# Implementation of GRU Model in Badminton Time Series Data for Movement Trajectory Prediction

Ming-Hung Lin [1], Luu-Ly Tran [2,*], Han-Yu Chen [3], and Chih-Chieh Chang [4]

[1] Graduate Institute of A.I. Cross-Disciplinary Technology, National Taiwan University of Science and Technology, Taiwan
[2] Department of Information Management, National Taiwan University of Science and Technology, Taiwan
[3] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taiwan
[4] School of Management, National Taiwan University of Science and Technology, Taiwan
Email: d11352002@mail.ntust.edu.tw (M.-H.L.); M11309813@mail.ntust.edu.tw (L.-L.T.); M11115Q24@mail.ntust.edu.tw (H.-Y.C.); ccchang@mail.ntust.edu.tw (C.-C.C.)
*Corresponding author

***Abstract*—This study tackles the challenge of image recognition in badminton due to its small size and complex movement trajectory by developing a new dataset and predicting shot sequences. The experimental procedure involves two main steps: detailed pre-processing of original data (extracting key features, normalizing, and dividing into training and test sets) and using deep learning algorithms to predict future shots. We validated the model with 20 repeated experiments and multiple evaluation metrics (Accuracy, Precision, Recall, F1-Score) and confirmed statistical significance and stability through T-tests. The results demonstrate that the Gated Recurrent Unit (GRU) effectively predicts shot sequences, benchmarking with Long Short-Term Memory (LSTM) and One-Dimensional Convolutional Neural Networks (1D-CNNs). The proposed architecture model with joint Rectified Linear Unit (ReLU) achieved 59.77% accuracy compared to a random guessing baseline of 16.67%, showcasing superior performance and promising applications. Therefore, this research provides a new tool for badminton data analysis and lays the groundwork for broader sports data analysis and behavior prediction.**

***Keywords*—sports analytics modelling, time-series analysis, sequential structured dataset, supervised learning**

## I. INTRODUCTION

Today's sports science is evolving rapidly, and its research and applications play a crucial role in improving athlete performance. In the context of badminton, existing sports science studies primarily focus on leveraging advanced sensing technologies to monitor and analyze athletic movements and biomechanical data. First, sensor technology is widely used to monitor the physiological parameters of players to provide instant physical condition assessment. For example, Sakurai and Ohtsuki [1], Faude *et al.* [2] used sensors to monitor players' heart rate, blood lactate concentration, and muscle activity patterns during sparring. Secondly, the use of wearable devices and Internet of Things (IoT) devices for human activity recognition has become one of the important directions in badminton sports scientific research. For instance, Ghosh *et al.* [3] captures the user's limb movements and scores the batting posture and analysis. Through in-depth analysis of sports data, researchers can identify athletes' movement patterns and technical flaws and provide targeted technical guidance based on the data to help athletes improve their technical levels. In addition, through motion sensors installed on the racket, such as accelerometers and gyroscopes, the player's data when hitting the ball is collected [4]. These data include the speed, power, angle of the bat, and the movement trajectory of the racket, which can help athletes better understand and optimize their batting movements. The motion sensor on the racket can record the details of each shot, including the position of the hitting point and changes in hitting force. This data is of great significance for refined action analysis and technical improvement.

Integrating artificial intelligence into badminton presents notable challenges due to the sport's fast-paced and complex nature. Shuttlecocks travel at high speeds with unpredictable trajectories, complicating detection and tracking for vision systems. Dynamic camera angles, lighting variations, motion blur, and occlusion further reduce recognition accuracy and stability [5]. Compared to other sports, badminton suffers from limited annotated datasets, often outdated and lacking in technical and trend-related information, thereby restricting model training and analysis. Moreover, replicating the players' real-time decision-making, driven by rapid opponent and shuttlecock responses, places high demands on algorithmic complexity and computational efficiency [6].

Manuscript received October 20, 2025; revised November 12, 2025; accepted January 6, 2026; published April 24, 2026.

doi: 10.12720/jait.17.4.749-758 

To address these limitations, we conducted in-depth study on individual badminton-match settings, specifically combining time-series datasets and deep learning to improve the stroke classification capability. The specific contributions are summarized as follows:

1) Formalize the hitting method and landing point for every shot type, thereby establishing a well-structured sequential dataset suitable for downstream predictive modelling.
2) Create an exclusive dataset for the battle between Lin Jun-Yi and Lee Zi-Jia.
3) Propose a GRU-based algorithm to predict the type of shot that the play will make. The prediction accuracy is 59.77%, which is greater than other architectures.
4) Find out the optimal parameters of the GRU model during training.

The remainder of this paper is organized as follows: Section II introduces several existing studies related to our work. Section III describes the model architecture and training objectives that we applied. Next, detailed model configuration and experimental results are introduced in Section IV. Finally, we highlight the main achievements and propose future work in Section V.

## II. Related Work and Background

### A. Time Series Data

Time series data refers to a data sequence arranged in chronological order, recording characteristics and phenomena observed at different points in time. This data type usually reflects the trend of specific traits or indicators changing over time, making the data continuous and sequential.

Using financial time series or historical data of the stock market, combined with deep learning technology, you can identify dependencies in the stock market price series and obtain helpful information from recent transaction data to predict the stock price trend at the next moment, effectively predicting the changes in the short term to support investment decisions [7, 8]. Patient health status and chronic diseases can be predicted by analyzing time series data in electronic health records, including diagnosis and medication codes. These predictions help medical professionals make timely diagnosis and treatment decisions, thereby improving patient outcomes [9].

The above applications demonstrate the high potential of combining time series data with artificial intelligence. This synergy not only helps us better understand and predict complex dynamic systems but also has the potential to revolutionize various aspects of our lives.

### B. Sports Data Analysis

#### 1) Badminton analytics and existing models

Although athlete performance has been widely quantified and analyzed in other sports, such as basketball and football, data analysis in badminton remains relatively preliminary, with few related studies. The previous works of Wang *et al*. [10, 11] proposed a deep learning model consisting of a novel short-term extractor and a long-term encoder designed to capture shot sequences in badminton matches. The core of these research is to convert batting behavior into quantifiable data and analyze it by predicting the game's outcome. The deep learning model can identify potential patterns and regularities by training on a large amount of game data. These patterns are significant for understanding athletes' performance and formulating game strategies. Wang *et al*. [12] proposed a novel framework, ShuttleNet, which combines pull progression and player-style position awareness. ShuttleNet incorporates game progress and player information through two modified encoder-decoder extractors, allowing the model to capture dynamic changes in the game more accurately. This framework can predict not only the type of return shot but also its location.

Traditional sequence-based models usually only consider the behavior of a single athlete and ignore the interactions between athletes. Based on this, Chang *et al*. [13] proposed a Dynamic Motion Forecasting (DyMF) model based on Player Movement (PM) graph and hierarchical fusion. This model features an interaction style extractor that captures interactions between players and those on both sides of a match, analyzing dynamic strategies over time. Introducing the DyMF model provides a new perspective for sports data analysis. It can not only analyze the behavior of individual players but also capture the interactions between them, thereby providing more comprehensive and accurate prediction strategies and tactics.

#### 2) Data analytics in other sports

Early data analysis mainly relied on manual records and simple statistical methods. With the advancement of digital and information technology, sports data analysis has developed rapidly.

Data analysis has become vital to the team's decision-making process in professional sports, such as Major League Baseball (MLB) and the National Basketball Association (NBA). The work of Lewis [14] used data analysis to measure player statistics, such as stolen base rate and batting average, and then selected players to formulate game strategies. This process was called the "Moneyball" revolution and was a classic case of an MLB data analysis application. Additionally, Oliver [15] studies of historically successful NBA teams describe and quantify the work of team leaders and players. These leagues are the first to adopt data analysis technology for player selection, game strategy, and player health management, demonstrating the potential and value of data analysis in professional sports. With the successful application of data analysis in these fields, other sports and amateur sports enthusiasts have gradually begun to accept and use these technologies. Sports data analysis is used to optimize athlete performance and enhance both event presentation and the overall experience for sports fans.

In addition, injuries are one of the major problems faced by athletes. By analyzing sports data, medical and training teams can identify potential injury risks early on. Bittencourt *et al.* [16] studies how to use mechanics to control stride frequency and length during running to reduce the risk of injury for runners. Analyzing athletes' movement data, Schubert *et al.* [17] predicts possible injury locations and provides preventive advice. As technology advances, the scope and depth of applications of sports data analysis will continue to expand.

*3) Sequence modeling techniques*

In modern deep learning, processing sequence data is essential yet challenging. Researchers have therefore proposed various model architectures capable of capturing temporal dependencies effectively. Among these, Gated Recurrent Units (GRUs) have received considerable attention. In 2014, Cho *et al.* [18] introduced the GRU architecture and demonstrated that appending a linear layer for output projection greatly enhances downstream performance in statistical machine translation systems by converting the hidden representations into task-specific outputs. Subsequently, Chung *et al.* [19] conducted a comprehensive empirical study comparing GRU and LSTM units across multiple sequence modeling tasks. Their findings show that although GRU has a simpler gating structure than LSTM, it frequently matches or surpasses LSTM performance while requiring fewer parameters and lower computational cost.

LSTM are designed specifically to address the vanishing-gradient issue in vanilla RNNs by introducing an explicit memory cell and a set of gates (input, forget, and output) that regulate information flow over time. This architecture often makes LSTM particularly effective when the target signal depends on longer-range temporal context, where preserving and selectively updating historical information is crucial. In data-limited settings like this study, the larger capacity can also raise overfitting risk unless regularization and careful tuning are applied. As a result, LSTMs are frequently a strong choice for capturing long-term dependencies, but they may be less attractive than GRUs when computational efficiency and lightweight deployment are primary constraints.

Beyond recurrent architectures, 1D-CNNs have also been widely adopted for sequence modeling tasks. Unlike RNN-based models, 1D-CNNs leverage convolutional kernels to extract local temporal patterns efficiently, offering strong feature extraction capability with parallel computation and reduced latency. Peralta *et al.* [20] have shown that 1D-CNNs can serve as a competitive baseline for time-series prediction and classification tasks due to their ability to model short-range dependencies with relatively lightweight computation.

In summary, GRU, LSTM, and 1D-CNNs represent three widely used paradigms for sequence modeling, each with distinct strengths. GRU provides an attractive balance of modeling ability and efficiency, capturing long- and short-term patterns with fewer parameters than LSTM. Conversely, 1D-CNNs excel at local pattern extraction and fast inference, but may struggle with long-range temporal dynamics compared to recurrent units. These complementary characteristics provide a strong foundation for the comparative experiments in this study, where we benchmark GRU against LSTM and 1D-CNNs under consistent settings to evaluate their relative performance.

## III. Methodology

This study proposes a sequential deep learning framework to predict the next stroke in professional badminton rallies, with specific focus on the exclusive dataset constructed from the high-level match between Lin Jun-Yi and Lee Zi-Jia. The dataset records every shot exchanged between the two players, including categorical stroke types and temporally ordered rally information. To model these exchanges effectively, the proposed architecture is organized into three main modules: (1) Input Stage, (2) Sequence Modeling Stage, and (3) Output Stage. Fig. 1 is the architecture of our proposed framework.

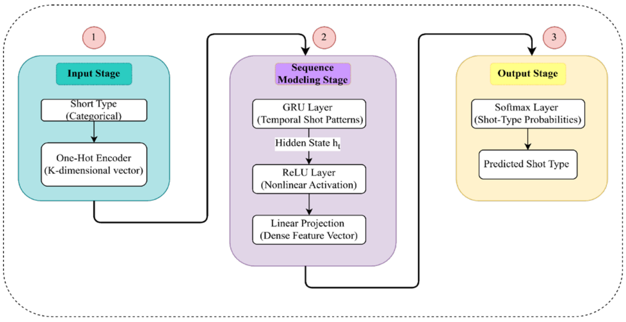



Fig. 1. Experimental architecture.

### A. *Input Stage*

The goal of the input stage is to convert each raw shot event into a compact numerical representation that can be processed by the recurrent model. Each shot in the dataset is recorded as a categorical label representing its type (e.g., smash, clear, net shot, lift, drop, drive). Because categorical variables cannot be directly processed by neural networks, each shot label is transformed into a one-hot vector.

Let $c_t \in \{1, 2, \dots, K\}$ be the categorical shot type at time step $t$, where $K$ is the total number of stroke categories in the Lin–Lee dataset. The one-hot encoded input is defined as:

$$x_t = one\_hot_{(c_t)} \in \mathbb{R}^K, \quad x_t[i] = \begin{cases} 1, if\ i = c_t, \\ 0, otherwise \end{cases} \quad (1)$$

Each rally becomes a sequence $X = (x_1, x_2, \dots, x_t)$, preserving the temporal order of the shots. Since the dataset contains rallies of varying lengths, the model naturally accommodates sequences of arbitrary duration without interpolation or alignment. This encoding preserves the pure categorical identity of Lin Jun-Yi and Lee Zi-Jia's tactical choices and forms the basis for subsequent sequential modeling.

### B. *Sequence Modeling Stage*

The core of the architecture is a GRU designed to capture temporal dependencies in the shot exchanges. Badminton rallies often exhibit structured patterns such as offensive-defensive transitions, pressure sequences, and counter-attack tendencies, all of which appear in the Lin–Lee match. GRUs are particularly suitable because they effectively model long-term dependencies while avoiding vanishing gradients.

*1) GRU formulation*

The structure diagram of GRU is shown in Fig. 2. Given input $x_t$ and previous hidden state $h_{t-1}$, the reset gate $r_t$, update gate $z_t$, hidden state $\tilde{h}_t$, and update hidden state of the GRU layer are calculated according to Eqs. (2)–(5) as below:

$$r_t = \sigma(W_r \cdot x_t + U_r \cdot h_{t-1} + b_r) \quad (2)$$

$$z_t = \sigma(W_z \cdot x_t + U_z \cdot h_{t-1} + b_z) \quad (3)$$

$$\tilde{h}_t = tanh(W_h \cdot x_t + r_t \circ (U_h \cdot h_{t-1}) + b_h) \quad (4)$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t \quad (5)$$

where $z_t$ controls the weight ratio of the old state $h_{t-1}$ to the new candidate hidden state $\tilde{h}_t$. The resulting $h_t$ is a learned latent representation of the rally state at time $t$, encoding: cumulative tendencies of Lin Jun-Yi, counter-strategies of Lee Zi-Jia, pace and tempo of the rally, local transitions between shot types.

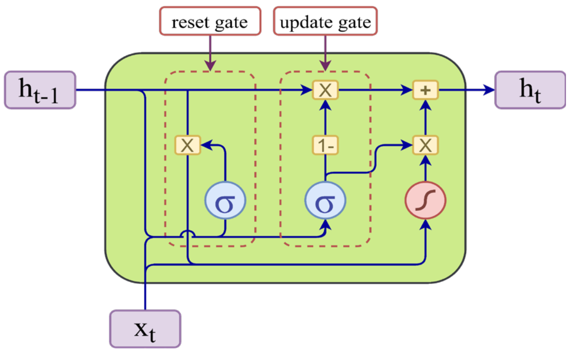



Fig. 2. GRU structure diagram.

*2) Nonlinear transformation*

To enhance expressiveness, the GRU output is passed through a Rectified Linear Unit (ReLU) activation:

$$v_t = ReLU(h_t) = max(0, h_t), \quad (6)$$

introducing nonlinearity and ensuring stable gradient propagation.

*3) Linear projection*

A fully connected layer transforms the $ReLU$ vector into a dense representation:

$$f_t = W_f v_t + b_f, \quad (7)$$

where $f_t \in \mathbb{R}^K$ corresponds to the logit scores of the next possible shot types. This stage effectively compresses the interaction between the two players into a temporally informed feature vector, learned directly from their real match dynamics.

## *C. Output Stage*

The output stage converts the dense feature vector into a probability distribution over all possible next-shot types. Therefore, a softmax layer is applied:

$$p_t(i) = \frac{\exp(f_t(i))}{\sum_{j=1}^{K} \exp(f_t(j))}, \quad (8)$$

where $p_t(i)$ is the probability that the next shot is of type $i$, and $f_t(i)$ are the logits produced by the Linear layer. The predicted shot class is obtained by:

$$\hat{c}_{t+1} = arg\max_i p_t(i) \quad (9)$$

This module transforms learned rally dynamics into actionable probabilistic predictions, revealing which shot is most likely to be executed next, based purely on the evolving interaction between Lin Jun-Yi and Lee Zi-Jia.

## *D. Training Objective*

To optimize model parameters, the predicted probability distribution $p_t$ is compared against the ground-truth one-hot label $y_t$ using the standard cross-entropy loss:

$$L = -\sum_{i=1}^{K} y_t(i) \log p_t(i) \quad (10)$$

Since $y_t(i) = 1$ only for the correct class, the loss simplifies to:

$$L = -\log p_t(c_t) \quad (11)$$

This loss encourages the model to assign high probability to the true next shot type, improving predictive accuracy over the Lin Jun-Yi versus Lee Zi-Jia rally dataset.

# IV. EXPERIMENT

## *A. Model Settings*

The experimental design uses 80% of the dataset for training and 20% for testing. The GRU model is trained with a learning rate of 0.0001, a hidden size of 64, a batch size of 8, and the Adam optimizer with a weight decay regularization parameter of 0.001. Number of epochs is 100. Use the above parameters and perform 100 iterations, and then use the four indicators of Accuracy, Precision, Recall, and F1-Score to evaluate the model's performance.

Each badminton shot is encoded as a 7-dimensional feature vector, representing the structured attributes of the rally (6 one-hot shot types and location). The model processes a fixed lookback window of 7 consecutive shots, which forms the temporal input to the GRU. A lookback window of 7 shots is chosen based on the tactical structure of badminton rallies, which typically unfold in short micro-patterns of 3 to 7 strokes [21]. The GRU produces a hidden representation for each time step, and the final hidden state is passed through a fully connected layer to generate a 6-dimensional output vector corresponding to the 6 shot-type classes. The output class corresponds to the predicted shot type at time *t*+1.

## *B. Dataset*

In a badminton match, the background information of each ball is crucial for game interpretation and analysis. Our tabular dataset includes detailed records of the scores of the two players (Lee's point and Lin's point), the player hitting the ball (player), the ball type (type), the hitting position (area), the return landing (location), and the real-time number of shots in this round (number).

The dataset comprises 244 rallies from a full match between the two players, totaling 2,936 shot events. Each shot is annotated with 6 shot types, the executing player, shot area (front/mid/back), shot location (left/center/right), and a shot index indicating its position within the rally. Although twelve shot types are defined in the dataset, only six primary categories were used for classification (the mapping is shown in Table I). This reduction addresses data scarcity and class imbalance issues while improving model stability. Raw frame-level metadata is standardized into four structured features: "type", "area", "location", "number" with consistent numerical representation across all samples. For non-spatial actions such as services, fixed placeholder values are used to preserve feature integrity.

TABLE I. DEFINITION OF BADMINTON TYPES

| Classification | Type | Definition |
|---|---|---|
| Attack | Drive | The ball has a straight flight path after passing over the net, the speed is relatively fast. |
| | Smash | The ball falls towards the opponent's court at a breakneck speed and a steep angle. |
| | Kill | An attack in front of the net. After contacting the racket surface, the ball quickly presses down on the opponent's court. |
| Back normal | Clear | Hit the ball back from your back-court to the opponent's backcourt. The flight path of the ball is high and deep. |
| | Cut | Use the angle between the racket surface and the ball to generate rotation, causing the ball to fall quickly after passing the net. |
| Defense | Return short | Return the opponent's attack and nudge the ball over the net. |
| | Return push | Return the opponent's attack and return the ball to the opponent's backcourt. |
| Lift | Lift | Return the ball from your front-court to the opponent's backcourt. |
| | Push | Similar to picking a ball, but the ball's flight path is straighter and more compressive. |
| Service | Short service | When serving, the ball falls immediately after passing the net and lands in the first half of the opponent's serving area. |
| | Long service | When serving, the ball flies a longer distance after passing through the net and lands in the second half of the opponent's serving area. |
| Short | Short | In front of your net, let the ball fall gently to the opponent's net after passing through the net. |

All fields are fully populated with 0% missing values. Player identity and shot type are complete and manually verified. Shot numbers are algorithmically assigned per rally. Rally scores are complete at rally boundaries and forward-filled for intermediate shots. Spatial fields (area/location) are provided for all relevant strokes. As a result, the dataset forms a clean, fully numerical, model-ready sequence dataset. Prior work of Wang *et al.* [22] also shows that shot type, court area, spatial direction, rally phase, and opponent's preceding stroke strongly influence decision-making and stroke selection. By encoding these elements, the dataset captures the essential context required for accurate next-shot prediction.

In selecting match combinations in the dataset, we selected rookies who have recently performed outstandingly in international badminton men's singles events: Lin Jun-Yi from Taiwan and Lee Zi-Jia from Malaysia. These two players are the new generation of badminton forces and show excellent skills and tactics on the court. Through future analysis, we hope to gain a deeper understanding of their tactical habits in the game and why they stand out compared to other rookies in the competition. Until now, Lin Jun-Yi and Li Zi-Jia have played against each other three times in official competitions, with Li Zi-Jia slightly having the upper hand with two wins and one loss. The details of these three battles are as follows:

1) 2023 Indonesia Masters: Li Zi-Jia defeated Lin Jun-Yi in straight games 21:18 and 21:17.
2) 2023 Malaysia Masters: Lin Jun-Yi defeated Li Zi-Jia in three games 21:19, 16:21, and 21:15.
3) 2024 Asian Badminton Team Championship: Li Zi-Jia defeated Lin Jun-Yi again 21:16 and 21:13.

First, we define the types of batting balls into 12 categories: short service, long service, lift, short, clear, cut, smash, push, kill, drive, return short, and return push. The definitions of each category can be found in Table I. Next, regarding the hitting position and impact point, we divide the court into 9 areas. The area division can be seen in Fig. 3. There are also Category 10 out-of-bounds and Category 11 hanging nets.

In addition, considering the usability of the dataset, breakpoints have been added to mark each round. The appearance of a segment point means that the round has ended. The breakpoint design allows other users to interpret the dataset more comprehensively and carry out analytical tasks with greater clarity.

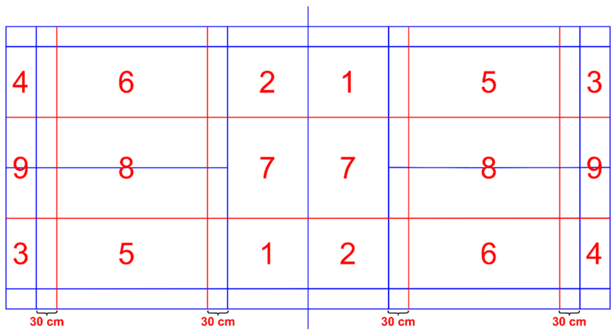



Fig. 3. Badminton court area division.

## C. Experimental Results

### 1) Experimental results display

This experiment uses past batting sequences to predict future batting ball types. The total training time is 141.54 seconds. During preprocessing, we examined multiple feature combinations derived from the "type", "number", "area", and "location" fields. Experimental results showed that the combination of six one-hot-encoded shot-type features and the "location" feature form the multi-feature input that yielded the best predictive performance. That leads to the exclusion of the "number" and "area" feature.

To demonstrate the effectiveness of these feature selections, we used the case of random guessing of six main categories as the baseline. The baseline prediction accuracy is 1/6 (approximately 16.67%). We conducted 20 repeated experiments and used a T-test to verify the difference of model performance with and without hyperparameters optimization. Detailed results can be found in Table II, with a statistically significant improvement (*p*-value < 0.05).

TABLE II. COMPARISON BEFORE AND AFTER HYPERPARAMETERS OPTIMIZATION

| Hyperparameters | Accuracy (%) | | | | | | | | | | Average (%) | *p*-value |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | 50.95 (±0.07%) | 0.00000049 |
| | 50.03 | 52.98 | 51.35 | 50.80 | 50.45 | 52.69 | 50.64 | 50.73 | 49.31 | 49.05 | | |
| | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | | |
| | 50.32 | 50.76 | 52.01 | 54.08 | 52.68 | 49.12 | 49.20 | 49.41 | 50.06 | 53.33 | | |
| Optimized | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | 53.45 (±0.02%) | |
| | 55.80 | 54.46 | 53.28 | 52.94 | 53.80 | 53.98 | 52.78 | 52.16 | 52.67 | 56.02 | | |
| | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | | |
| | 53.49 | 53.76 | 53.05 | 51.30 | 52.98 | 52.60 | 53.08 | 52.54 | 53.71 | 54.60 | | |

Based on the proposed architecture in Fig. 1, we first performed feature extraction through a layer of GRU, a type of RNN, and then performed linear transformation through Linear. We repeated the experiment 20 times and used a T-test to verify the difference before and after adding the ReLU activation function to the model. The experimental results show that adding the ReLU activation function can significantly improve the prediction accuracy of the model and enhance the prediction stability. Detailed results can be found in Table III.

To further validate the effectiveness of our proposed GRU-based architecture, we benchmarked it against two widely used sequence-modeling baselines: LSTM and 1D-CNNs. All three models were trained and evaluated using the same sequence input, identical preprocessing procedures, and matched parameter settings to ensure a fair comparison. Although LSTM is traditionally known for its ability to better preserve long-term dependencies, our experiments indicate that GRU delivers superior performance in this specific application. In particular, the GRU model achieves the highest prediction accuracy while maintaining greater computational efficiency. By contrast, the 1D-CNNs baseline performs moderately well but remains less competitive than GRU, and the LSTM model ranks last under the same conditions. The quantitative results supporting this comparison are summarized in Table IV. This indicates that GRU has better practicality and balance when processing badminton match data. After many parameter adjustments, we determined the configuration that best suited the model. To evaluate the performance of this configuration, we used four metrics: Accuracy, Precision, Recall, and F1-Score. Table V shows that after adding the ReLU activation function to the model, all indicators perform better than in the model without it. These results further demonstrate the effectiveness of our model in predicting pitch type, significantly exceeding the baseline accuracy of random guessing.

TABLE III. COMPARISON BEFORE AND AFTER OPTIMIZATION WITH RELU

| Joined ReLU | Accuracy (%) | | | | | | | | | | Average (%) | *p*-value |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | 53.45 (±0.02%) | 0.00000012 |
| | 55.80 | 54.46 | 53.28 | 52.94 | 53.80 | 53.98 | 52.78 | 52.16 | 52.67 | 56.02 | | |
| | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | | |
| | 53.49 | 53.76 | 53.05 | 51.30 | 52.98 | 52.60 | 53.08 | 52.54 | 53.71 | 54.60 | | |
| Optimized | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | 59.77 (±0.05%) | |
| | 60.58 | 60.58 | 59.38 | 59.38 | 60.41 | 59.55 | 60.07 | 59.89 | 60.41 | 59.72 | | |
| | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | | |
| | 59.73 | 60.23 | 59.38 | 59.21 | 58.87 | 58.87 | 59.21 | 59.73 | 59.73 | 60.67 | | |

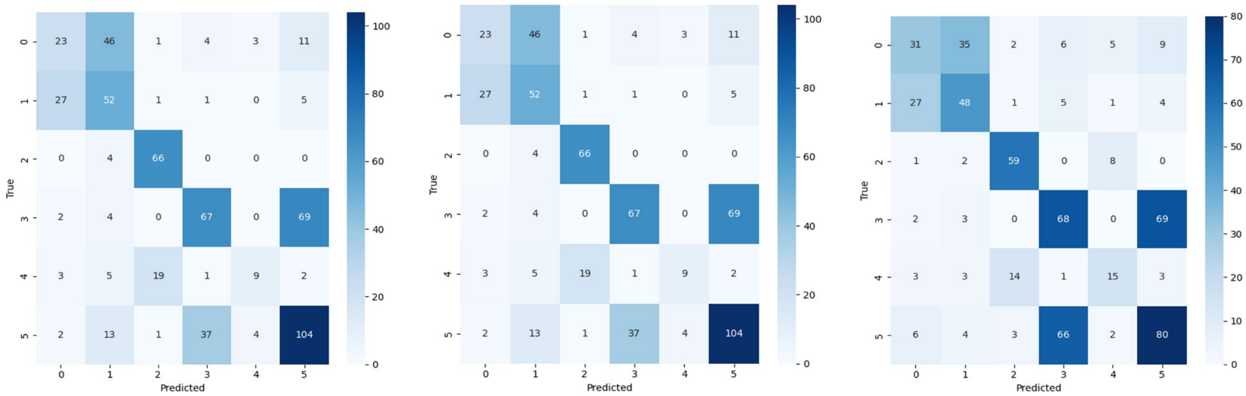
TABLE IV. COMPARISON OF LSTM, 1D-CNNS AND GRU

| Models | Accuracy (%) | | | | | | | | | | Average (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LSTM | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | 56.80 (±0.06%) |
| | 57.16 | 55.80 | 56.48 | 56.99 | 56.14 | 56.65 | 57.84 | 55.97 | 56.31 | 57.84 | |
| | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | |
| | 56.65 | 57.16 | 56.48 | 56.82 | 56.65 | 57.33 | 57.50 | 56.14 | 56.82 | 57.16 | |
| CNN | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | 59.22 (±0.06%) |
| | 58.87 | 58.70 | 59.21 | 58.70 | 58.36 | 58.19 | 59.38 | 59.21 | 59.04 | 58.87 | |
| | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | |
| | 59.04 | 59.38 | 60.58 | 59.21 | 60.06 | 59.38 | 59.38 | 58.87 | 59.38 | 60.40 | |
| GRU | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | 59.77 (±0.05%) |
| | 60.58 | 60.58 | 59.38 | 59.38 | 60.41 | 59.55 | 60.07 | 59.89 | 60.41 | 59.72 | |
| | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | |
| | 59.73 | 60.23 | 59.38 | 59.21 | 58.87 | 58.87 | 59.21 | 59.73 | 59.73 | 60.67 | |

As shown in Table V and Fig. 4, the GRU model delivers the best performance, which exhibits strong temporal modeling capability and the most stable class-wise predictions, with only moderate confusion between similar classes. The 1D-CNNs baseline performs well when local discriminative features are prominent but struggles to capture longer-range dependencies, resulting in lower overall accuracy than GRU. In contrast, the LSTM model shows excellent performance for a few specific classes (particularly Class 5) but suffers from substantial misclassifications across several other categories. Therefore, it is the least stable and least accurate among the three architectures.

TABLE V. MODEL PERFORMANCE IN VARIOUS INDICATORS

| Models | Accuracy (%) | Precision (%) | Recall (%) | F1-Score |
|---|---|---|---|---|
| **No ReLU** | 53.45 (±0.07%) | 53.84 | 55.27 | 0.54 |
| **Joint ReLU** | 59.77 (±0.05%) | 58.51 | 59.59 | 0.58 |

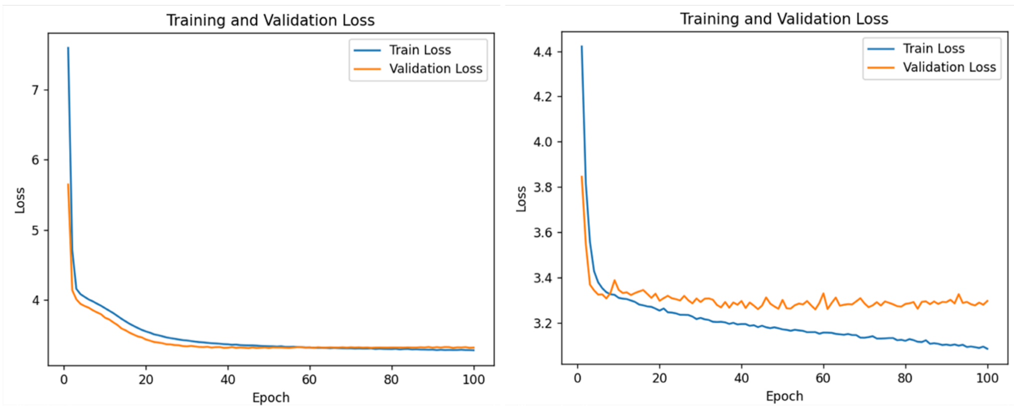



(a) (b) (c)

Fig. 4. Confusion matrix of three architectures: (a) LSTM; (b) 1D-CNNs; (c) GRU.

*2) Discussion on convergence conditions of experimental parameters*

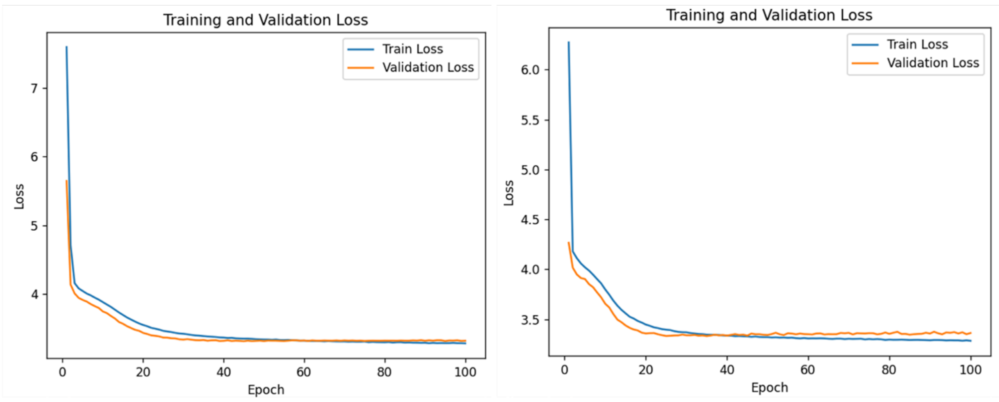
To optimize model performance, we systematically evaluated the effects of various hyperparameters through controlled experiments. Fig. 5 compares convergence behaviours under two learning rates: 0.0001 (Fig. 5(a)) and 0.001 (Fig. 5(b)). A learning rate of 0.0001 yields a smooth and stable loss curve, while 0.001 causes severe oscillation and divergence, emphasizing the critical role of careful learning rate selection for stable training. Fig. 6 shows the convergence test results for different hidden layer sizes (hidden size = 64 (Fig. 6(a)) and hidden size = 128 (Fig. 6(b)).




(a) (b)

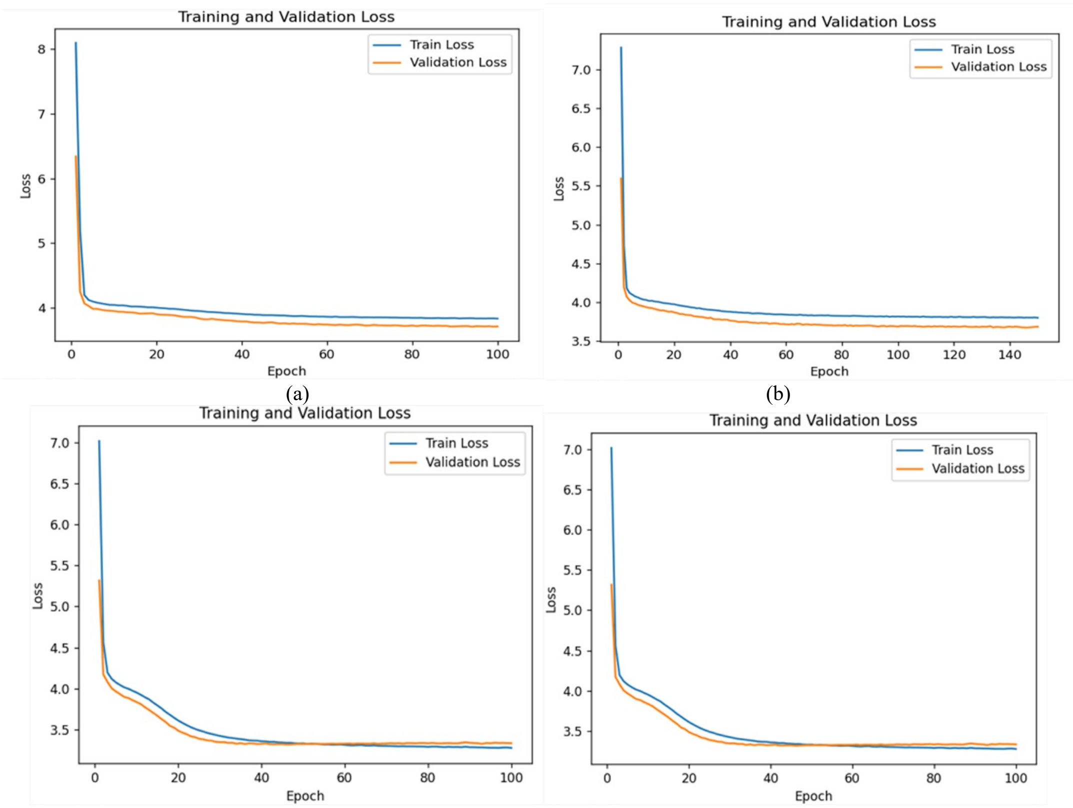
Fig. 5. Comparison of different learning rates. (a) Learning rate = 0.0001; (b) Learning rate = 0.001.

We observed that the model shows good convergence performance when the hidden size is set to 64. However, the model exhibits overfitting when the hidden size is increased to 128. Therefore, selecting the hidden size at 64 ensures that the model can avoid overfitting, enhancing its ability to generalize. Fig. 7 shows the convergence test results for different regularization parameter weight decay sizes (weight decay = 0.1 (Figs. 7(a) and 7(b)), weight decay = 0.0001 (Fig. 7(c)), weight decay = 0.001 (Fig. 7(d)) in the Adam optimizer. When the weight decay is set to 0.1, the model suffers from underfitting, and even when the number of epochs is increased, the model cannot converge. On the other hand, when the weight decay is reduced to 0.0001, although the initial convergence effect is better, the loss curve begins to diverge slightly as the training progresses. Eventually, we set the weight decay to 0.001 so that the model can maintain stable convergence while avoiding underfitting from overregularization and divergence from insufficient regularization.




(a) (b)

Fig. 6. Comparison of different hidden size = 128. (a) Hidden size = 64; (b) Hidden size = 128.




(c) (d)

Fig. 7. Comparison of different weight decays. (a) Weight decay = 0.1; (b) Weight decay = 0.1 with epoch = 150; (c) Weight decay = 0.0001; (d) Weight decay = 0.001.

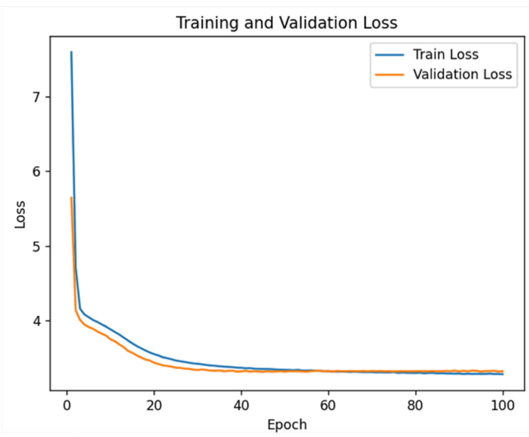


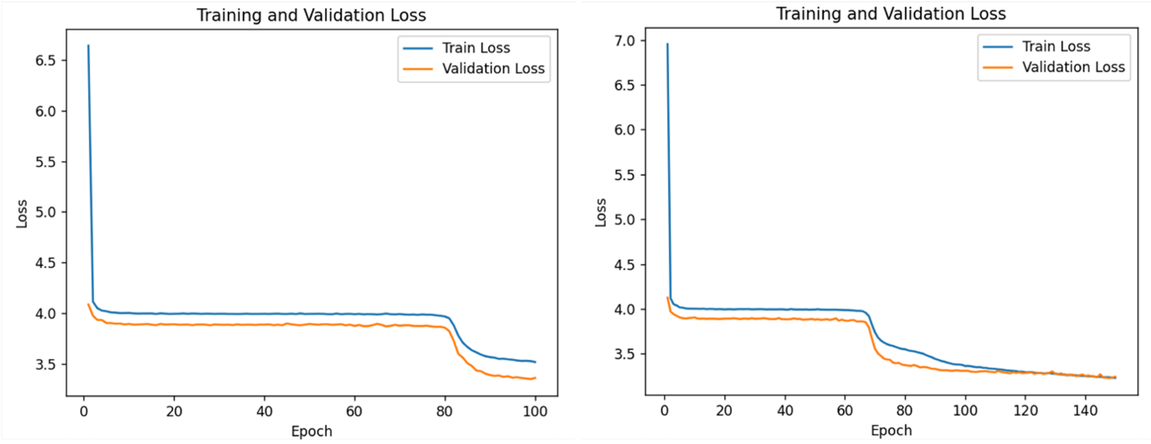

Fig. 8. Loss graph.

Next, we explored the impact of the number of epochs on model convergence performance. After determining the optimal settings of other parameters, it can be seen from Fig. 8 that when the number of epochs is set to 50, the model has begun to converge. So, we have tested and compared the prediction results when it was set to 50 and 100. Table VI shows the experimental results. When the number of epochs is set to 100, the model's accuracy is better than that of 50 on both the training and test sets. Therefore, we set it to 100 for better model performance.

TABLE VI. THE IMPACT OF EPOCH SIZE ON ACCURACY

| Epoch Size | Epoch = 50 (%) | Epoch = 100 (%) |
|---|---|---|
| train | 57 | 61 |
| test | 54 | 60 |

Finally, we discussed the impact of the number of GRU layers and batch size on model convergence. Experimental results show that the settings of these two parameters can achieve model convergence by increasing the number of epochs. Taking the number of GRU layers as an example, as shown in Fig. 9, with 100 training epochs (Fig. 9(a)) and 5 GRU layers, the model fails to achieve satisfactory convergence. In contrast, when increasing the number of epochs to 150 (Fig. 9(b)), the model can successfully converge, showing improved training performance.




(a) (b)

Fig. 9. Increase epoch to allow the model to converge. (a) GRU layer-5 and epoch = 100; (b) GRU layer-5 and epoch = 150.

## V. Conclusion and Future Work

This study utilizes deep learning algorithms to analyse and verify badminton match data from Lin Jun-Yi and Li Zi-Jia, successfully predicting the sequence of shots. Key features such as hitting position, shot type, and landing point were extracted and one-hot encoded to ensure consistency. The data was split into training and test sets in an 80–20 ratio, and a GRU model with Linear and ReLU activation functions was used to capture temporal dependencies and non-linear characteristics. Performance metrics, including accuracy, precision, recall, F1-Score, and T-test, demonstrated the model's effectiveness. Parameter adjustments showed improved convergence and stability. While the study is limited by its small dataset, it lays the groundwork for future research, suggesting broader applications and deeper analyses with more diverse data. Expanding the dataset to include a larger pool of players and match scenarios would enhance generalization and support more comprehensive modelling. In future work, we will investigate the direct use of video-based features, which can substantially enrich the input representation and facilitate the development of multi-modal, state-of-the-art prediction frameworks.

## Ethical Statement

The data is derived from publicly available match recordings without leakage of private information; no infringement of athletes' portrait rights or event copyrights, ensuring compliance with data usage regulations.

## Conflict of Interest

The authors declare no conflict of interest.

## Author Contributions

Ming-Hung Lin has provided dataset for this research; Luu-Ly Tran has written the final manuscript, reviewed literature, executed experiments, and made revision; Han-Yu Chen analysed the data and wrote the original draft; Chih-Chieh Chang offered the idea concept and methodology; all authors had approved the final version.

## References

[1] S. Sakurai and T. Ohtsuki, "Muscle activity and accuracy of performance of the smash stroke in badminton with reference to skill and practice," *J. Sports Sci.*, vol. 18, pp. 901–914, 2000.

[2] O. Faude, T. Meyer, F. Rosenberger, M. Fries, G. Huber, and W. Kindermann, "Physiological characteristics of badminton match play," *Eur. J. Appl. Physiol.*, vol. 100, pp. 479–485, 2007. doi: 10.1007/s00421-007-0441-8

[3] I. Ghosh, S. R. Ramamurthy, and N. Roy, "Stancescorer: A data driven approach to score badminton player," in *Proc. 2020 IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Austin, TX, USA, 2020, pp. 1–6. doi:10.1109/PerComWorkshops48775.2020.9156220

[4] T. Steels, B. van Herbruggen, J. Fontaine, T. de Pessemier, D. Plets, and E. de Poorter, "Badminton activity recognition using accelerometer data," *Sensors*, vol. 20, 4685, 2020. https://doi.org/10.3390/s20174685

[5] X. Wang and J. Li, "A badminton recognition and tracking system based on context multi-feature fusion," arXiv preprint, arXiv.2306. 2023.

[6] A. Islam and K. Chang, "Real-time AI-based informational decision-making support system utilizing dynamic text sources," *Appl. Sci.*, vol. 11, 6237, 2021. https://doi.org/10.3390/app11136237

[7] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS ONE*, vol. 12, no. 7, e0180944, 2017. https://doi.org/10.1371/journal.pone.0180944

[8] M. Wen, P. Li, L. Zhang, and Y. Chen, "Stock market trend prediction using high-order information of time Series," *IEEE Access*, vol. 7, pp. 28299–28308, 2019. doi: 10.1109/ACCESS.2019.2901842

[9] H. Xiong, J. Zhang, Y. Huang, K. Leach, and L. E. Barnes, "Daehr: A discriminant analysis framework for electronic health record data

and an application to early detection of mental health disorders," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 3, 47, May 2017. https://doi.org/10.1145/3007195

[10] W.-Y. Wang, T.-F. Chan, H.-K. Yang, C.-C. Wang, Y.-C. Fan, and W.-C. Peng, "Exploring the long short-term dependencies to infer shot influence in badminton matches," in *Proc. 2021 IEEE Int. Conf. Data Mining (ICDM)*, 2021, pp. 1397–1402.

[11] W.-Y. Wang, T.-F. Chan, W.-C. Peng, H.-K. Yang, C.-C. Wang, and Y.-C. Fan, "How is the stroke? Inferring shot influence in badminton matches via long short-term dependencies," *ACM Trans. Intell. Syst. Technol.*, vol. 14, no. 1, 7, Feb. 2023. https://doi.org/10.1145/3551391

[12] W.-Y. Wang, H.-H. Shuai, K.-S. Chang, and W.-C. Peng, "ShuttleNet: Position-aware fusion of rally progress and player styles for stroke forecasting in badminton," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 4, 2022, pp. 4219–4227. https://doi.org/10.1609/aaai.v36i4.20341.

[13] K.-S. Chang, W.-Y. Wang, and W.-C. Peng, "Where will players move next? Dynamic graphs and hierarchical fusion for movement forecasting in badminton," in *Proc. Thirty-Seventh AAAI Conf. Artif. Intell. Thirty-Fifth Conf. Innov. Appl. Artif. Intell. Thirteenth Symp. Educ. Adv. Artif. Intell. (AAAI'23/IAAI'23/EAAI'23)*, AAAI Press, 2023, vol. 37, pp. 6998–7005. https://doi.org/10.1609/aaai.v37i6.25855

[14] M. Lewis, *Moneyball: The Art of Winning an Unfair Game*, New York, NY, USA: W. W. Norton, 2003.

[15] D. Oliver, *Basketball on Paper: Rules and Tools for Performance Analysis*, Dulles, VA, USA: Brassey's, Incorporated, 2004.

[16] N. F. N. Bittencourt, W. H. Meeuwisse, L. D. Mendonça, A. Nettel-Aguirre, J. M. Ocarino, and S. T. Fonseca, "Complex systems approach for sports injuries: Moving from risk factor identification to injury pattern recognition—Narrative review and new concept," *Br. J. Sports Med.*, vol. 50, no. 21, pp. 1309–1314, 2016. https://doi.org/10.1136/bjsports-2015-095850

[17] A. G. Schubert, J. Kempf, and B. C. Heiderscheit, "Influence of stride frequency and length on running mechanics," *Sports Health Multidiscip. Approach*, vol. 6, pp. 210–217, 2013. doi: 10.1177/1941738113508544

[18] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN Encoder–decoder for statistical machine translation," in *Proc. 2014 Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, Doha, Qatar, 2014, pp. 1724–1734.

[19] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint, arXiv.1412.3555, 2014.

[20] D. Peralta, B. Van Herbruggen, J. Fontaine, W. Debyser, J. Wieme, and E. de Poorter, "Badminton stroke classification based on accelerometer data: from individual to generalized models," in *Proc. 2022 IEEE Int. Conf. Big Data (Big Data)*, Osaka, Japan, 2022, pp. 5542–5548. doi: 10.1109/BigData55660.2022.10020984

[21] W. Gawin, C. Beyer, and M. Seidler, "A competition analysis of the single and double disciplines in world-class badminton," *Int. J. Perform. Anal. Sport*, vol. 15, no. 3, pp. 997–1006, 2015. http://dx.doi.org/10.1080/24748668.2015.11868846

[22] K.-D. Wang, W.-Y. Wang, P.-C. Hsieh, and W.-C. Peng, "Offline imitation of badminton player behavior via experiential contexts and brownian motion," *CoRR*, abs/2403.12406, 2024.