used Adam [14] as the optimizer to adjust the weights of the model.The Adam algorithm considers the

first-moment estimation (the mean value of the gradient) and the second-moment estimation (the non-centralized variance of the gradient) to calculate the update step size.

D. MLP based model

In the last section, we used RSSI as input to predict the tennis ball's placement point. With these coordinates, we can help athletes better analyze their own or opponent's technical characteristics, so as to improve their skills or change strategies. However, in the tennis match, we are more concerned about whether tennis is out of bounds, so we can use the MLP model to simplify the problem to the classification problems.

The processing of the data is basically the same as in the previous section: 500 coordinates are generated randomly, the distance between the coordinates and the sensor is calculated, and then the received power is calculated according to the RSSI formula. However, the labels of the dataset need to judge the distance between the coordinate and the sensor according to the size of the tennis court. If the coordinate out of the boundary, that is, the coordinates beyond the range of size of the tennis court are marked as 0. Also, mark the coordinates that are not out of bounds as 1. Establish the label of the data set in the above method.

The MLP model structure is shown in the figure 9. The model is composed of three fully-connected layers and two dropout layers. The number of neurons in the first two fully-connected layers is 32. In the last fully-connected layer, only one neuron is needed because only one value is output for the classification problem. The purpose of the dropout layer is to discard some neural network units from the network according to a certain probability during the training process, which is equivalent to finding a more concise network from the original network. Our ratio is set at 0.2, which means that 0.2 of neurons will be discarded from the MLP network through the dropout layer. Setting the dropout layer can prevent the model from overfitting and save training time.

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_12 (Dense)             (10, 32)                  160
_________________________________________________________________
dropout_6 (Dropout)          (10, 32)                  0
_________________________________________________________________
dense_13 (Dense)             (10, 32)                  1056
_________________________________________________________________
dropout_7 (Dropout)          (10, 32)                  0
_________________________________________________________________
dense_14 (Dense)             (10, 1)                   33
=================================================================
Total params: 1,249
Trainable params: 1,249
Non-trainable params: 0
_________________________________________________________________
```

Figure 9: The MLP Model

The number of the epoch is set to 800. We tried to adjust this number, but the accuracy of the 800 epoch is the highest. Like the LSTM model, the batch size is set to 10, 20% of the training set is used as the validation set, the validation frequency is 1, and the ReduceLROnPlateau function is set to monitor the training. Unlike the LSTM model, the loss function used to validate the model is binary_crossentropy instead of the original MAE. The expression of the binary_crossentropy function is:

$$loss = -\sum_{i=1}^{n} y_i log\hat{y}_i + (1 - y_i) log(1 - \hat{y}_i) \tag{21}$$

The binary_crossentropy is for the loss function between probabilities, and the loss is zero only if y_i is equal to $\hat{y}_i$, otherwise, the loss is a positive number. Moreover, the greater the difference in probability, the greater the loss.

E. SVM-based model

As mentioned in last section, we could use SVM, a non neuron network model, to classify the data. For SVM there are some parameters we have to set. One of the most import parameter is the C value also named regularization parameter. The larger the C is, the smaller the margin is, conversely, if the C-value is small, the margin will be big, which means there will be more data-points could be misclassified by the model. To find a best C, first we set a random list and to observe the tendency of the curve. Then we change the range of the list which will near the best parameter obtained by last trying, Therefore, from figure 10 and 11, we got 18 as the best value for C. Besides parameter C, we use RBF as our kernel function which map our data to high dimension which let them be linearly separable and use scale as the gamma of the kernel function.

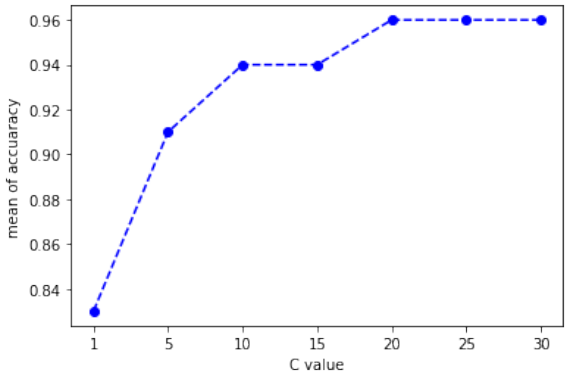

Figure 10: The Mean of Accuracy with Different C Values

IV. Evaluation

Now, we investigate the performance of our various method. As mentioned before, the RNN method mainly used to predict the location of tennis balls whereas MLP method and SVM mainly used to classify the tennis whether be out of the bound or not. So we use different evaluation metric on different methods.

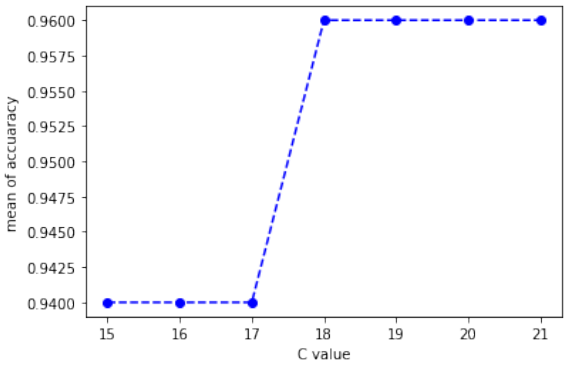

Figure 11: The Mean of Accuracy with Different C Values

A. *Prediction of location*

For RNN method, because it predict the location of the tennis balls so we select two evaluation indicators: mean square error (MSE) and mean absolute error (MAE). The formula of mean square error is as follows:

$$\frac{1}{m}\sum_{i=1}^{m}(y_{test}^{(i)} - \hat{y}_{test}^{(i)})^2 \tag{22}$$

we can get the result as shown in the figure 12below. From the figure, we can see that the value of MSE changes with the increase in the number of epochs. Within 100 epochs, the value of MSE drops sharply, and up to 250 epochs, there is basically no fluctuation, which means that the model has been fitted.

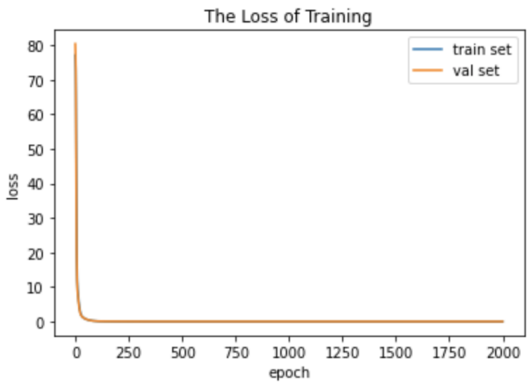

Figure 12: The MSE of training

Another evaluation standard is MAE. Its principle is: to minimize the distance between the true value and the predicted result, we can directly subtract the absolute value, add m times and divide by M to find the average distance. The expression is as follows:

$$\frac{1}{m}\sum_{i=1}^{m}|y_{test}^{(i)} - \hat{y}_{test}^{(i)}| \tag{23}$$

Figure **??** shows the result. The figure shows that the value of MAE decreases with the increase in the number of epochs. The curve changes are basically similar to the MSE curve, which proves the previous conclusion that the model has been fitted.

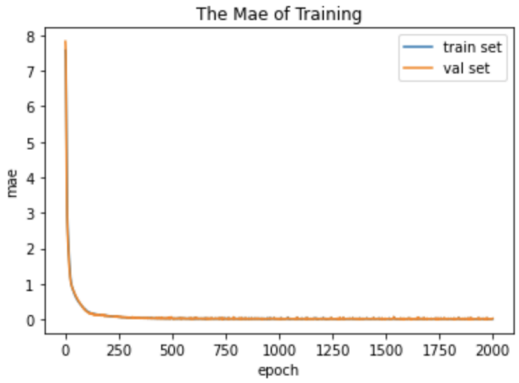

Figure 13: The MAE of training

We use the trained model and the input value of the test set to predict. Figure 14 shows the predicted results of the test set and the actual label value. From the figure, we can see that each predicted value and the actual value coincide, and the centroid of the predicted value basically coincides with the actual value. When we use MSE and Mae to evaluate the performance of model on test set, the MSE value is 2.1721e-04, MAE value is 0.0119.

B. *Classification of legal or not*

We use MLP method and SVM to classify the tennis balls whether it is out of bound or not.

We print out the history of the loss value for the training model and make a plot, we can get the result as shown in the following figure. As shown in the figure 15, with the increase in the number of epochs, the loss gradually decreases. After 800 epoch, the loss of the validation set is only 0.0454.

Use the same method to print the accuracy graph, as shown in the figure 16. As the number of epoch increases, the accuracy gradually increases. Finally, through 800 epoch the accuracy of the validation set can reach 98.75%.

For SVM,we use the parameter which mentioned in the last section and then use test set to predict the label,the final accuracy is 96%.

We further applied the values of the test set to the trained model. By using the evaluate function, the loss of the model on the test set is 0.2388, and the accuracy could reach 97%.

To better evaluate our MLP model, we introduced the confusion matrix, F1-Score, and the ROC curve. The predicted result and the actual value can be composed of four situations:

True Positive (TP): the actual value is 0 and the predicted value is 0;

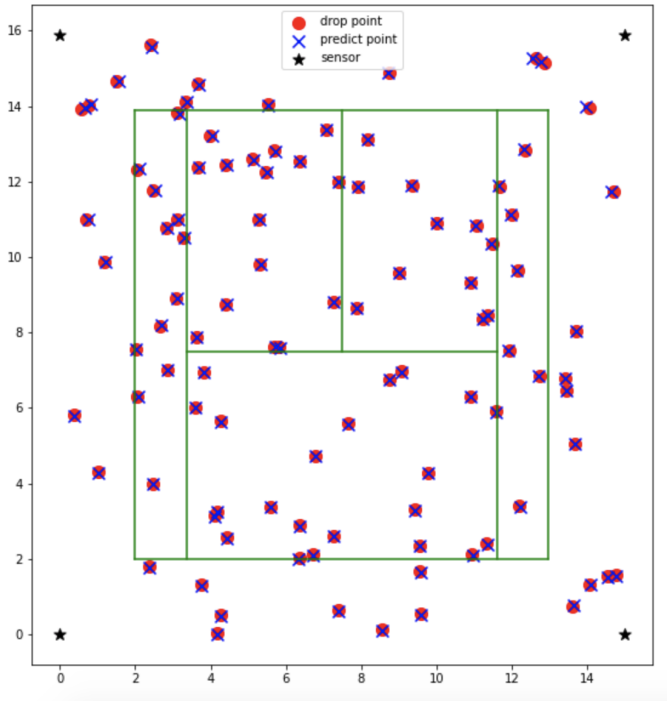

Figure 14: The Comparison of Prediction Values and Labels

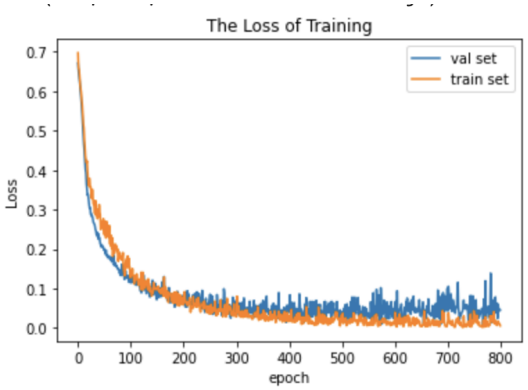

Figure 15: The Loss of Training

False Negative (FN): The actual value is 0, the predicted value is 1;

False Positive (FP): the actual value is 1 and the predicted value is 0;

True Negative (TN): The actual value is 1 and the predicted value is 1. The confusion matrix is a situation analysis table that summarizes the prediction results of the classification model. In the form of a matrix, by using the TP, FN, FP and TN to record and summarize the dataset according to the two criteria of real classification and classification judgment made by the classification model. The result is shown figure 17 and 18:

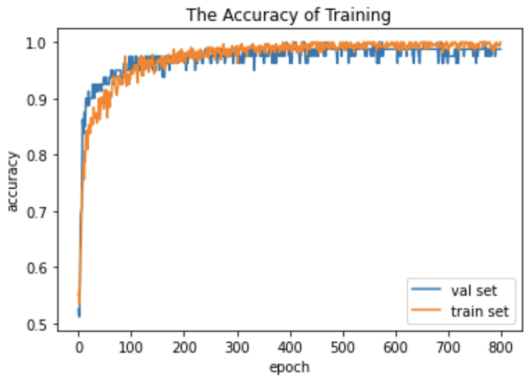

Figure 16: The Accuracy of Training

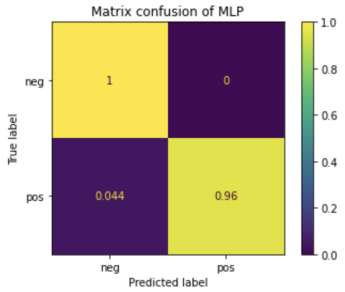

Figure 17: The Confusion Matrix of MLP

As described above, by using the definition of TP, FN, FP and TN, the $Precison$ could be expressed as:

$$Precison = \frac{TP}{TP + FP} \tag{24}$$

the $Recall$ could be written as:

$$Recall = \frac{TP}{TP + FN} \tag{25}$$

From the above formulas, we could have:

$$F1 - Score = \frac{2 \cdot Precison \cdot Recall}{Precison + Recall} \tag{26}$$

For a certain classification, F1 score combines precision and recall as the judgment criteria. From the formula, it can be seen that the larger the F1 score value is, the stronger the classification ability of the model is. As shown in the following figure 19 and 20, the F1-Score of the MLP model and SVM was calculated. For class 0, the F1-Score is 0.96 and class 1 is 0.98. Both values are close to 1 which means the built MLP model has a strong ability of classification.

The ROC curve represents the corresponding changes in the proportion of True Positive Rate (TPR) and False Positive Rate (FPR) of the model for the correct classification of real Positive samples and the proportion of False Positive Rate (FPR) for the negative samples when the selected classification threshold is gradually increased, with the horizontal axis representing FPR and the vertical

	precision	recall	f1-score	support
0	0.89	1.00	0.94	32
1	1.00	0.94	0.97	68
accuracy			0.96	100
macro avg	0.94	0.97	0.96	100
weighted avg	0.96	0.96	0.96	100

Figure 20: The F-1 Score of SVM

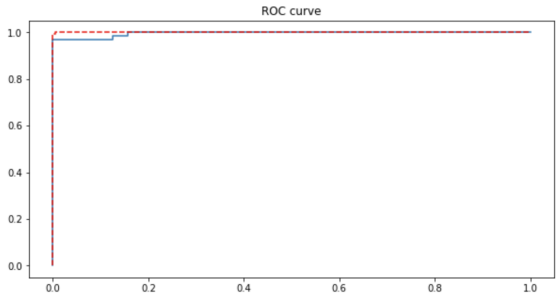

Figure 21: The Roc Curve of MLP

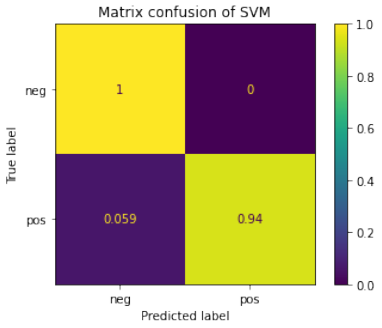

Figure 18: The Confusion Matrix of SVM

	precision	recall	f1-score	support
0	0.91	1.00	0.96	32
1	1.00	0.96	0.98	68
accuracy			0.97	100
macro avg	0.96	0.98	0.97	100
weighted avg	0.97	0.97	0.97	100

Figure 19: The F-1 Score of MLP

axis representing TPR. Generally speaking, the closer the ROC curve of the model is to the upper left, the better the performance of the model. As shown in figure 21 and 22, the curve of both the MLP model and SVM is close to the upper left of entire plot.

V. Conclusion

A. Performance

In this paper, we design two kind of models which could apply to the tennis game. With the help the wireless sensor network, one could predict the location of the drop point of the tennis by the recieved pover and the other one could judge the tennis ball whether is out of bound or not. The accuracy of both models are relatively high. For the former one, we implement LSTM model, a RNN variant, and the loss of prediction could be less than 0.05m. For the latter one, we implement MLP model and SVM model, the accuracy is 97% and 96% respectively. Compared with MLP model, SVM is more lightweight but has similar accuracy, so if the sensor has very limited computing resources, SVM is a better choice.

B. Future work

The accuracy obtained in this paper are all derived from simulations. Due to the limitations of the equipment and some other reasons, we do not carry out the piratical experiments. In the future, if the condition is permitted, we may carry out experiment to verify the feasibility of the method since the accuracy of WSN depend on the real environment.

References

[1] G. Han, H. Xu, T. Q. Duong, J. Jiang, and T. Hara, "Localization algorithms of wireless sensor networks: a survey," *Telecommunication Systems*, vol. 52, no. 4, pp. 2419–2436, 2013.

[2] Wikipedia contributors, "Tennis — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Tennis&oldid=973071992, 2020, [Online; accessed 16-August-2020].

[3] Y. Baodong, "Hawkeye technology using tennis match," *Computer Modelling and New Technologies*, vol. 18, pp. 400–402, 2014.

[4] S. Gangal and S. Raje, "The hawkeye technology," *Computer Science and Engineering (CSE) Department, Indian Institute of Technology, Bombay*, 2007.

[5] C. Chao, "Tennis hawkeye system based on labview platform," *Modern Electronics Technique*, no. 3, p. 29, 2016.

[6] Q. Lai and Z. Yu-ling, "Study on the effects of" hawk-eye" on development of tennis [j]," *Journal of Guangzhou Sport University*, vol. 3, 2007.

[7] R. Kaune, "Accuracy studies for tdoa and toa localization," in *2012 15th International Conference on Information Fusion*. IEEE, 2012, pp. 408–415.

[8] M. Laaraiedh, L. Yu, S. Avrillon, and B. Uguen, "Comparison of hybrid localization schemes using rssi, toa, and tdoa," in *17th European Wireless 2011-Sustainable Wireless Technologies*. VDE, 2011, pp. 1–5.

[9] H. P. Mistry and N. H. Mistry, "Rssi based localization scheme in wireless sensor networks: a survey," in *2015 Fifth International Conference on Advanced Computing & Communication Technologies*. IEEE, 2015, pp. 647–652.

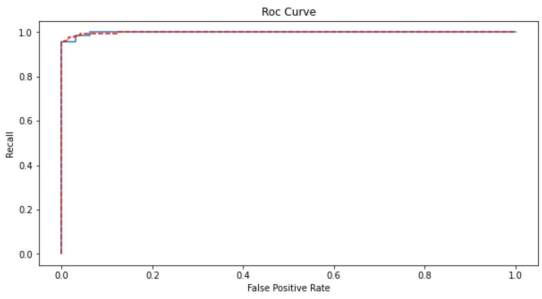

Figure 22: The Roc Curve of ROC

[10] K. M. Tarwani and S. Edem, "Survey on recurrent neural network in natural language processing," *Int. J. Eng. Trends Technol*, vol. 48, pp. 301–304, 2017.

[11] K. Smagulova and A. P. James, "A survey on lstm memristive neural network architectures and applications," *The European Physical Journal Special Topics*, vol. 228, no. 10, pp. 2313–2324, 2019.

[12] B. B. Sahoo, R. Jha, A. Singh, and D. Kumar, "Long short-term memory (lstm) recurrent neural network for low-flow hydrological time series forecasting," *Acta Geophysica*, vol. 67, no. 5, pp. 1471–1481, 2019.

[13] H. Taud and J. Mas, "Multilayer perceptron (mlp)," in *Geomatic Approaches for Modeling Land Change Scenarios*. Springer, 2018, pp. 451–455.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.