Coach AI Report

Team 5

109705001 陳以瑄、109705011 陳盈均、109705003 吳振豪、110700022 陳沂亨

1. Intoduction

The competition of CoachAI Badminton Challenge in 2023 focuses on stroke forecasting in badminton, which is a popular sport in Taiwan. We are provided with comprehensive data containing information of the past stroke sequences. The objective is to predict future strokes' shot types and locations based on the past 4 strokes.

The evaluation metric for the competition is by getting the minimum value of 6 consecutive averages of cross-entropy and mean absolute error.

To tackle the challenge, we modify the given ShuttleNet model provided by the hosts, including connecting necessary functions, preprocessing the data, and incorporating new features. The modified model is then trained using the training data, and the correctness of the predictions is assessed using the testing data. We also explore the patterns and characteristics of the data to enhance the performance of our model.

2. Literature Review / Related Work

Sports analytics involves the utilization of advanced data mining techniques to analyze and interpret collected data, with the aim of predicting and assessing player performance, as well as identifying their strengths and weaknesses in the game[1]. Recent studies have witnessed the integration of various methodologies in the realm of artificial intelligence (AI) research, including Artificial Neural Network, Bayesian Network, Support Vector Machine, and so on[2]. For instance, Tümer et al. [3] introduced a pioneering approach by leveraging artificial neural networks (ANNs) to predict team rankings in the domain of volleyball. This study marks the first known instance of applying ANN technology specifically for team ranking prediction in the field of volleyball. Joel et al. [4] have put forward an innovative player ranking system that integrates the value of completed passes using a support vector machine (SVM) methodology.

In the domain of badminton sport, Ghosh et al. [5] have introduced a data-driven approach for assessing a player's performance by analyzing their stance or posture. This innovative methodology enables a comparative analysis of the stances employed by players of varying skill levels. Cuiping [6] proposed an enhanced CAMSHIFT algorithm aimed at achieving accurate human posture estimation. This novel algorithm presents advancements in the field of posture estimation, offering improved performance and accuracy compared to existing methods. Sharma et al. [7] conducted a comparative analysis of various machine learning classifiers for the prediction of badminton match outcomes. This study contributes to the field by assessing the performance and effectiveness of different classifier models in the context of predicting match results in badminton.

In contrast, our study addresses the field of forecasting badminton movement, which constitutes a significant area within sports analytics. Existing forecasting methods in the domain of badminton encompass diverse aspects, including trajectory prediction [8][9] and shuttle landing point estimation [10]. Our proposed method places emphasis on the prediction of badminton strokes by integrating both shot type classification and landing point estimation. Adopting advanced sequence prediction methodologies such as the sequence-to-sequence model [11] and Transformer network [12], our approach utilizes an encoder-decoder architecture to extract relevant player information and generate future stroke predictions. By leveraging these state-of-the-art techniques, we aim to enhance the accuracy and effectiveness of stroke forecasting in badminton. This allows for the anticipation of upcoming player movements and stroke patterns, enabling improved strategic planning and decision-making during gameplay. Prior research in the field includes methods such as prediction with incomplete action executions [13] and trajectory prediction of individual players [14]. However, these studies primarily concentrate on predicting the same target within a sequence, rendering them unsuitable for direct application to turn-based sequences, as encountered in badminton stroke forecasting.

3. Dataset

Preprocessing

1. We use the coordinate of the player and opponent to calculate the distance. We analyze three kinds of distance including the Euclidean distance, the x-axis distance (for their corresponding standing position is straight or slanting), and the y-axis distance(measurement of far or near), and classify these features based on some domain knowledge of badminton.
2. We use the current time minus the last time to capture the time difference of the dataset in order to capture the relation of time.
3. We calculate the player and opponent's position to calculate the angle of the player against opponent because we could see that the angle of the player and the opponent is significantly related to the shot type he or she is going to use.
4. We compute the shot preference of each player by the distribution of shot type - `type` for players.
5. By subtracting roundscore_A from roundscore_B, we get the difference in scores between the two players.

Characteristics of Dataset

1. The data is sequential. The data is sorted by the time of each match, which consists of several rallies, and each rally in a match contains sequential ball rounds.
2. There is spatial information (coordinates x, y of the location, area) of the strokes and the players. It allows us to know the position of the player and also gets implied information like `distance` from that data.
3. The dataset has various types of features – numerical and categorical data. We deal with them with different methods.
4. After the game is over(somebody wins), the time will not turn to 0, it will jump to other times to start a new game.
5. Apart from the concrete coordinate of the player, the dataset also included some attributes like player_location_area, so the numerical and the classification both exist. It implies us to do some kinds of stuff to strengthen our model, but we still have to analyze it in advance.
6. The duration of the match has a maximum of 2 hours.
7. The volume of the dataset is large. It contains a great amount of data and provides enough information for us to derive meaningful insights.
8. Instead of giving all raw datasets, some of them are already classified to be as categorical data.

4. Baseline

The baseline method we utilize for badminton stroke forecasting is ShuttleNet [15], an advanced neural network approach that integrates key features such as shot type and landing coordinate in a 2-dimensional space. ShuttleNet employs embeddings of 32 dimensions and incorporates a multi-head attention mechanism with 2 heads.

To enhance its forecasting capabilities, ShuttleNet combines two Transformer-based components: the Transformer-based rally extractor (TRE) and the Transformer-based player extractor (TPE). The TRE focuses on extracting relevant information from the rally context, while the TPE captures player-specific information. These components enable the model to consider both the overall rally dynamics and individual player characteristics.

To effectively combine the information from both the rally and player contexts, ShuttleNet employs a position-aware gated fusion network (PGFN). The PGFN weighs the contributions of the rally and player information based on their respective positions and information content. This fusion mechanism enables the model to adaptively utilize the most relevant and informative features for predicting future strokes.

5. Main Approach

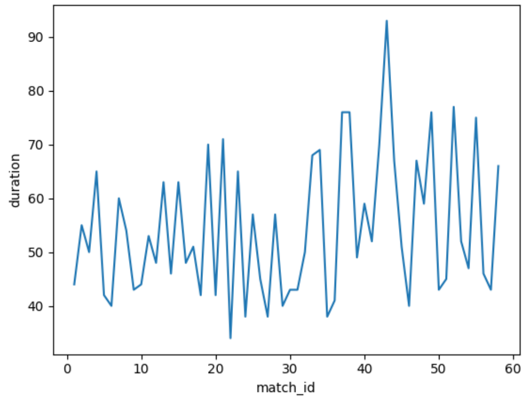
Feature Engineering & Extraction

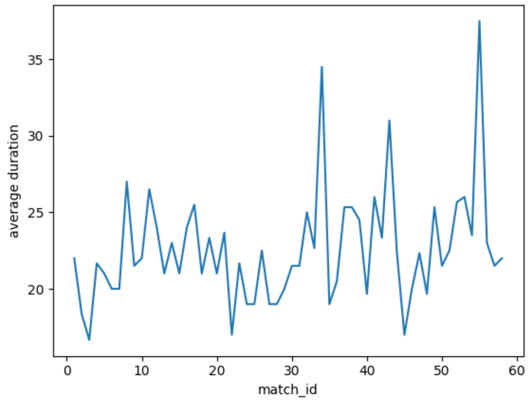
Metadata

In this dataset, we plot the trend of the duration of matches. The graph on the left uses duration averaged by the number of sets in the match.

We observe that the duration fluctuates greatly, even after averaging the duration, indicating that the duration may not be a stable indicator for matches.

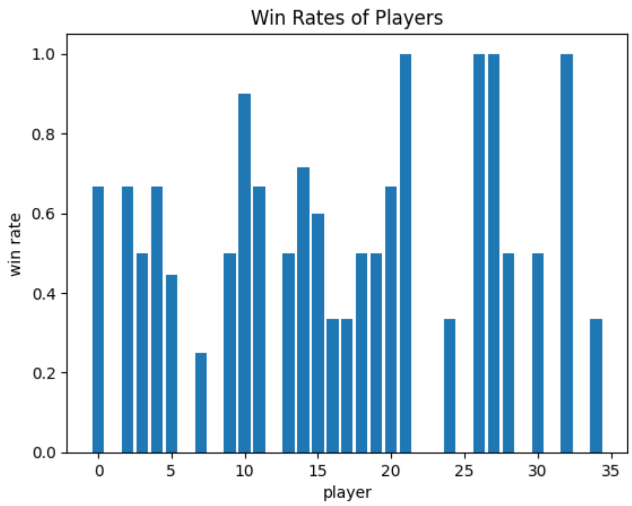
The average duration is 22.603448275862068. (Already averaged by #sets)





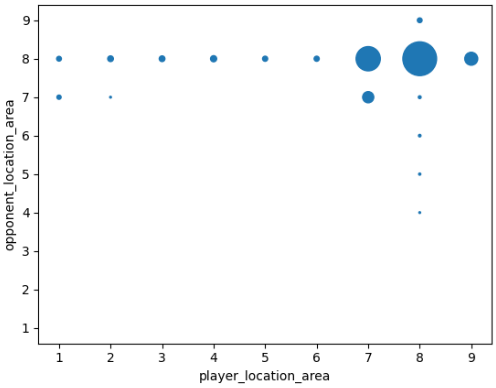
Win Rates

To know how well the players perform, we utilize the data in meta_data.csv and get the win rates with win_count / total_count. We can see that most players have win rates equal to or more than 50%. Some players even excel at the match and win several times, but some players have a 100% win rate only because they participate in just one match. The best player of all is Player 10, he joins 10 games and wins 9 of them.



val_given

To see the positioning behavior of players, we look into the data and observe the relationship between the player's location_area and the opponent's.



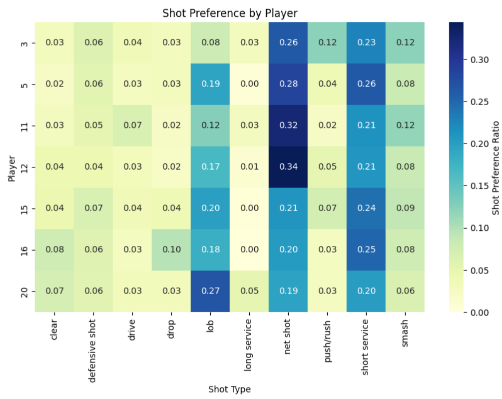
In most cases, including in the training data, they tend to stand in area 8, which is the middle of the field. Since they only need to stand in other areas that are close to the edge when they do certain strokes like shot service, they often stay in the middle in all other circumstances. Also, the player moves along area 7~9 more than the opponent. The opponent mostly stays in area 8.

Top 3 for the counts of location_area pairs:

1. (8, 8): 705
2. (7, 8): 363
3. (9, 8): 104

<u>Shot preference</u>

The shot preference/frequency distribution of a player may affect its choice on shot types. With this intuition, the shot preference is then computed with `type` and `player`, and the result is shown on the right.



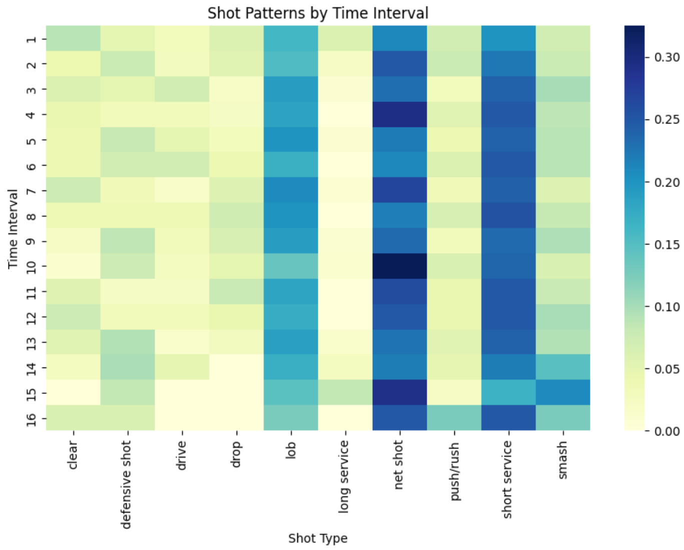
It is worth mentioning that the majority of players like three certain types of shots, which are lob, net shot, and short service. Also, some players do have a strong preference for shot-type choices. For instance, Player 12 prefers net shots more than others do, so we can definitely observe that the shot preference does make a difference.

The graph below is an alternate presentation of shot preference for players.

Shot Type v.s. Time

Here, we consider the relationship between shot types and the time sequence. The time interval is computed by `time`, setting the interval to 5 minutes. The index of the time interval is assigned to the data based on `time`. And the counts of shot types are normalized to be between 0 and 1. We observe that the shot patterns among all players for each time interval have similar results as the shot preference.

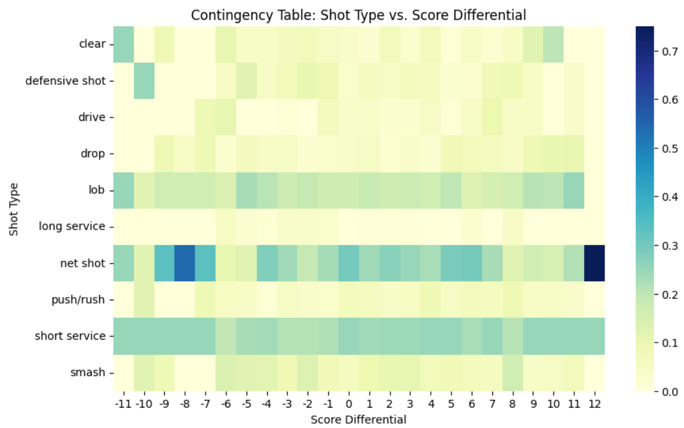
While reaching the end of the match, smash and push/rush are used more often than in previous time intervals.



Shot Type v.s. Score Difference

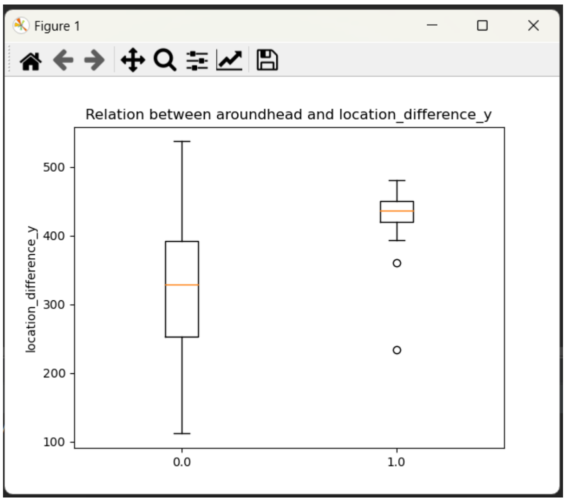
Since the score difference may put pressure on the player if the difference is large, we look into the relationship between the shot type and score difference. The shot patterns also have similar results as the two analyses above, where lob, net hot, and short service are used most frequently.

When the score difference is really large, as shown on the two sides of the graph, we observe that players tend to perform net shots.



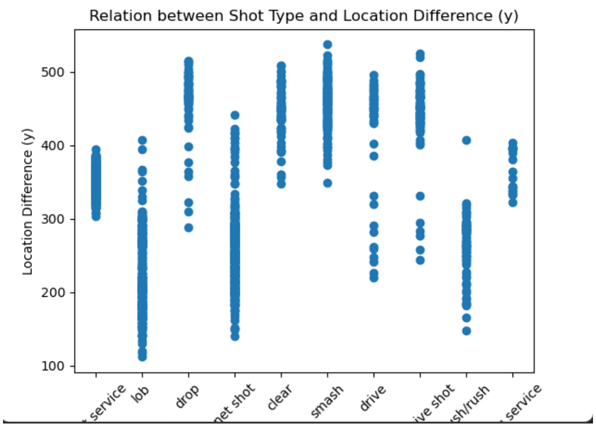
Shot Type v.s. y-axis Difference

The distance on the y-axis also has an impact on whether the shuttle is around the head or not. If the distance is far enough, the player might hit the shuttle around the head.



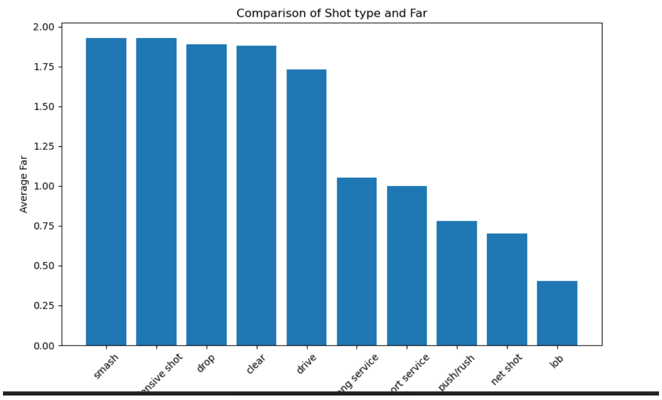
Shot Type v.s. Distance (continuous)

When the distance between the player and the opponent on the y-axis is at some level, some of the shot types might occur rather than others of the shot types. It helps us strengthen our model by giving more detailly information.



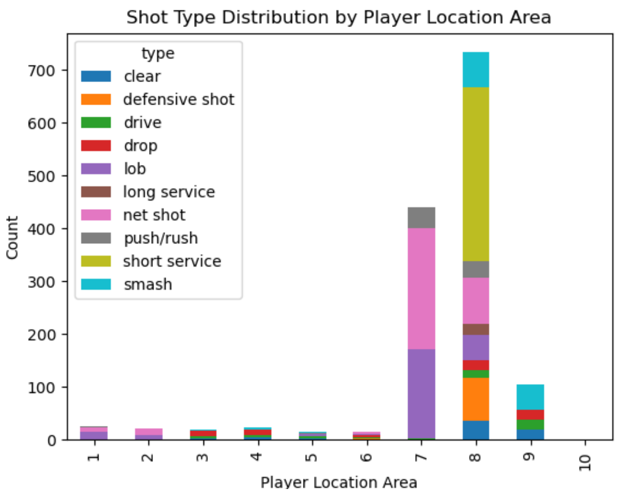
Shot Type v.s. Distance (category)

From the above graph we can see that some of the shot types have a higher average of the classified distance, so the new feature is able to help us judge if some of the shot types are more probable to occur.



Shot Type v.s. Player Location Area

We can see that player_location_area has an impact on the shot type to use, giving us more confidence in predicting some of the shot types.



Modeling

The framework we use is ShuttleNet. However, since the basic one only used two features, area and shot type, we modified some parts, so that the model can fit the multiple features we gave.

Features

We use 12 features:

1. shot-type (s): the type of shot, 10 categories.
2. area (a): the x-y coordinates of the shuttle destinations.
3. score-difference (sd): the difference between A and B (PointA-PointB), categorical feature.
4. time-difference (td): the time between this hit and the last hit.
5. aroundhead (ah): hit the shuttle around the head or not, 2 categories.
6. backhand (bh): hit the shuttle with a backhand or not, 2 categories.
7. landing_height (lh): if the shuttle destination is hit above or below the net, 2 categories.
8. shot_angle (sa): the angle of the players' location.
9. distance (d): the distance between players
10. x_distance (xd): whether the two players are standing in the same direction, 2 categories.
11. y_distance (yd): whether the two players are standing near or far, 3 categories.
12. player (p): the player who hits the shuttle.

Categorical features: shot-type, score-difference, aroundhead, backhand, landing_height, x_distance, y_distance, player

Continuous features: area, time-difference, shot_angle, distance

Problem Formulation

Let $R = \{S_r, P_r\}^{|R|}_{r=1}$ denote historical rallies of badminton matches, where the r-th rally is composed of a stroke sequence with the first 11 features we mention above, $S_r = (<s_1, a_1, sd_1, td_1, ah_1,bh_1, lh_1, sa_1, d_1, xd_1, yd_1>, \cdot\cdot\cdot, < s_{|Sr|}, a_{|Sr|}, sd_{|Sr|}, td_{|Sr|}, ah_{|Sr|}, bh_{|Sr|}, lh_{|Sr|}, sa_{|Sr|}, d_{|Sr||}, x\_d_{|Sr|}, y\_d_{|Sr|}>)$, and a player sequence $Pr = (p_1, \cdot\cdot\cdot, p_{|Sr|})$. We formulate the problem of stroke forecasting as follows. For each y, given the observed τ strokes $(<s_i, a_i,...,x\_d_i,y\_d_i>)^{\tau}_{i=1}$ with players $(p_i)^{\tau}_{i=1}$, the goal is to predict the future strokes including shot types and area coordinates for the next n steps.

Data preprocessing: Padding

For those are categorical features, we pad them and regard them as integer type. As for those continuous features, we pad them and view them as float type.

Embedding Layer

For categorical features f, f' is a categorical feature embedding projected from f_i using $M^f \in R^{Ns \times d}$, where N_s is the number of categories. As for those continuous features g, g' is a continuous feature embedding projected from g_i using $M^g \in R^{k \times d}$ with the ReLU activation function, where k is the feature's dimension. In order to make use of the rest 10 features in shot types and area, those 10 embeddings are added to both type embeddings and area embeddings. As a result, the output of the embedding layer at i-th stroke e_i is calculated as follows:

$e_i = <e^s_i , e^a_i >$

$e^s_i = s'_i+p'_i+sd'_i+td'_i+ah'_i+bh'_i+lh'_i+sa'_i+d'_i+xd'_i+yd'_i$

$e^a_i = a'_i+p'_i+sd'_i+td'_i+ah'_i+bh'_i+lh'_i+sa'_i+d'_i+xd'_i+yd'_i$

Other part of framework

We did not modify the rest of part in ShuttleNet framework, including, Type-area-attention layer, Transformer-based Player Extractor (TPE), Position-aware Gated Fusion Network (PGFN), and Prediction Layer.

Hyperparameter tuning

Since the data is quite small, we think that a complex model might lead to overfitting. Thus, we did some changes to prevent the model from this problem:

1. batch size: change from 32 to a smaller one, 8.
2. encode-dimension: change from 32 to a smaller one, 8.

However, after we add multiple features, the model seems a little bit underfitting, so we did another change:

1. epoch: change from 150 to 400.

6. Evaluation Metric

The evaluation metric we used is identical to the metric used in the competition:

$$\text{Score} = \min(l_1, l_2, l_3, l_4, l_5, l_6)$$
$$l_i = \text{AVG}(\text{CE}+\text{MAE})$$

The score is getting the minimum value of 6 consecutive averages of cross-entropy and mean absolute error, where Cross Entropy (CE) is the loss of shot type prediction, and Mean Absolute Error (MAE) is the loss of landing area prediction.

7. Results & Analysis

Results

<u>Baseline</u>
ShuttleNet described in the Baseline chapter.

<u>Case 1</u>
ShuttleNet with all columns in the training dataset as features without preprocessing. Including, shot-type, area, player, roundscore_A, roundscore_B, aroundhead, backhand, landing_height, landing_area, player_location_area, player_location_x, player_location_y, opponent_location_area, opponent_location_x, and opponent_location_y.

<u>Case 2</u>
ShuttleNet with the 12 features after feature extraction.

model	epoch	batch	encode dim	Training loss			Validation Score	row No.
				CE	MAE	Total		
ShuttleNet (Wang's)	150	32	32	1.7121	2.1508	3.8629	2.8836	1
ShuttleNet (raw data)	150	32	32	1.4838	2.2143	3.6982	3.2094	2
	50	32	32	1.9727	2.8773	4.8499	3.1415	3
ShuttleNet (feature extraction)	50	32	32	1.9943	2.8309	4.8252	3.1237	4
	50	8	16	1.962	2.7782	4.7403	3.1067	5
	200	8	16	1.9336	2.3976	4.3311	2.8518	6
	200	8	8	1.946	2.5129	4.4588	2.8295	7
	400	8	8	1.9401	2.375	4.3151	2.7975	8(best)
	500	8	8	1.9383	2.347	4.2853	2.8088	9

Compare to the baseline, the model with feature extraction that trains for 400 epoch, with batch size 8 and encode-dimension 8 reach a better validation score.

Analysis

1. Comparing the 1st and 2nd rows under the same epoch, batch size, and encoding dimension, the model trained with all columns has a lower performance than the

model trained only on two features. We speculate that the higher feature dimension might make it difficult to generalize, and lead to overfitting.
2. Comparing the 2nd and 3rd rows, when we train the model with a large batch size of 32 and a high of 32, it tends to easily become overfitting. Reducing the training epoch can help alleviate this problem.
3. Comparing the 3rd and 4th rows, after performing feature extraction, the model's performance slightly improves. However, due to the batch size and encoding dimension being too large for the model, it still suffers from overfitting. As we can see in the 5th row, training with a smaller batch size of 8 and a smaller encoding dimension of 16 really improves the performance.
4. Comparing the 5th and 6th rows, since we use a small batch size of 8 and a small encoding dimension of 16, training for only 50 epochs is not enough and leads to underfitting. Therefore, we improved the performance by increasing the epoch from 50 to 200.
5. Comparing the 6th and 7th rows, changing the encoding dimension from 16 to 8 improves the model's performance. Furthermore, as observed from the 7th to the 9th rows, training with 400 epochs results in a better validation score. Therefore, we assume that training with 200 epochs is slightly underfitting while training with 500 epochs is slightly overfitting.

8. Future Work

Our proposed method primarily focuses on extracting information from the available data and expanding the model to accommodate additional turn-based features. As part of future work, we intend to enhance our method's feature extraction capabilities by incorporating techniques such as Convolutional Neural Networks (CNNs). This integration will enable us to obtain more precise information regarding player movements and decision-making processes. Additionally, we plan to extend the model to encompass more "global" information, such as individual players' playing styles or preferred sweet spots. By incorporating these aspects, we aim to provide a more comprehensive and nuanced view of the game, enhancing the overall analysis and understanding of badminton strokes.

9. Code

Github link

[drink970082/CoachAI-Projects: Official research projects of badminton CoachAI (github.com)](https://github.com/drink970082/CoachAI-Projects)

Environment

Python 3.7.0

Usage

- Train a model
  ```
  ./script.sh
  ```
- Generate prediction
  ```
  python generator.py {model_path}
  ```
- Run evaluation metrics
  Both ground truth and prediction files are default in the data folder
  ```
  python evaluation.py
  ```

Hyperparameters

'model_type': 'ShuttleNet', 'seed_value': 42, 'max_ball_round': 70, 'encode_length': 4, 'batch_size': 8, 'lr': 0.0001, 'epochs': 400, 'n_layers': 1, 'shot_dim': 8, 'area_num': 5, 'area_dim': 8, 'player_dim': 8, 'encode_dim': 8, 'num_directions': 1, 'K': 5, 'sample': 10, 'gpu_num': 0

Result

- total_loss: 4.3151
- total_shot_loss: 1.9401
- total_area_loss: 2.375
- validation score: 2.7975

10. Contribution

Name	StudentID	Task	Contribution
陳盈均	109705011	Feature Engineering	25%
陳沂亨	110700022	Feature Engineering	25%
陳以瑄	109705001	Modeling	25%
吳振豪	109705003	Modeling	25%

11. References

[1]Ghosh, I., Ramasamy Ramamurthy, S., Chakma, A., & Roy, N. (2023). "Sports analytics review: Artificial intelligence applications, emerging technologies, and algorithmic perspective." *WIREs Data Mining and Knowledge Discovery,* e1496.
[2]Claudino, João Gustavo, et al. "Current approaches to the use of artificial intelligence for injury risk assessment and performance prediction in team sports: a systematic review." *Sports medicine-open 5 (2019)*: 1-12.
[3]Abdullah Erdal Tümer & Sabri Koçer (2017) "Prediction of team league's rankings in volleyball by artificial neural network method", *International Journal of Performance Analysis in Sport*, 17:3, 202-211
[4]Brooks, Joel, Matthew Kerr, and John Guttag. "Developing a data-driven player ranking in soccer using predictive model weights." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016.
[5]Ghosh, Indrajeet, Sreenivasan Ramasamy Ramamurthy, and Nirmalya Roy. "Stancescorer: A data driven approach to score badminton player." *2020 IEEE international conference on pervasive computing and communications workshops (PerCom Workshops).* IEEE, 2020.
[6]Cuiping, Cai. "Badminton video analysis based on player tracking and pose trajectory estimation." *2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA).* IEEE, 2021.
[7]Sharma, Manoj, et al. "Badminton match outcome prediction model using Naïve Bayes and Feature Weighting technique." *Journal of Ambient Intelligence and Humanized Computing* 12 (2021): 8441-8455.
[8]Nokihara, Yuka, et al. "Prediction of Shuttle Trajectory in Badminton Using Player's Position." (2023).
[9]Nokihara, Yuka, et al. "Future Prediction of Shuttlecock Trajectory in Badminton Using Player's Information." *Journal of Imaging* 9.5 (2023): 99.
[10]Waghmare, Govind, et al. "Badminton shuttlecock detection and prediction of trajectory using multiple 2 dimensional scanners." *2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI)*. IEEE, 2016.
[11]Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems* 27 (2014).
[12]Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
[13]Kong, Yu, Zhiqiang Tao, and Yun Fu. "Deep sequential context networks for action prediction." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
[14]Giuliari, Francesco, et al. "Transformer networks for trajectory forecasting." *2020 25th international conference on pattern recognition (ICPR)*. IEEE, 2021.
[15]Wang, W.-Y., Shuai, H.-H., Chang, K.-S., & Peng, W.-C. (2022). ShuttleNet: Position-Aware Fusion of Rally Progress and Player Styles for Stroke Forecasting in Badminton. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4), 4219-4227.