

SPADE-LSTM: An Integrated Sequential Pattern Mining and Deep Learning for Badminton Next-Stroke Prediction

Jefita Resti Sari^{1*}, Sachnaz Desta Oktarina², Erfiani³

^{1,2,3}Master Program in Statistics and Data Science, School of Data Science, Mathematics and Informatics, IPB University

Article Info

Article history:

Received Jan 05, 2026

Revised Jan 27, 2026

Accepted Feb 14, 2026

Keywords:

Badminton

Stroke Pattern

Prediction

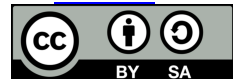
SPADE

LSTM

ABSTRACT

Badminton rallies consist of complex and rapid stroke transitions that reflect players' tactical decision-making. While prior studies have examined stroke patterns descriptively or applied standalone predictive models, limited research integrates interpretable sequential pattern mining with deep learning for next-stroke prediction. This study proposes an integrated SPADE-LSTM framework to analyze and predict badminton stroke sequences using a 10-class scheme (drive, dropshot, lob, netting, and smash for two athletes). Match data were transformed into structured stroke sequences and contextual features, then divided into training, validation, and test sets using a match-set-rally grouping strategy to prevent information leakage. Sequential patterns were first extracted using the Sequential Pattern Discovery using Equivalent Classes (SPADE) algorithm to capture frequent tactical transitions. These pattern-based features were subsequently used to train a Long Short-Term Memory (LSTM) model for multi-class classification. The proposed model achieved an accuracy of 88.68%, with weighted precision, recall, and F1-score of 0.9075, 0.8868, and 0.8851, respectively. Misclassifications were mainly observed in tactically similar stroke transitions and minority classes. The results indicate that integrating interpretable sequential pattern mining with deep learning provides both strong predictive performance and meaningful tactical insights for badminton performance analysis.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Jefita Resti Sari,

Master Program in Statistics and Data Science,

IPB University,

Jl. Raya Dramaga, Kampus IPB Dramaga, Bogor, 16680, West Java, Indonesia.

Email: jefitarestisari@apps.ipb.ac.id

1. INTRODUCTION

Badminton is a sport played with a shuttlecock and is very popular in Indonesia. The game can be played as singles or doubles, with a scoring system where the player or pair who first reaches 21 points is declared the winner [1]. Badminton requires speed, physical endurance, game strategy, and quick decision-making when facing opponents. During a match, players perform a series of repeated strokes that form specific patterns, so that badminton games can naturally be modeled as sequential data. These patterns reflect the player's strategy, response to their opponent, and adaptation to the situation in the match. Previous studies such as [2], [3], have analyzed the factors that determine victory and defeat in badminton, including biomechanical studies that affect athlete performance during matches. Each rally represents a player's tactical decisions, such as attack patterns, tempo changes, and adjustments to game conditions. Therefore, the ability to analyze attack patterns is important for coaches and performance analysts in understanding the structure of the game and designing more effective strategies. However, the analysis process, which

still relies heavily on manual observation of match recordings, tends to be time-consuming, prone to subjectivity, and less capable of identifying consistently recurring sequential patterns [2].

In addition, notational analysis studies show that game characteristics, such as rally duration, number of strokes per rally, and game time structure, can differ between levels or periods. This indicates that sequence-based analysis is necessary so that strategy dynamics are understood not only qualitatively, but also quantitatively and systematically [4]. Changes in competition format can also be accompanied by differences in temporal structure, the application of techniques and tactics, and point-scoring patterns. Therefore, the use of stroke sequences as a unit of analysis is important for a more in-depth study of game strategy [5]. In terms of performance demands, physiological research on trained junior badminton players report that physiological and metabolic responses are closely related to rally characteristics, particularly rally duration and game intensity. These findings further emphasize that sequential analysis plays an important role in comprehensively understanding the dynamics of badminton [6].

In line with developments in data analysis, various computational methods have been introduced to understand game structures through strike sequence data. One relevant approach is sequential pattern mining, which is a pattern mining technique that aims to find sequences of events that frequently appear in data [7]. Sequential Pattern Discovery using Equivalent Classes (SPADE) is an effective sequential pattern mining technique for identifying patterns that frequently appear in data that is highly sensitive to time sequences [8]. SPADE has been applied in various domains, such as transaction analysis [9], movie recommendations system [10], and medical records [11]. Although effective in finding frequently occurring patterns of hits, the study was still limited to two specific matches and did not develop a predictive model to project the next hit. The limitation of the SPADE method is that it can only identify dominant sequence patterns based on frequency, without considering the possibility of future predictions because SPADE's output focuses on "frequent patterns." This approach still needs to be combined with a predictive model so that the patterns found can be used to predict the next shot in the context of the ongoing rally. To overcome this, another method with predictive capabilities is needed.

Sequential rules are performed or extracted first from the entire corpus of strike sequences and treated as a global knowledge base. To overcome the shortcomings of SPADE, which can only identify dominant sequence patterns based on frequency without considering future predictions, another method with predictive capabilities is needed. In the context of sequential data, the Long Short-Term Memory (LSTM)-based machine learning approach can be used to predict the next shot of a badminton player. LSTM is a deep learning model designed to handle sequential data so that it can remember longer time steps [12], such as the sequence of shots in badminton. The LSTM model is capable of learning the relationship between strokes in each rally and predicting the next stroke based on the pattern of previous strokes. The LSTM approach has been widely applied in sports such as tennis [13], and soccer [14], for sequence-based tactical and strategic analysis [15]. Additionally, recent research on stroke prediction in badminton also focuses on sequence modeling to predict the next stroke by utilizing the context of the rally, point, match, and stroke sequence. It also shows that selecting the right context/features is important for more stable and explainable predictions.

Although this study does not incorporate explicit positional data of players or the shuttlecock, previous studies have shown that stroke sequences alone contain rich implicit tactical information, including attack patterns, tempo variations, and responses to opponents. By modeling these sequential dependencies, the SPADE-LSTM framework can effectively capture tactical behaviors without relying on explicit spatial coordinates [16], [17]. The absence of positional data is acknowledged as a limitation however, the sequential representation combined with contextual match features is sufficiently informative to model next-stroke behavior and uncover recurring tactical patterns.

Based on this background, this study positions the integration between SPADE and LSTM as an effort to bridge interpretable pattern analysis with sequential learning-based prediction capabilities. This integration aims to utilize the strength of SPADE in summarizing frequently occurring strategy patterns while utilizing the capabilities of LSTM in modeling temporal dependencies for predicting the next stroke. Thus, the main objectives of this study are to develop

an analysis flow that: (1) produces dominant sequential patterns using SPADE as a representation of repetitive game strategies, and (2) trains the LSTM model to predict the next shot into 10 classes based on the previous shot sequence of two athletes. Practically, the results of this study are expected to assist coaches and performance analysts in identifying frequent rally patterns and anticipating opponents' stroke tendencies to support training planning and match preparation. Scientifically, this study enriches the integrative study between interpretable sequential pattern mining and sequence-based prediction modeling in multi-class problems in badminton.

2. RESEARCH METHOD

This study uses an integrated experimental design to predict the next shot in a badminton rally in a 10-class classification scheme. The analysis procedure was carried out sequentially, starting from the collection of observational data, sequence pattern mining using SPADE, rule/pattern-based feature formation, LSTM training, to evaluation. A summary of the research workflow is shown in Figure 1, which combines two main blocks: the SPADE stage (pattern mining) and the LSTM stage (predictive modeling).

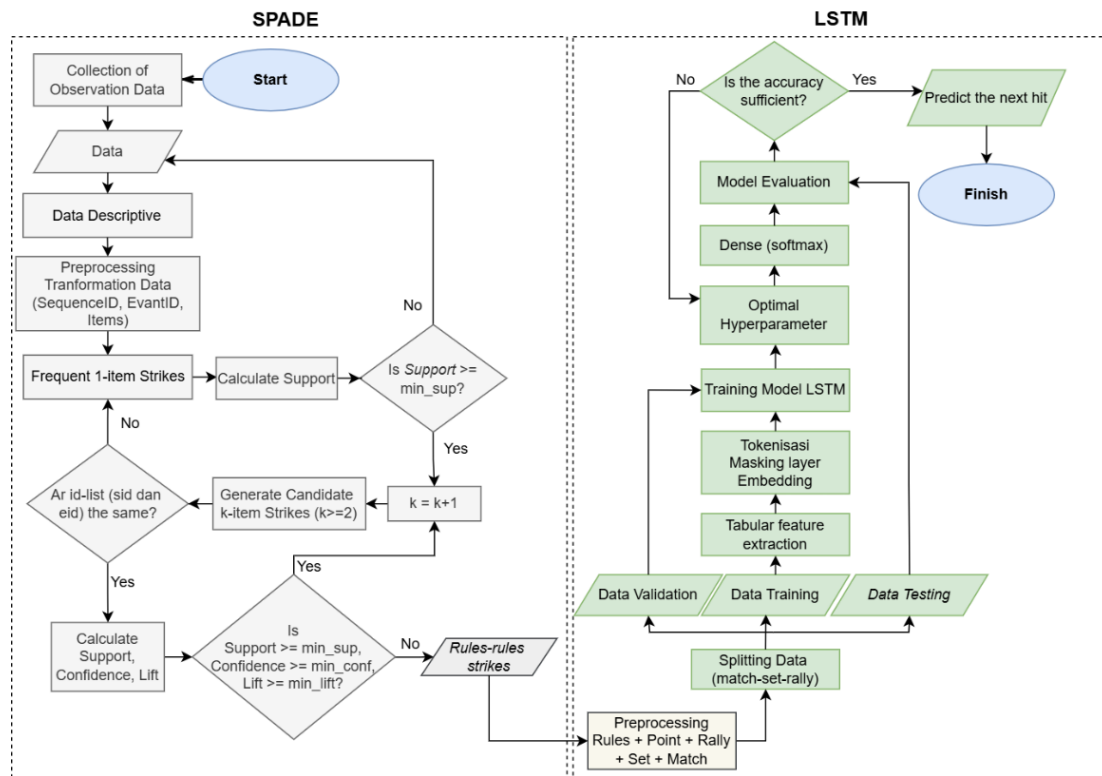


Figure 1. Flowchart of data analysis procedure

2.1. Dataset

The data in this study was obtained from two badminton matches in the World Championships organized by the Badminton World Federation (BWF) and taken from the official BWF YouTube channel. The match videos are public and were used as a source of observation. The annotation process was carried out manually by the author based on the definitions and classifications of shot types that had been studied and formulated previously. To minimize annotation errors, the annotation results were then systematically reviewed with the help of the GPT-4o model as a validation support tool, not as a substitute for human annotators. GPT-4o was used to detect potential label inconsistencies and assist in the annotation alignment process, while the final decision remained with the author. The units of analysis in this study include athlete names, points, rallies, stroke sequences, stroke types, athlete positions in the match set, and match information. The dataset used is the result of annotation and further processing of the video source.

Each rally is represented as a discrete sequence $S = \{s_1, s_2, \dots, s_N\}$ with S_t being the label of the stroke type in to- t step (10 classes). The analysis was conducted using Rstudio to process the SPADE analysis and Python to perform predictive analysis using the LSTM model.

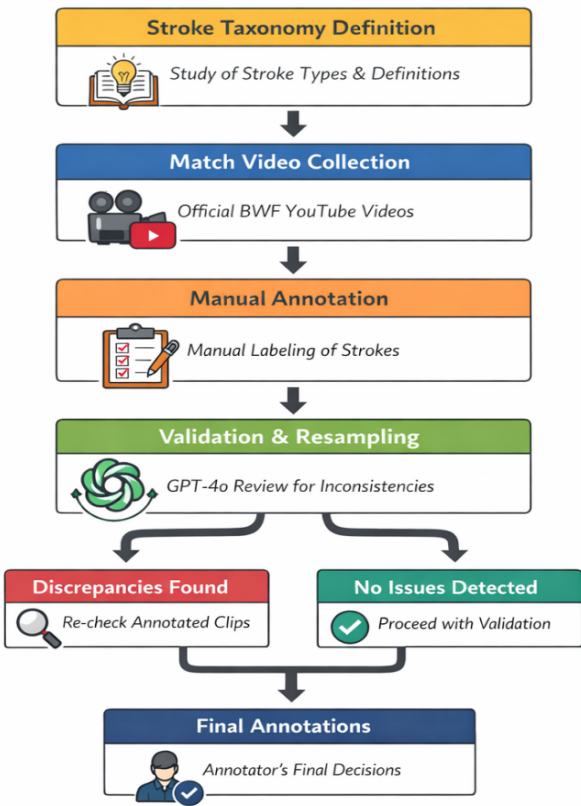


Figure 2. Observation and annotation process

Table 1. Match profile and descriptive Toyota Thailand Open 2025					
Match	TOYOTA Thailand Open 2025				
Date	May 18, 2025				
Player A	Anders Antonsen (DEN)				
Player K	Kunlavut Vitidsarn (THA)				
Score	17-21 21-17 9-21				
Parameters	Percent	Antonsen (DEN)	Stats	Vitidsarn (THA)	Percent
	Service				
	73%	36	Short	51	89%
	27%	13	Flick	5	9%
	0%	0	Errors	1	2%
	0%	0	Faults	0	0%
	Points breakdown				
	43%	21	Winners	21	37%
	27%	13	Unforced Errors	17	30%
	31%	15	Forced Errors	19	33%

Table 2. Match profile and descriptive BWF World Championship Open 2025					
Match	BWF World Championships 2025				
Date	August 31, 2025				
Player A	Anders Antonsen (DEN)				
Player K	Kunlavut Vitidsarn (THA)				
Score	17-21 15-21				
Parameters	Percent	Antonsen (DEN)	Stats	Vitidsarn (THA)	Percent
	Service				
	88%	28	Short	36	88%
	9%	3	Flick	5	12%
	3%	1	Errors	0	0%
	0%	0	Faults	0	0%
	Points breakdown				
	34%	11	Winners	16	39%
	31%	10	Unforced Errors	15	37%
	34%	11	Forced Errors	10	24%

Table 1 and Table 2 summarize the match profiles and descriptive statistics of the two matches analyzed, namely the TOYOTA Thailand Open 2025 and the BWF World Championships 2025, featuring players Anders Antonsen (DEN) vs Kunlavut Vitidsarn (THA). The information displayed includes the match date, final score, and a summary of serve statistics (short, flick,

error/fault) and points breakdown, which divides the points earned into winners, unforced errors, and forced errors for each player. In general, these tables are used to show the context of the match and the differences in playing styles before moving on to stroke sequence analysis and modeling.

Table 3. Data match

No	Anders Point	Kunlavut Point	Name	Rally	Type	Number of Shot	Set	Match
1	0	0	Kunlavit Vitidsarn	1	Netting(K)	1	1	Toyota Thailand Open
2	0	0	Anders Antonsen	1	Netting(A)	2	1	Toyota Thailand Open
3	0	0	Kunlavit Vitidsarn	1	Lob(K)	3	1	Toyota Thailand Open
4	0	0	Anders Antonsen	1	Dropshot(A)	4	1	Toyota Thailand Open
:	:	:	:	:	:	:	:	:
2206	15	20	Anders Antonsen	72	Lob(A)	1	2	Paris BWF Championship
2207	15	20	Anders Antonsen	73	Netting(A)	1	2	Paris BWF Championship
2208	15	21	Kunlavit Vitidsarn	73	Lob(K)	2	2	Paris BWF Championship

Table 3 is a combination of the two matches from the 2025 TOYOTA Thailand Open and the 2025 Paris BWF Championship. In the 2025 TOYOTA Thailand Open match, there were 1,457 rows of shot data and 105 rallies with 3 sets of play, while in the 2025 Paris BWF Championship, there were 751 rows of shots and 73 rallies with 2 sets of play. The total number of stroke data rows from both matches is 2,208 stroke sequences across the two matches from the 2025 TOYOTA Thailand Open and the Paris BWF Championship.

2.2. Preprocessing SPADE

This stage prepares the data in sequential pattern mining format. Data variables will be selected and transformed as follows.

Table 4. Trasformation Data SPADE Algorithm

Variable	Transformation
Rally	SequenceID
Number of Shot	EventID
Type	Item

The data is then processed in the following steps: (1) sorting events based on eventID, (2) removing duplicates or invalid events, (3) normalizing the database sequence $S = \{s_1, s_2, \dots, s_N\}$ which becomes the input for SPADE. The output of this stage is a set of clean sequences for the mining process.

2.3. Sequential Pattern with SPADE Algorithm

Sequential Patterns Discovery using Equivalent Classes (SPADE) was first introduced in the journal “SPADE: An Efficient Algorithm for Mining Frequent Sequences” by Mohammed J. Zaki. SPADE is a pattern mining algorithm that uses vertical data format in databases. In vertical data format, the sequence database forms a sequence collection with the format [itemset: (sequence_ID, eventID)] [18].

SPADE ($min_sup, \mathcal{D}$):
 $\mathcal{F}_1 = \{ \text{frequent items or 1-sequences} \};$
 $\mathcal{F}_2 = \{ \text{frequent 2-sequences} \};$
 $\mathcal{E} = \{ \text{equivalence classes } [X]_{\theta_1} \};$
for all $[X] \in \mathcal{E}$ **do** *Enumerate-Frequent-Seq*($[X]$);

Figure 3. SPADE algorithm [8]

The SPADE algorithm for finding frequent sequences and forming rules can be summarized as follows [8].

1. Frequent 1-sequence: scan the sequence database for each item, form an id-list (pair of sid = sequenceID and eid = eventID), calculate support based on the number of unique sid, then save items with support $\geq \text{min_sup}$.
2. Frequent 2-sequence: from the frequent 1-sequence, join item pairs to form candidates (A,B), $A \rightarrow B$, or $B \rightarrow A$ based on the eid relation in the same sid, then calculate the support and save candidates with support $\geq \text{min_sup}$.
3. Frequent k-sequence: for $k \geq 3$, join frequent (k-1) sequences that have the same prefix, calculate support, save those that meet min_sup, and stop when there are no new candidates that pass.
4. Rule formation: from frequent sequences (length ≥ 2), antecedent = all items except the last and consequent = last item calculate confidence and lift, rule accepted if confidence $\geq \text{min_conf}$ and lift is greater than or equal to 1, indicating that the rules are strong.

For sequential pattern extraction [19] n “Mining Frequent Sequential Patterns” compares the GSP and SPADE algorithms, showing that SPADE is more efficient on long sequence databases. Meanwhile, [20] has applied sequential pattern mining to sports event data in “Supervised Sequential Pattern Mining of Event Sequences in Sport to Identify Important Patterns of Play: An Application to Rugby Union,” which is relevant as an example of application in the sports domain.

2.4. Splitting Data

Preprocessing for LSTM is done by forming a final dataset that combines stroke sequence information and match context features. First, the results of sequential pattern mining (rules/patterns from SPADE) are combined with context variables for points, rallies, sets, and matches so that each sample has pattern information (rules) as well as the game conditions at that time (Preprocessing: Rules + Point + Rally + Set + Match).

Before modeling, the data was prepared to be suitable for LSTM with the target being the last shot in a 10-class scheme. The final dataset was formed from a combination of shot sequences, SPADE rule features, and point-rally-set-match context features. Next, the data is separated into three subsets, namely training data (70%), validation data (10%), and test data (20%) using a group-based separation scheme (match-set-rally), so that one rally is not spread across different subsets and the risk of information leakage can be minimized, thereby preventing one rally from being spread across different subsets.

It is important to note that SPADE is an unsupervised sequential pattern mining method, and the support and confidence values represent global transition tendencies of stroke patterns, not label-specific information from the test set. These values are calculated as aggregated statistics from the overall pattern distributions in the corpus, and therefore do not encode information about the actual labels in the test set, meaning that no instance-level data leakage occurs. Performing SPADE prior to the data split ensures that all patterns are represented in the feature space; splitting data before rule mining could result in missing patterns in the test set and a loss of feature representation, which empirically reduces predictive accuracy rather than creating leakage. For methodological transparency, future work may explore temporal-based or match-based splitting strategies to further evaluate generalization on truly unseen distributions. This issue is also discussed in detail in the Discussion section and is included as a suggestion for improvement in future studies.

From the training data, tokenization was performed to form a vocabulary, with PAD as the filler token. To prevent target leakage through the sequence of strokes, the sequence from the last stroke position to the end was masked with PAD, so that the model only learned from the context before the target.

In addition to the sequence, a tabular feature was also formed consisting of the numeric variables Set, Rally, Support, and Confidence, as well as the categorical variable Match. Missing numeric values were imputed using the median in the training data, then standardized using StandardScaler installed on the training data and applied consistently to the validation data and test

data. The Match categorical variable was encoded using LabelEncoder, which was also applied to the training data so that the category indices were uniform across all subsets. Thus, three main model inputs were obtained, namely sequence input, numerical input, and categorical input. The training data is used to learn the model weights, while the validation data is used as an internal evaluation to select the best configuration (including hyperparameters and early stopping when performance does not improve), and the test data is kept separate to objectively measure the model's generalization ability at the final stage.

2.5. Long Short-Term Memory (LSTM) Models

Long Short-Term Memory (LSTM) is an artificial neural network specifically developed to handle sequential data, such as text or time series data, which is a derivative of recurrent neural networks (RNN) [21], [22]. LSTM uses a gating mechanism to learn long-term dependencies [23]. LSTM utilizes several switch gates that enable it to pass through certain units, allowing it to remember information over a longer period of time [24]. The LSTM architecture consists of memory units (cells) that receive current input and previous status as input [25], [26]. These units determine what information needs to be stored or discarded, then combine the current input with the previous status to generate the next input. This process allows LSTM to effectively capture long-term dependencies [27].

The LSTM model consists of three main components called gates: (1) Input Gate i_t : Determines what new information will be stored in memory. At this stage, the input gate processes the incoming data and selects whether the information is relevant to be stored or not. (2) Forget Gate f_t : Determines which information from the previous status C_{t-1} needs to be forgotten. (3) Output Gate o_t : Generates output based on the updated memory cell status. The following are the function equations of the LSTM Model [28].

$$i_t = \sigma(W_i \cdot [h_{t-1}, Y_t] + b_i) \quad (1)$$

$$\bar{C}_t = \tanh(W_c Y_t + W_c h_{t-1} + b_c) \quad (2)$$

$$f_t = \sigma(W_f Y_t + W_f h_{t-1} + b_f) \quad (3)$$

$$C_t = i_t * \bar{C}_t + f_t * C_{t-1} \quad (4)$$

$$out_t = \sigma(W_o Y_t + W_o h_{t-1} + b_o) \quad (5)$$

$$h_t = out_t * \tanh(C_t) \quad (6)$$

$$\sigma = \frac{1}{1 + e^{-x}} \quad (7)$$

$$\tanh(x) = 2\sigma(2x) - 1 \quad (8)$$

Research on the topic of sports [14], has used LSTM to predict player movements, game strategies, and scoring patterns. Research by [12], [23] shows that LSTM is effective in capturing long-term relationships from sequential data such as strokes in badminton matches. LSTM is applied to predict the next stroke of a badminton player based on the sequence of previous strokes in each rally. In classifying the next shot (10 classes), the distribution of shot labels in the rally sequence data tends to be imbalanced, so the model risks being biased towards the majority class and ignoring the rare classes. This condition can reduce the quality of predictions in minority classes because the algorithm learns from the most frequently occurring patterns [29], [30].

To mitigate this issue without altering the natural distribution of match data, a class-weighted categorical cross-entropy loss function was applied during model training. Higher weights were assigned to minority stroke classes to penalize their misclassification more strongly. This approach allows the model to improve minority-class learning while preserving the authentic tactical structure of the game [31]. Unlike aggressive resampling strategies, the weighted loss method maintains the original sequence characteristics, which is crucial in sports analytics where class imbalance reflects real gameplay dynamics. In the context of sports analytics, imbalanced distributions inherently represent real player strategies; therefore, preserving the original distribution is important to maintain the tactical validity of the model [32].

2.6. Model Evaluation

Modeling was performed using Long Short-Term Memory (LSTM) architecture to predict the next shot in a badminton game. The next step was to evaluate the model's performance. This evaluation was carried out by calculating various performance metrics, including accuracy, which is the proportion of correct predictions to the total number of predictions made by the model; precision, which assesses the accuracy of the model in predicting each specific type of shot; recall, which assesses the model's ability to detect all relevant shots; and F1-Score, which represents the balance between precision and recall. The equation can be written as follows [33].

$$precision = \frac{TP}{TP + FP} \quad (11)$$

$$recall = \frac{TP}{TP + FN} \quad (12)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (13)$$

$$accuracy = \frac{TP + TN + FP + FN}{TP + TN + FP + FN} \quad (14)$$

In addition, confusion matrix analysis is also used to assess the performance of classification models by comparing actual classes (actual labels) and predicted classes (model results). In multi-class classification, i.e., 10 classes, the confusion matrix takes the form of a K×K matrix, where the rows represent actual classes and the columns represent predicted classes. The elements on the main diagonal indicate the number of correct predictions (actual class is the same as the predicted class), while the elements outside the diagonal indicate classification errors when the model predicts a different class. This visualization facilitates the identification of frequently confused classes and forms the basis for calculating evaluation metrics such as precision, recall, and F1-Score, both per class and in aggregate [34], [35]. The matrix configuration for 10 classes in this study can be seen in Figure 4 below.

		Predicted Class									
		Drive (A)	Drive (K)	Dropshot (A)	Dropshot (K)	Lob (A)	Lob (K)	Netting (A)	Netting (K)	Smash (A)	Smash (K)
Actual Class	Drive (A)	True	False	False	False	False	False	False	False	False	False
	Drive (K)	False	True	False	False	False	False	False	False	False	False
	Dropshot (A)	False	False	True	False	False	False	False	False	False	False
	Dropshot (K)	False	False	False	True	False	False	False	False	False	False
	Lob (A)	False	False	False	False	True	False	False	False	False	False
	Lob (K)	False	False	False	False	False	True	False	False	False	False
	Netting (A)	False	False	False	False	False	False	True	False	False	False
	Netting (K)	False	False	False	False	False	False	False	True	False	False
	Smash (A)	False	False	False	False	False	False	False	False	True	False
	Smash (K)	False	False	False	False	False	False	False	False	False	True

Figure 4. Multi-class confusion matrix scheme 10 classes

3. RESULTS AND DISCUSSION

This chapter presents the research results in sequence, starting from a summary of the dataset characteristics, preprocessing, pattern mining results using SPADE, feature formation for LSTM, hyperparameter tuning results, to model performance evaluation in predicting the next stroke in a 10-class scheme.

3.1. Data Descriptive

In the descriptive data, a brief description of the characteristics of the stroke sequence data used is provided, including the number of observations in each tournament, the definition of 10 classes of strokes distinguished by athlete, namely athlete Anders Antonsen (A) and athlete Kunlavut Vitidsarn (K), and the frequency/proportion distribution of each class. This description is important for observing general game pattern trends, such as dominant strokes and strokes that

rarely appear, while also identifying potential class imbalances that could affect the training of the next stroke prediction model.

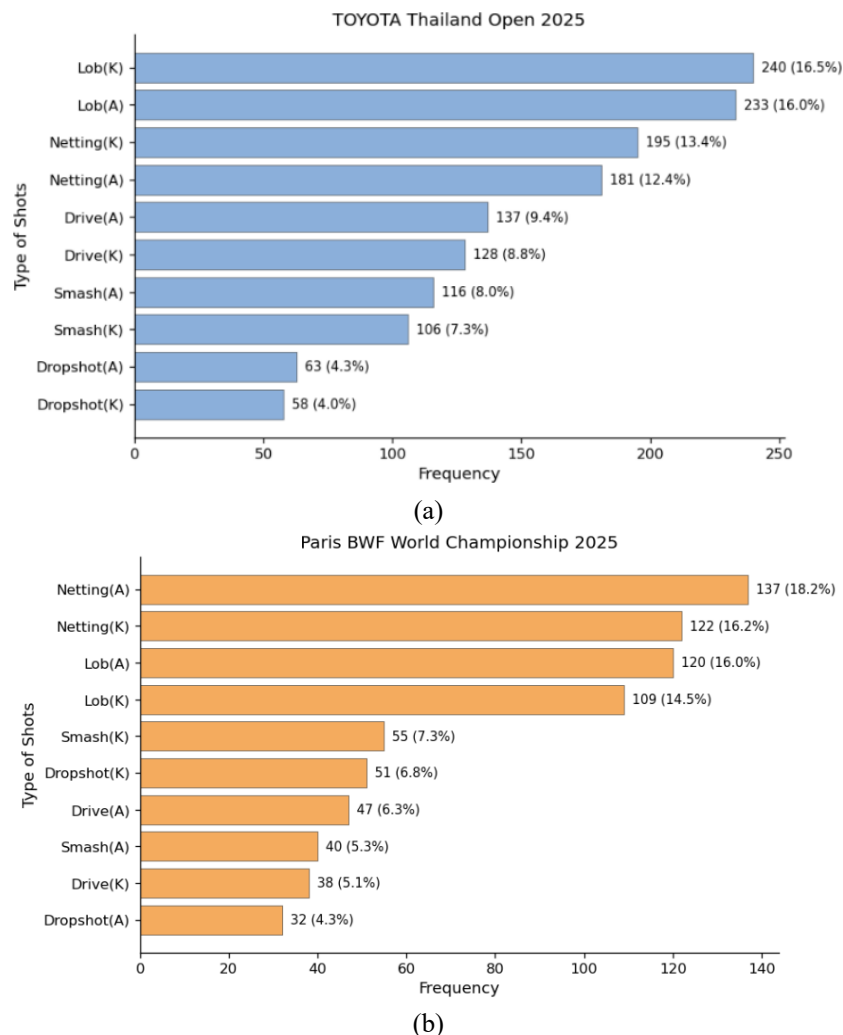


Figure 5. Distribution of rally lengths from the TOYOTA Thailand Open and Paris BWF Championship matches

Based on the visualization in Figure 5 TOYOTA Thailand Open 2025 (total 1457 strokes), the most dominant class is Lob (K) with 240 (16.5%) and Lob (A) with 233 (16.0%), indicating that lob is the most frequent stroke pattern for both players. The next highest strokes are Netting (K) 195 (13.4%) and Netting (A) 181 (12.4%), indicating that rallies are also frequently conducted near the net. Meanwhile, Dropshot is the class with the lowest frequency, namely Dropshot (A) 63 (4.3%) and Dropshot (K) 58 (4.0%), indicating that this shot is relatively less frequently performed compared to other shot types in the TOYOTA Thailand Open 2025 match.

In the visualization of the 2025 BWF World Championships in Paris (total of 751 shots), the dominant pattern is slightly different, with Netting being the highest. Netting (A) 137 (18.2%) and Netting (K) 122 (16.2%). Following that, Lob (A) 120 (16.0%) and Lob (K) 109 (14.5%) remain at the top, so rallies are often formed from a combination of net play and lob. For other strokes, the frequency is lower, with Drive (K) at 38 (5.1%), and the lowest being Dropshot (A) at 32 (4.3%). Overall, compared to the Thailand Open, the distribution at the Paris BWF shows a stronger emphasis on net play, while the drive/smash category tends to be smaller, leading to rallies dominated by control battles and transitions in the front court area.

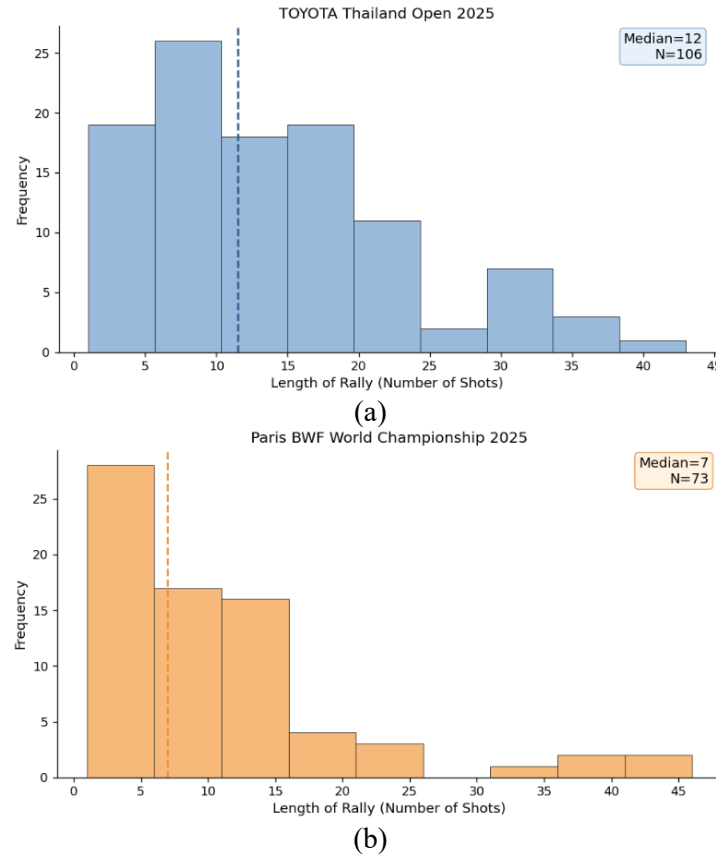


Figure 6. Distribution of shot types from the TOYOTA Thailand Open and Paris BWF Championship matches

Next, Figure 6 shows the distribution of rally length in terms of the number of strokes per rally in both matches of the 2025 TOYOTA Thailand Open. A total of 106 rallies were analyzed, with a median of 12 strokes, meaning that half of the rallies observed lasted up to 12 strokes or less. The distribution of rally length was quite diverse, with most rallies falling in the short to medium range of around 1–20 strokes, but there were still several very long rallies of up to nearly 40 strokes. This pattern shows that the matches in the tournament were not only dominated by short rallies, but that longer rallies also occurred quite frequently, although not in large numbers.

At the 2025 Paris BWF World Championship, the number of rallies analyzed was 73 with a median of 7 strokes, so in general the rallies tended to be shorter than at the Thailand Open. The majority of rallies were in the range of 1–10 strokes, but there were still some long rallies of up to about 45 strokes. Comparatively, the lower median value in Paris indicates that rallies ended more quickly, while the Thailand Open showed that medium-length rallies occurred more frequently. This variation in rally length is important for the modeling stage because the non-uniform sequence of strokes needs to be handled by standardizing the length of the input (e.g., through padding) so that it can be processed consistently by the model.

3.2. Data Preprocessing

During the data preprocessing stage, raw strike data, which was originally in the form of records per incident, was converted to a format suitable for sequence pattern mining using SPADE.

Table 5. Data format of sequence pattern mining with SPADE algorithm

No	SequenceID	Item	EventID	Size	Match
1	1	Netting(K)	1	1	Toyota Thailand Open
2	1	Netting(A)	2	1	Toyota Thailand Open
3	1	Lob(K)	3	1	Toyota Thailand Open
4	1	Dropshot(A)	4	1	Toyota Thailand Open

⋮	⋮	⋮	⋮	⋮	⋮
2206	72	Lob(A)	1	1	Paris BWF Championship
2207	73	Netting(A)	1	1	Paris BWF Championship
2208	73	Lob(K)	2	1	Paris BWF Championship

The data format in Table 5 represents one stroke event in a rally per row. The SequenceID column is used as a rally identifier, so all rows with the same SequenceID are treated as one stroke sequence. The Item column contains the type of stroke and the player who performed the stroke, for example, Netting(K) or Lob(A), so that the stroke class remains consistent across all observations. The EvantID column shows the sequence of events in the rally (1, 2, 3, and so on) so that the chronological order is maintained. The Size column has a value of 1 because each event only contains one type of shot. Meanwhile, the Match column is included to mark the source of the match (e.g., Toyota Thailand Open and Paris BWF Championship), so that data from several matches can be combined but their origins can still be traced. With this arrangement, the structure of the stroke sequence per rally has been formed through the SequenceID–EvantID pair, so that the data is ready to be used in the sequential pattern extraction stage using SPADE.

3.3. Sequential Pattern with SPADE Algorithm Result

The SPADE algorithm is used to extract sequential patterns from match data that has been arranged in the form of a series of strokes per rally. The analysis is performed on the sequence of strokes after the serve so that the patterns formed represent the rally phase. SPADE is run with a minimum support value of 0.05 and a minimum confidence of 0.3.

The processing and analysis performed using the SPADE algorithm yielded the stroke rules of Anders Antonsen and Kunlavut Vitidsarn, as shown in Tables 4 and 5 below.

Table 6. Results of the SPADE algorithm rules for the TOYOTA Thailand Open

No	rule	support	confidence	lift
1	<{Netting(K)}, {Drive(A)}, {Lob(K)}> => <{Dropshot(A)}>	0.055	1	3.042
2	<{Dropshot(K)}, {Netting(A)}, {Drive(K)}, {Drive(A)}> => <{Drive(K)}>	0.055	1	2.92
3	<{Dropshot(K)}, {Netting(A)}, {Lob(K)}, {Lob(A)}, {Lob(K)}> => <{Dropshot(A)}>	0.84	1	2.535
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
2508	<{Dropshot(K)}, {Lob(A)}, {Lob(K)}> => <{Drive(A)}>	0.124	0.619	1
2509	<{Lob(A)}, {Dropshot(K)}, {Lob(A)}, {Lob(K)}> => <{Drive(A)}>	0.124	0.619	1
2510	<{Smash(K)}, {Netting(A)}, {Netting(K)}, {Dropshot(A)}> => <{Lob(K)}>	0.124	0.867	1

Table 6 shows the attack patterns formed by Anders Antonsen (A) and Kunlavut Vitidsarn (K) at the 2025 TOYOTA Thailand Open tournament. These rules were formed using a minimum support of 0.05 and a minimum confidence of 0.3, resulting in 2510 rules. The rule with the highest probability is the rule <{Netting(K)}, {Drive(A)}, {Lob(K)}> => <{Dropshot(A)}>, which explains that the rule has support = 0.05, confidence = 1, and lift ratio = 3.042. This means that the probability of Kunlavut Vitidsarn hitting a net shot followed by a drive shot by Anders Antonsen, and a lob shot by Kunlavut Vitidsarn resulting in a dropshot by Anders Antonsen from all rallies during the match, is 5.5%. Where the probability of Kunlavut Vitidsarn performing a lob shot if a sequence of netting shots by Kunlavut Vitidsarn and a drive shot by Anders Antonsen and a lob by Kunlavut Vitidsarn had previously occurred is 1 with a rule strength of 3.042.

Table 7. Hasil rules SPADE algoritma Paris BWF Championship Open

No	rule	support	confidence	lift
1	<{Lob(A)}, {Netting(K)}, {Lob(A)}, {Dropshot(K)}, {Netting(A)}, {Lob(K)}, {Lob(A)}, {Lob(K)}> => <{Dropshot(A)}>	0.055	1	3.042
2	<{Smash(A)}, {Netting(K)}, {Drive(A)}> => <{Drive(K)}>	0.055	1	2.92
3	<{Dropshot(K)}, {Netting(A)}, {Netting(K)}, {Lob(A)}> => <{Dropshot(K)}>	0.151	0.786	1.978

⋮	⋮	⋮	⋮	⋮
14711	<{Lob(A)}, {Lob(K)}, {Drive(A)}> => <{Netting(K)}>	0.068	0.71	1
14712	<{Lob(K)}, {Smash(A)}, {Netting(K)}, {Lob(A)}, {Netting(K)}> => <{Lob(A)}>	0.096	0.7	1
14713	<{Lob(A)}, {Dropshot(K)}> => <{Drive(A)}>	0.137	0.37	1

Meanwhile, Table 7 shows the attack patterns formed by Anders Antonsen (A) and Kunlavut Vitidsarn (K) at the 2025 BWF Paris Championships, resulting in 14713 rules. The rule with the highest probability is the rule $\langle \{Lob(A)\}, \{Netting(K)\}, \{Lob(A)\}, \{Dropshot(K)\}, \{Netting(A)\}, \{Lob(K)\}, \{Lob(A)\}, \{Lob(K)\} \rangle \Rightarrow \langle \{Dropshot(A)\} \rangle$, which explains that the rule has a support of 0.055, confidence of 1, and lift ratio of 3.042. This means that the probability of Anders Antonsen performing a lob followed by a netting shot followed by Kunlavut Vitidsarn, a lob by Anders Antonsen, a dropshot by Kunlavut Vitidsarn, a netting shot by Anders Antonsen, a lob by Kunlavut Vitidsarn, a lob by Anders Antonsen, and a lob by Kunlavut Vitidsarn resulting in a dropshot by Anders Antonsen from all rallies during the match is 5.5%. Where the probability of Anders Antonsen performing a dropshot if the sequence of shots is a lob by Anders Antonsen, a net shot by Kunlavut Vitidsarn, a lob by Anders Antonsen, a dropshot by Kunlavut Vitidsarn, a net shot by Anders Antonsen, a lob by Kunlavut Vitidsarn, a lob by Anders Antonsen, and a lob by Kunlavut Vitidsarn has occurred is 1 with a rule strength of 3.42. These rules represent unique sequential patterns extracted from each match and are later mapped to their contextual occurrences for modelling.

Table 8. Class distribution based on all rules from SPADE

Class	Count
Drive (A)	986
Drive (K)	914
Dropshot (A)	2919
Dropshot (K)	9714
Lob (A)	11950
Lob (K)	11222
Netting (A)	17595
Netting (K)	4788
Smash (A)	4354
Smash (K)	6315

The rules obtained from the pattern mining process by SPADE are then rearranged in a concise table as shown in Table 9.

Table 9. Combined results of two SPADE matches

No	Rule	Se t	Rall y	Point	Suppor t	Confidenc e	Macth
1	<{Smash(A)}, {Netting(K)}, {Drive(A)}> => <{Drive(K)}>	2	1	Anders Antonse n	0.055	1	Toyota Thailand Open
2	<{Dropshot(K)}, {Netting(A)}, {Netting(K)}, {Lob(A)} > => <{Dropshot(K)}>	1	2	Anders Antonse n	0.151	0.786	Paris BWF Championshi p
3	<{Dropshot(K)}, {Netting(A)}, {Netting(K)}, {Lob(A)} > => <{Dropshot(K)}>	1	13	Kunlavut Vitidsarn	0.151	0.786	Paris BWF Championshi p
4	<{Smash(A)}, {Netting(K)}, {Drive(A)}> => <{Drive(K)}>	1	25	Kunlavut Vitidsarn	0.055	1	Toyota Thailand Open
5	<{Smash(A)}, {Netting(K)}, {Drive(A)}> => <{Drive(K)}>	1	58	Kunlavut Vitidsarn	0.055	1	Toyota Thailand Open
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
7056 2	<{Drive(A)}, {Netting(K)}, {Netting(A)}> => <{Netting(K)}>	2	72	Kunlavut Vitidsarn	0.685	0.357	Paris BWF Championshi p
7056 3	<{Smash(K)}, {Lob(A)}, {Lob(K)}, {Smash(A)}> => <{Netting(K)}>	2	24	Kunlavut Vitidsarn	0.685	0.71	Paris BWF Championshi p

7056 4	<{Smash(A)}, {Netting(K)}, {Lob(A)}, {Netting(K)}> => <{Lob(A)}>	2	56	Anders Antonsen	0.096	0.7	Paris BWF Championship
7056 5	<{Lob(A)}, {Dropshot(K)}> => <{Drive(A)}>	2	71	Kunlavit Vitidsarn	0.137	0.37	Paris BWF Championship

Table 9 presents the list of rules identified in each rally and match, allowing the results from multiple matches to be combined into a uniform format. The Rule column shows the sequential form of each rule, where the left-hand side represents the antecedent (the sequence of previous shots) and the right-hand side represents the consequent (the predicted next shot). The Set, Rally, and Point columns indicate the context in which the rule occurs, while the Match column specifies the tournament source (Toyota Thailand Open or Paris BWF Championship). Support denotes the frequency of rule occurrences, and Confidence indicates its reliability.

It should be emphasized that the increase in the number of rows (70,565) does not result from artificial duplication or synthetic upsampling. This increase arises from the contextual enumeration of rule occurrences combined with the sequential representation using a sliding window mechanism. Using the sliding window mechanism, a single rally can be transformed into multiple sequentially generated subsequences, where each subsequence constitutes a valid training instance with its own contextual attributes (set, rally, and point). Moreover, a sequential rule can appear in multiple rallies and different match situations, with each occurrence recorded separately for modeling purposes. Therefore, the growth in data size reflects the natural transformation of sequential match data into structured instances for modeling, rather than identical data replication.

3.4. Long Short-Term Memory (LSTM) Models Result

This study aims to develop a model to predict the next shot in a 10-class badminton game scheme through the integration of SPADE and LSTM results. During the modeling stage, the sequential patterns extracted using SPADE are not limited to pattern discovery, but are transformed into tabular features that represent the strength and context of the occurrence of rules that have been applied in the LSTM preprocessing and data splitting stages. To obtain the most suitable model configuration, hyperparameter tuning was performed using Bayesian Optimization, covering the dimension of the stroke embedding, the number of LSTM units, the size of the dense layer in the sequence and tabular branches, dropout, and the learning rate. The best hyperparameters obtained from the tuning process are presented in Table 8.

Table 10. Best tuned hyperparameters for the LSTM model

Parameters	Values
Stroke embedding size	64
LSTM units	128
Sequence dense units	96
Match embedding size	12
Tabular dense units	32
Dropout	0.1
Learning rate	0.002
Epoch	300
Batch size	256
Optimizer	Adam

Based on the hyperparameter tuning process that has been carried out, the best configuration was selected as the final model because it provided the highest validation performance and was stable during training.

Table 11. Best tuned hyperparameters for the LSTM model

Models	Accuracy	Macro-F1
LSTM-Only	0.52	0.54
SPADE-LSTM	0.88	0.88

The results in Table 11 show that the LSTM-Only model achieved an accuracy of 0.52 and a Macro-F1 score of 0.54, indicating that using raw sequential data alone is insufficient to capture

the complexity of stroke transition patterns. In contrast, the integrated SPADE–LSTM model achieved both an accuracy and a Macro-F1 score of 0.88, demonstrating a substantial performance improvement. This finding confirms that the inclusion of SPADE-based sequential rule features provides additional structural information that enhances the LSTM’s ability to learn tactical patterns. Therefore, the integration of SPADE and LSTM is empirically proven to be more effective than using LSTM without sequential pattern enrichment.

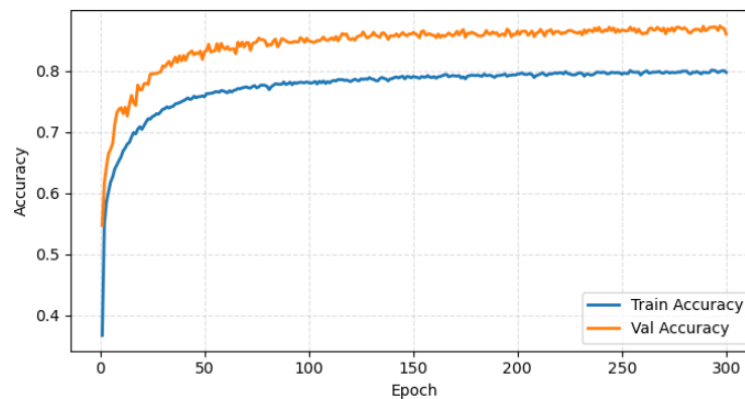


Figure 7. Train vs validation accuracy of the LSTM models

Figure 7 shows that the train vs validation accuracy curve indicates that the model learning process is stable and convergent. In the early epochs, the accuracy increases sharply, indicating that the model quickly captures the basic patterns of the data. After about the middle of the epoch, the increase in accuracy begins to level off and approaches a fixed value in the range of 0.88, so that training no longer provides significant improvement. The gap between the training and validation curves is relatively small, with the validation curve even slightly higher in some epochs. This pattern indicates that the model does not experience severe overfitting and has good generalization capabilities on the validation data. Thus, the model configuration used is sufficiently balanced between the ability to learn from the training data and performance stability on untrained data.

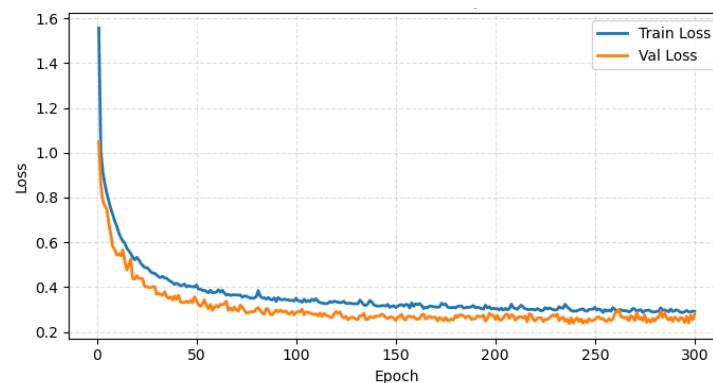


Figure 8. Train vs validation loss of the LSTM models

Meanwhile, Figure 8 shows the train vs validation loss curve, which indicates a sharp decline in loss during the early epochs, followed by a slight decrease until it reaches a stable condition. This pattern shows that the optimization process is running effectively, with the model quickly correcting prediction errors at the beginning of training, then gradually improving as the number of epochs increases. The validation loss value tends to be in line with the training loss and is even slightly lower in most epochs. This condition indicates that there are no strong signs of overfitting, because the validation loss does not increase when the training loss continues to decrease. Thus, the model has good generalization on the validation data, and the training has reached a convergence point when both curves begin to flatten.

After training using the LSTM model, the final evaluation was carried out using test data to predict the next punch based on existing sequence patterns using the best hyperparameter combination.

Table 12. Top accuracy prediction SPADE-LSTM

	Metric	Value
0	Top 1 accuracy	0.8868
1	Top 2 accuracy	0.9895
2	Top 3 accuracy	0.9994

The evaluation results show that the model achieved a Top 1 accuracy of 0.8868, which means that the model's main prediction matches the actual label in 88.68% of the test data. When the evaluation is expanded using a Top 2 accuracy of 0.9896, almost all correct labels are included in the two prediction candidates with the highest probability. Furthermore, a Top 3 accuracy of 0.9994 indicates that the actual labels are almost always among the top three prediction candidates, making the model effective for scenarios involving recommendations for multiple next-punch options. The evaluation results are presented in the form of a classification report containing precision, recall, F1-score, and support for each punch class, as shown in Table 10.

Table 13. Classification report of the LSTM models

Class	precision	recall	f1-score	support
Drive(A)	0.8974	1.0000	0.9459	35
Drive(K)	0.8684	1.0000	0.9296	33
Dropshot(A)	0.8623	0.9754	0.9154	122
Dropshot(K)	0.9341	0.9163	0.9251	263
Lob(A)	0.7491	0.9906	0.8531	428
Lob(K)	0.9853	0.5555	0.7103	362
Netting(A)	0.9980	0.8755	0.9328	1149
Netting(K)	0.7579	0.9980	0.8616	502
Smash(A)	0.9880	0.9347	0.9606	352
Smash(K)	0.8502	0.8529	0.8516	306
Accuracy			0.8868	3552
Macro Avg	0.8890	0.9099	0.8886	3552
Weighted Avg	0.9075	0.8868	0.8851	3552

The accuracy value obtained of 88,68% shows that the LSTM model is able to effectively learn the sequence of strokes and the context of the match (set, rally, and match information) to predict the next stroke. This performance can also be seen from the strong weighted average results, where the precision value is 0.9075, the recall is 0.8868, and the F1-score is 0.8851, indicating that the model's predictions tend to be stable and consistent. In terms of prediction accuracy, the high precision values in classes such as Netting(A) (0.9980) and Smash(A) (0.9880) indicate that when the model predicts these strokes, the results are almost always correct. Meanwhile, the high recall values in a number of classes such as Drive(A) (1.0000) and Drive(K) (1.0000) show the model's ability to capture most of the actual events in those classes. The dominant classes, especially Smash(A) and Drive(A) with support of 352 and 35, respectively, have very high F1-scores of 0.9606 and 0.9459, which reinforces that these shot patterns are relatively consistent and easily recognizable from the sequence context.

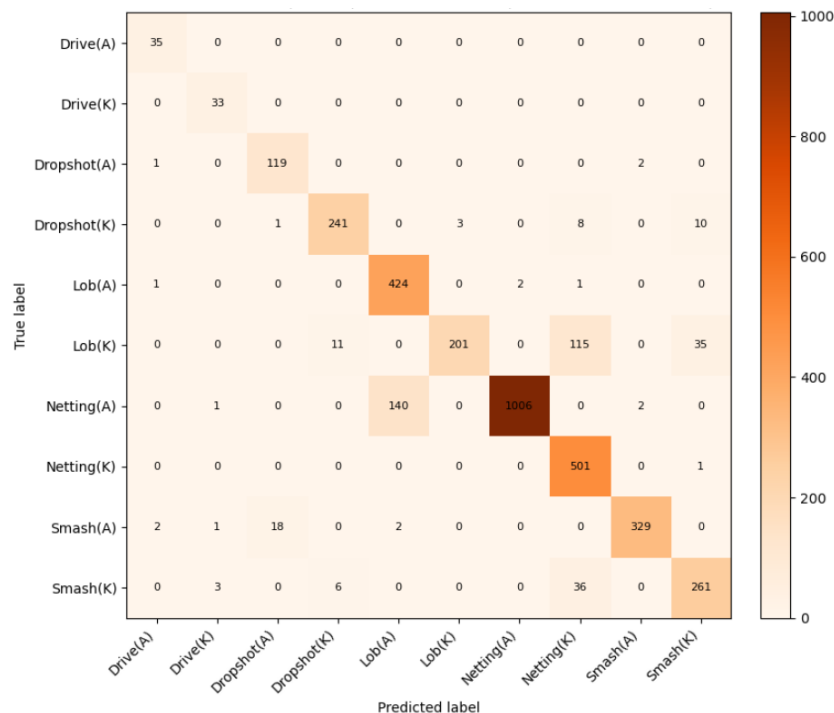


Figure 9. Confusion matrix of the LSTM models

The confusion matrix in the Figure 9 shows that most predictions are concentrated on the main diagonal, indicating that the LSTM model is able to distinguish punch classes well in a 10-class scheme. The most consistent performance is seen in the Netting(A) and Netting(K) classes, marked by very dominant correct predictions (e.g., Netting(A) = 1006 and Netting(K) = 501 on the diagonal). Other classes such as Lob(A), Smash(A), and Smash(K) also show high correct predictions on the diagonal (e.g., Lob(A) = 424, Smash(A) = 329, and Smash(K) = 261), while classes with small support such as Drive(A) and Drive(K) remain very well classified (diagonals 35 and 33) with almost no errors. However, errors are still evident in certain classes that are the main source of prediction errors. The biggest errors appear in the Lob(K) class, which is quite often predicted as Netting(K) (115 cases) and Smash(K) (35 cases), and a small portion as Dropshot(K) (11 cases). In addition, there is a prominent misclassification pattern in Netting(A), which is quite often mistaken for Lob(A) (140 cases). In other classes, the errors were relatively smaller, for example, Smash(A) was sometimes predicted as Dropshot(A) (18 cases) and Dropshot(K) was occasionally predicted as Smash(K) (10 cases) or Netting(K) (8 cases). This pattern indicates that under certain rally conditions, especially when shot transitions occur rapidly, some shot types have overlapping sequence contexts, making the boundaries between classes less distinct.

4. CONCLUSION

This study demonstrates that the integration of the SPADE algorithm and the LSTM model provides an effective framework for sequential pattern analysis and next-shot prediction in badminton. SPADE successfully extracts frequent and interpretable rally patterns, enabling structured identification of tactical tendencies. These rule-based sequential features enrich the representation of gameplay dynamics before predictive modeling.

The SPADE–LSTM model achieved strong predictive performance, with an accuracy of 0.8868 and consistent macro and weighted F1-scores, indicating balanced performance across the 10 stroke classes. The convergence of training and validation curves further confirms stable learning behavior without significant overfitting. These findings suggest that incorporating global sequential rules (support–confidence) into a deep learning framework substantially enhances the model’s ability to capture tactical stroke transitions.

Beyond predictive performance, the proposed framework contributes methodologically by bridging interpretable sequential pattern mining and non-linear sequence modeling. Practically, the

model offers potential support for coaches and performance analysts in identifying recurring rally structures and anticipating stroke tendencies. While the study is conducted on a controlled match setting, the framework itself is flexible and can be extended to broader datasets and diverse competitive contexts in future research.

REFERENCES

- [1] J. R. Sari and A. Ahdika, "Modeling the Badminton Stroke Pattern Through the Sequential Pattern Discovery Using Equivalent Classes (SPADE) Algorithm," *Mendel*, vol. 29, no. 1, pp. 37–44, Jun. 2023, doi: 10.13164/mendel.2023.1.037.
- [2] D. Cabello Manrique and J. J. González-Badillo, "Analysis of the characteristics of competitive badminton," *Br. J. Sports Med.*, vol. 37, no. 1, pp. 62–66, 2003, doi: 10.1136/bjsm.37.1.62.
- [3] E. Edmizal *et al.*, "Biomechanical Analysis of Smash Stroke in Badminton: A Comparative Study of Elite and Recreational Players: a systematic review," *Retos*, vol. 57, pp. 809–817, 2024, doi: 10.47197/retos.v57.107662.
- [4] G. Laffaye, M. Phomsoupha, and F. Dor, "Changes in the Game Characteristics of a Badminton Match : A Longitudinal Study through the Olympic Game Finals Analysis in Men ' s Singles," vol. 1992, no. January, pp. 584–590, 2015.
- [5] Y. Zhao, A. Zhu, S. Zhang, and Y. Zhang, "Comparison of the temporal and technical-tactical characteristics in badminton men ' s singles under different competition formats," no. September, 2025, doi: 10.3389/fpsyg.2025.1634776.
- [6] R. Green, A. T. West, and M. E. T. Willems, "Notational Analysis and Physiological and Metabolic Responses of Male Junior Badminton Match Play," 2023.
- [7] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and a I. Verkamo, "Fast discovery of association rules," 1996. [Online]. Available: <http://www.cs.helsinki.fi/hannu.toivonen/pubs/advances.pdf>
- [8] M. J. Zaki, "SPADE: An efficient algorithm for mining frequent sequences," *Mach. Learn.*, vol. 42, no. 1–2, pp. 31–60, 2001, doi: 10.1023/A:1007652502315.
- [9] D. A. Saputra, E. T. Tosida, S. I. Komputer, and U. Pakuan, "Penentuan Pola Sekuensial Data Transaksi Penjualan menggunakan Algoritma Sequential Pattern Discovery using Equivalent Classes (SPADE) (Diki Andika S, Eneng Tita T, Fajar Delli W)," vol. 16, no. 2, 2019, [Online]. Available: <https://journal.unpak.ac.id/index.php/komputasi> ■271
- [10] A. Noah, R. Intan, and A. N. Tjondrowiguno, "Menggunakan SPADE Algorithm Untuk Sistem Rekomendasi Film," *J. INFRA*, vol. 8, no. 1, 2020.
- [11] S. K. Fillah and D. Rahman Prehanto, "Sequential Pattern Mining Data Pemeriksaan Pasien Menggunakan Algoritma SPADE (Sequential Pattern Using Discovery Equivalent Classes) Studi Kasus BPI Albasrah Wajak," *Jeisbi*, vol. 03, no. 1, pp. 9–17, 2022.
- [12] W. W. T. Fok, L. C. W. Chan, and C. Chen, "Artificial intelligence for sport actions and performance analysis using Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM)," *ACM Int. Conf. Proceeding Ser.*, pp. 40–44, 2018, doi: 10.1145/3297097.3297115.
- [13] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Memory augmented deep generative models for forecasting the next shot location in tennis," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 9, pp. 1785–1797, 2020, doi: 10.1109/TKDE.2019.2911507.
- [14] Q. Zhang, X. Zhang, H. Hu, C. Li, Y. Lin, and R. Ma, "Sports match prediction model for training and exercise using attention-based LSTM network," *Digit. Commun. Networks*, vol. 8, no. 4, pp. 508–515, 2022, doi: 10.1016/j.dcan.2021.08.008.
- [15] L. Xiaoyan, R. C. Raga, and S. Xuemei, "GloVe-CNN-BiLSTM Model for Sentiment Analysis on Text Reviews," *J. Sensors*, vol. 2022, 2022, doi: 10.1155/2022/7212366.
- [16] W. Wang, T. Ik, N. Yang, and M. Chiao, "ShuttleSet : A Human-Annotated Stroke-Level Singles Dataset for Badminton Tactical Analysis," 2022, doi: 10.1145/3580305.3599906.
- [17] 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," 2021.

- [18] S. K. Dini, M. N. Aidi, and B. Sartono, "Sequential pattern mining of rawi hadis (Case study: Shahih hadis of Imam Bukhari from software Ensiklopedi Hadis Kitab 9 Imam)," *AIP Conf. Proc.*, vol. 1827, no. 2017, 2017, doi: 10.1063/1.4979446.
- [19] R. KRIBII and Y. FAKIR, "Mining Frequent Sequential Patterns," *J. BIG DATA Res.*, vol. 1, no. 2, p. 20, 2021, doi: <https://doi.org/10.14302/issn.2768-0207.jbr-21-3455>.
- [20] R. Bunker, K. Fujii, H. Hanada, and I. Takeuchi, "sport to identify important patterns of play : an application," pp. 1–19, 2021.
- [21] V. K. R. Chimmula and L. Zhang, "Time series forecasting of COVID-19 transmission in Canada using LSTM networks," *Chaos, Solitons and Fractals*, vol. 135, 2020, doi: 10.1016/j.chaos.2020.109864.
- [22] R. Wesley and R. Gunawan, "LITERATUR REVIEW : METODE DEEP LEARNING UNTUK ANALISIS TEKS," vol. 8, no. 5, pp. 11020–11023, 2024.
- [23] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Syst. Appl.*, vol. 185, no. July, p. 115524, 2021, doi: 10.1016/j.eswa.2021.115524.
- [24] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [25] S. Hochreiter and J. Schmidhuber, "Long short-term memory. Neural Com- putation," vol. 9(8), pp. 1735–1780, 1997.
- [26] R. C. Staudemeyer and C. W. Omlin, "Evaluating performance of long short-term memory recurrent neural networks on intrusion detection data," pp. 218–224, 2013.
- [27] T. T. H. Le, J. Kim, and H. Kim, "An Effective Intrusion Detection Classifier Using Long Short-Term Memory with Gradient Descent Optimization," *2017 Int. Conf. Platf. Technol. Serv. PlatCon 2017 - Proc.*, no. February, 2017, doi: 10.1109/PlatCon.2017.7883684.
- [28] K. N. Khikmah, "Kajian Kinerja Multiple Long Short-Term Memory Untuk Analisis Data Deret Waktu Takstasioner," Institut Pertanian Bogor, 2024.
- [29] B. Krawczyk, "Learning from imbalanced data : open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, 2016, doi: 10.1007/s13748-016-0094-0.
- [30] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks* Mateusz," pp. 249–259, 2018.
- [31] M. N. Razali, N. Arbaiy, and P. Lin, "Optimizing Multiclass Classification Using Convolutional Neural Networks with Class Weights and Early Stopping for Imbalanced Datasets," pp. 1–14, 2025.
- [32] J. Davis *et al.*, "Methodology and evaluation in sports analytics: challenges, approaches, and lessons learned," *Mach. Learn.*, vol. 113, pp. 6677–7010, 2024, doi: <https://doi.org/10.1007/s10994-024-06585-0>.
- [33] G. Cynthia and A. Qoiriah, "Klasifikasi Tingkat Kesiapan Kerja Mahasiswa Universitas Negeri Surabaya Menggunakan Algoritma C5.0," *J. Informatics Comput. Sci.*, vol. 6, no. 02, pp. 383–396, 2024, doi: 10.26740/jinacs.v6n02.p383-396.
- [34] M. Grandini, E. Bagli, and G. Visani, "METRICS FOR MULTI-CLASS CLASSIFICATION: AN OVERVIEW," *Statistics (Ber.)*, pp. 1–17, 2020, doi: 10.48550/arXiv.2008.05756.
- [35] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009, doi: <https://doi.org/10.1016/j.ipm.2009.03.00>.