Pervasive and Mobile Computing 83 (2022) 101608

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

DeCoach: Deep Learning-based Coaching for Badminton Player Assessment

Indrajeet Ghosh [a,*], Sreenivasan Ramasamy Ramamurthy [b], Avijoy Chakma [a], Nirmalya Roy [b]

[a] *Mobile Pervasive & Sensor Computing Lab and Department of Information Systems, University of Maryland Baltimore County, Baltimore, USA*

[b] *Mobile Pervasive & Sensor Computing Lab, Center for Real-time Distributed Sensing and Autonomy (CARDS) and Department of Information Systems, University of Maryland Baltimore County, Baltimore, USA*

A R T I C L E I N F O

Article history:
Received 29 October 2021
Received in revised form 2 May 2022
Accepted 5 May 2022
Available online 14 May 2022

Keywords:
Sports analytics
Coaching
Scoring
Error
Assessment
Badminton
Activity recognition
Deep learning
Racquet sports

A B S T R A C T

Wearable devices have gained immense popularity among various pervasive computing and Internet-of-Things (IoT) applications in the past decade. Sports analytics researchers recently focused on improving a player's performance to help devise a winning strategy based on the player's gameplay. Especially in a racquet-based badminton sport, it is assumed that handling the racquet during the gameplay is one of the primary reasons to influence the players' performance. On the contrary, we posit that the players' stance, body movements, and posture are equally significant in evaluating a player's performance during the game. A shot characterized by a recommended posture, stance, and body movements allows a player to play a stroke efficiently, thus aiding the player in guiding the shuttle to strategic spots and making it difficult for the opponent to return the shot and score a point. Relying on this hypothesis, we propose *DeCoach*, a data-driven framework that leverages the stance and posture of the players and ranks them based on their performances. In this effort, we first employ a deep learning-based algorithm to classify the strokes and stances of the players. Secondly, we propose a distance-based methodology to compare the obtained stance of a player with that of a professional player. Finally, we devise a deep learning-based regressor to predict the player's performance which commences with ranking based on their performance. We evaluate *DeCoach* using our in-house dataset, *Badminton Activity Recognition (BAR) Dataset* that is collected using inertial measurement unit (IMU) sensors by placing them on the upper and lower limbs of the players. The BAR dataset is collected from 11 players in the *controlled* and *uncontrolled environment settings* for 12 frequently played shots in the game. Empirical results indicate that *DeCoach* achieves *89.09%* accuracy for strokes detection and R^2 score of *88.84%* in estimating the players' performance.

1. Introduction

In recent times, human activity recognition with the power of wearable and IoT devices has paved paths for exciting research directions, especially in domains involving competitions. Common people can acquire and use wearable devices

* Corresponding author.
E-mail addresses: indrajeetghosh@umbc.edu (I. Ghosh), rsreeni1@umbc.edu (S. Ramasamy Ramamurthy), achakma1@umbc.edu (A. Chakma), nroy@umbc.edu (N. Roy).

https://doi.org/10.1016/j.pmcj.2022.101608
1574-1192/

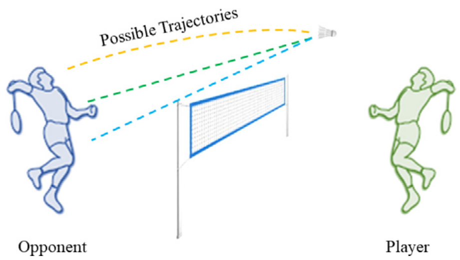


Fig. 1. Figure demonstrates the badminton gameplay strategy where the player is trying to outplay the opponent to achieve a score/point.

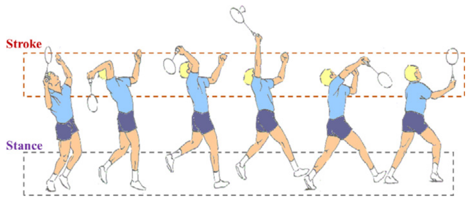


Fig. 2. Figure demonstrates the importance of coordinating the player's upper and lower limbs to play a successful shot. Dominant hand and legs equally contribute to an ideal body posture which is instrumental in executing a successful shot.

and activity trackers due to their convenience and affordability in the market. The popularity of such devices and the recent developments in the field of machine learning can facilitate the use of such devices in various fields such as ambulatory care, sports analytics, fitness tracking and training, workforce tracking and management, and many more [1]. *Wearable sports analytics* is a developing field that is gaining immense popularity globally. In major sports, professional and amateur players rely on the analytics team to hone their skills, assess the opponents, and develop winning strategies. Badminton is one such strategic and competitive sport that can help players significantly improve with the help of analytics.

Badminton game primarily relies on the strategical strokes, stamina, stances and positive mindedness of the player. A successful stroke is usually characterized by the racquet's power, position, and direction, which results in strategic placements of the shuttle based on the opponent's position and reflex recovery from the previous shot. For a player to win, the player must execute a successful shot and plan to recover from the effort of the previous shot and be ready for the opponent's subsequent attack. Therefore, it is of utmost importance to effectively execute their body posture and limb movements. Motivated by this, we believe sensing the movements of both upper and lower limbs can help understand the player's posture and thus improve the player's performance.

The most popular sensing modality for the application of sensing the movement of the limbs can be identified as the Inertial Measurement Unit (IMU) on a wearable device from the plethora of literature [2,3]. For instance, the accelerometer data acquired from Actigraph [4] has been used to detect fine-grained and nuanced dance activities [5]. Specifically, in the sports analytics domain, commercially available devices such as Zepp Sensor [6] are used in golf, soccer, etc., to track the player's performance. Particularly for the badminton game, Smart Xiaoyu [7] and Actofit [8] trackers are a few commercially available applications to estimate and analyze the player's performance based on 3D motion tracking. Moreover, these devices can only be positioned at the end of the racquet handle and only consider six different game strokes. We observed that most state-of-the-art industrial devices (particularly for racquet games) are placed at the end of the racquet handle. However, one of the significant drawbacks of the devices mentioned above is that they only consider the upper limb movements to classify and track the player's performance during the training sessions. Consequently, reducing the generalizability and scalability characteristics of these devices. On the contrary, we assume that the upper and lower limbs are equally crucial for playing a shot correctly and precisely. Figs. 1 and 2 depict the importance of upper and lower limbs and their coordination to maintain the ideal body posture to play a shot. This research gap is one of the reasons that motivated us to propose a novel algorithm that enables us to analyze the movements of the upper and lower limbs combinedly and measure the player's performance. Moreover, to support our proposition, in Fig. 3, we noticed that the novice player's limbs pose movements are inconsistent and have irregular execution patterns. We postulate that identifying and studying those erratic movements will help the players improve their game performance.

Several researchers focus on developing state-of-the-art sensing modules or machine learning (ML) algorithms in various sports analytics applications. The authors of [9] propose a multi-modal sensor sensing system that captures the smash shot and studies its correlation with the velocity of the shuttle. The velocity is measured using two-axis accelerometers (ADXL321), and the sounds of the shots are recorded using an acoustic sensor (BRT1615-06). Similarly, [10], the authors propose a multi-modal fusion of IMU and acoustic sensors to detect and record each badminton stroke considered in the study. They develop a voiceprint to detect the shuttlecock hitting the racket and data-driven classification algorithms to detect various strokes. Furthermore, data-driven ML algorithms such as Convolutional Neural Networks (CNNs) and

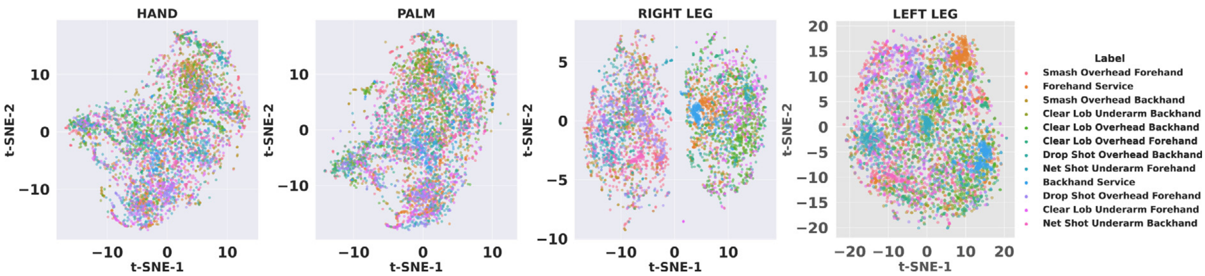


Fig. 3. The t-SNE plots correspond to each limb of the novice player in a 2D feature space and noticed that the novice player limbs' movements are mostly erratic and have indefinite patterns. We believe that each player's limb plays a vital role in studying and analyzing the style (fast/frequent, indefinite and inconsistency), gameplay planning, players' performance.

Deep Neural Networks (DNNs) are extensively used in many state-of-the-art algorithms to classify various badminton strokes [11]. The authors collected and experimented at various sampling frequencies (12.5 Hz, 25 Hz, 50 Hz, 100 Hz) using IMU (accelerometer and gyroscope) sensors in the Axivity AX6 6-Axis Logging device, placing three wearable sensors on the player's upper limb positions. Besides, they limited themselves by collecting only seven different badminton strokes from only two players. Contrastingly, we mount sensors on four different limb positions (wrist, palm, left leg and right leg) and consider an increased number of strokes compared to the above study. In [12], the authors develop a classifier model that can detect badminton strokes. The data is collected using a Magnetic Pickup Unit (MPU 6050 sensor), and an Arduino module is employed as a hub. Although the work concentrate on detecting the restricted number of strokes using the racquet's swing, the work does not consider the movements of all the limbs. Moreover, the dataset consider is small, which causes the concern of misinterpretation of the results. Furthermore in another study, [13], the authors leverage pre-trained models such as AlexNet [14], GoogleNet [15], VGG-16 and VGG-19 Net [16] to detect the action of badminton from the broadcast of the sport. The authors used a minimal dataset in the study, which caused the accuracy to vary drastically with a small change (1–2) in true positives. Furthermore, the study concentrates on a binary classification of whether the shuttle was hit or otherwise.

Scoring is defined as a machine-generated score that can help evaluate a player based on the skill compared to a professional or a rule book. In this study, we are looking at identifying the appropriate metrics to score a player. Considering the challenges of (1) leveraging a body sensor network that can capture the micro and complex sports activities (2) state-of-the-art industrial devices and research studies lacking in generalizability and scalability characteristics, which motivated us to propose a robust framework that addresses the above challenges. The contributions of our paper are summarized as follows:

- **Stroke Classification Module**: We collect *12* different types of badminton strokes and develop a *time-based decay learning rate scheduler convolutional neural networks (CNNs) classifier* that outperforms the state-of-the-art classification algorithms. Furthermore, we compare the stroke detection accuracy for each player by employing both shallow and deep learning algorithms and achieve a high score of *89.09% accuracy*.
- **Error Learning for Stance Retrieval Module**: Using the determined strokes of the professional player, we develop a *kNN-based error estimation* algorithm. By using the most appropriate stance from the professional player, we compute the error between the most appropriate stance and other players stance. Finally, we also hypothesize that the error learned can be used to determine the players' performance during the game.
- **Score Prediction Module**: We develop *deep regressor-based convolutional neural networks (CNNs)* to predict and analyze the players' performance in the game. We compare the scoring results of the players using both shallow regression and deep learning algorithms and achieve a high R^2 *score of 88.84%* in estimating the players' performance.
- **Evaluation**: We evaluate the proposed algorithms with our in-house *Badminton Activity Recognition (BAR)* dataset, comprising of **11** players (4 and 7 players collected in controlled & uncontrolled environment settings, respectively). We introduce a *cross-person validation approach* by introducing another professional player; to demonstrate the robustness of the *DeCoach* framework. We propose a *Success Rate* metric to estimate the players' performance during the training sessions for the qualitative analysis. Lastly, we demonstrate the generalizability and scalability characteristics of the *DeCoach* framework.

The paper is organized in the following. Section 2 discusses state-of-the-art research in the sports domain and applied methodologies related to the sport of badminton and enumerate predefined terms used in this study. The heuristic architecture and methodologies are explained in Section 3. In Section 4, we demonstrate the experiment performed in our proposed model. Section 5 illustrates the outcome and discussion of our proposed *DeCoach* model. We conclude in Section 6 and discuss the limitations and future research work in Section 7.

2. Related work

This section reviews the related literature in sports analytics, shallow and deep learning for sports activities and scoring activities in the sports domain. Recently researchers have fused various sports analytics applications with wearable-based human activity recognition. Lastly, we discuss the background and enumerate a few terms frequently used in this study.

- **Sports Analytics in AI Domain**: Recently, researchers have proposed novel approaches and techniques in sensor-based and camera-based applications in sports analytics covering a vast spectrum of topics ranging from activity classification [12], segmentation techniques [17,18], virtual reality console [19] and sports training decision support system [20]. Moreover, nowadays, researchers are investigating to understand the dynamics behind the game, which will help them postulate novel approaches to assist the players in improving their performances. One such work where the authors of [21] investigate the unprecedented surrounding contextual events which can influence the outcome of the evaluation model. To achieve the goal, the authors propose a novel multitasking algorithm that detects 5 different types of activities from 11 different sports and design a novel model for evaluating unseen sports activities. They also develop a two-stream deep learning multitasking model to collectively classify different activities and sports using images. We assume that data mining and ML techniques have paved new research directions to the community.
  ML techniques have proven to be the most helpful approach to understanding the game and player's performance and help the trainers and coaches keep track of the players' training sessions. Moreover, [22] design a novel model to capture and classify the badminton actions using statistical features to improve the training accuracy of the model. Statistical features might require low computational resources, but in contrast, deep learning features extractor gives a better feature representation of the data, illustrated in papers [3]. In [23] the authors introduce an image capture framework in which they capture the movements of the player by computing real-time focal length and camera parameters without any prior knowledge of the camera position and orientation in the field. They collect three different datasets of different video sessions and one publicly available soccer dataset. In the other study, the authors of [24] leverage visual analysis to detect the players, badminton court, classify strokes, and player's strategy using the hidden markov model (HMM). However, during experiment setup, video-based systems face a few challenges such as shadows, camera positioning (viewpoints), motion parallax, privacy-related concerns, fast, complex and micro activities of sports, etc. Such challenges might reduce the proposed systems' robustness, adaptability, and generalizability characteristics.
- **Sensor-based Micro & Complex Activity Recognition in Racquet Sports**: Sensor-based applications have proven to be one of the active fields, particularly in sports analytics. IMU Sensors has paved new research directions across the research community. One of the research directions is predicting and analyzing the activities performed by the participants across various sports. Sensors allow us to capture and analyze the significant change in IMU signal while performing the activities. Moreover, state-of-the-art novel techniques help us quantify the change in IMU signal while performing complex and simple activities. In [25] the authors develop a model to measure the acceleration of the upper and lower arm and the racket. It mentions that acceleration showcase a high correlation between ball velocity for national and international players. The racket swings are the composite of fast swings activities, and IMU sensor signals can help us to capture those swift swings. Similarly in [26] propose a model known as Tennis-Eye to calculate the speed of the tennis ball and prove the model is better than the state-of-the-art work. The authors confine themselves to using only the upper limb movements of the participants. However, we hypothesize that ideal body posture and proper coordination between the upper and lower limbs are instrumental in executing a successful shot, particularly for racquet games.
  We find interesting and novel approaches that help analyze and predict various micro and complex activities in racquet sports. One such literature [27], where the authors propose a novel approach that reduces the number of k-NN instances through the k-means clustering method by assigning labels to each cluster based on the maximum likelihood. Moreover, such a novel approach helps to reduce the computational time complexity of the proposed model compared to the brute force time complexity of the standard k-NN algorithm. Similarly, in [28] the authors designed an e-learning model for practicing the racket swings and classifying similar tennis strokes. They implement principal component analysis and K-means clustering to classify the swing data. In another work [29], the authors develop HMM to classify different types of badminton strokes and showcase that the model outperforms shallow machine learning algorithms: naive bayes, support vector machine, decision tree. Similarly, [30], the authors introduce a long-term recurrent region-guided convolutional network which used to track the actions and used SPPnet (a convolutional neural networks based architecture) to extract the features from the tracked portion. Furthermore, the authors use two standard datasets on their proposed model to check the generalizability of the proposed framework. They employ one publicly available UIUC2 sports dataset and another in-house video sports dataset for their study.
- **Applications for Scoring Activities in Sports Domain**: In the sports analytics domain, an automated scoring framework plays a vital role in analyzing and quantifying the players' activities during the training sessions or real-time gameplay. Therefore, we highlight and discuss available state-of-the-art algorithms and methodologies based solely on the sensor-based scoring pipeline in the sports analytics domain. For example, in work [31], the authors

propose a k-NN based stance retrieval approach in which they computed the k nearest data instances from the participants with respect to a professional player. Authors compute handcrafted error metrics based on the stroke and stance retrieval data matrix to model the error that occurred while performing the activities by the participants. In [32], the authors design a data-driven approach for the evaluation of performances of the soccer players by combining a player quality in a match. Their performance methodology is based on unsupervised role-based ranking as the role-based detector and formed k-means clustering for eight classes. In addition, the authors compute the normalized root mean square error and distribute it for the ranking. Finally, Pearson correlation coefficients are computed to determine the player's average performance. In contrast, *DeCoach framework* captures the body posture of the players (professional, novice and intermediate expertise skills levels) and learns the distinct variations and unique features. Furthermore, we use a template-based matching approach to compute the difference between players' postures.

In addition, to quantify the activities, an automated scoring framework can be extended and employed as the baseline for profiling the players based on the scoring framework. One such study [33], where the authors discuss an approach to devise attributes profiling techniques to improve the competence of the player and can also help them to formulate the gameplay plan. They designed Restricted Boltzmann Machine (RBM) to classify the shots. They also detect the change point detection in statistical analysis to develop attributes of profiling the players. However, the scoring specified in their study refers to the player's ability in the soccer game. Therefore, they constitute attributes of technical judgment, tactical awareness, and physical aspects to develop a personalized soccer profiling module. In contrast, in our study, the scoring module estimates the players' performance by capturing the posture of the players and quantifying those by employing metrics shown in paper [34].

2.1. Background

Badminton game is a popular, competitive and intense racket game that features multiple global tournaments, including the Olympics, World Championships, Commonwealth Games, and many more. We highlight and enumerate all the predefined terms frequently used in the badminton game and also in this study are as follows:

- **Strokes**: refers to swinging the racket to hit the shuttle directing to the opponent's side.
- **Stances**: refers to the leg movements associated with each stroke played shown in Fig. 1.
- **Error**: represent the difference of the variations and inconsistency between the professional's and other player's leg movements (stance) while playing a successful stroke.
- **Error Metrics**: quantifies the error obtained and moreover we have defined *13* different types of errors quantities: *MSE*, *RMSE*, *MAE*, etc.
- **Handcrafted Score**: is a score/point assigned to each player's actions based on their performance during the data collection sessions.
- **Anchor Player (AP)**: is the hypothetical player used for conducting the experiments w.r.t the professional player (PP), intermediate player (IP) and novice player (NP).

3. DeCoach architecture

This section enumerate the problem formulation and four major modules of the *DeCoach* framework: *stroke classification*, *error learning for stance retrieval*, *error metrics computing* and *score prediction* modules.

3.1. Problem formulation

We illustrate and highlight the problem formulation of the *DeCoach framework*. Assume there are two given datasets, D_p and $D_{\overline{p}}$, corresponding to the professional and participant players dataset, respectively. Moreover, the given dataset D_p and $D_{\overline{p}}$ consists of N number of samples, $(x_i^N, y_i^N, Y_i^N)_{i=1}^N$ where x_i, y_i and Y_i represent the data samples, corresponding activity & score labels, respectively. The datasets, D_p and $D_{\overline{p}}$, are comprised of groups of n sub-datasets $(D_1, \ldots, D_n)$, where the subscript denotes different sensor positions (hand, palm, right leg and left leg). Furthermore, we leverage the activity labels (y_i) of the participant player's upper limbs (hand and palm) for the stroke classification module. We aim to estimate the errors by employing lower limbs (right and left leg), leveraging the activity label information, and proposing a distance-based error learning paradigm. We calculated the distance between the data instances of a professional player and participant players. After that, we computed the various error metrics: EM $(E_1, \ldots, E_n)$, where n refers to the number of error metrics computed for this study. Lastly, we employed score label information (Y_i) and error metrics (EM) to predict the players' performance during the data collection sessions.

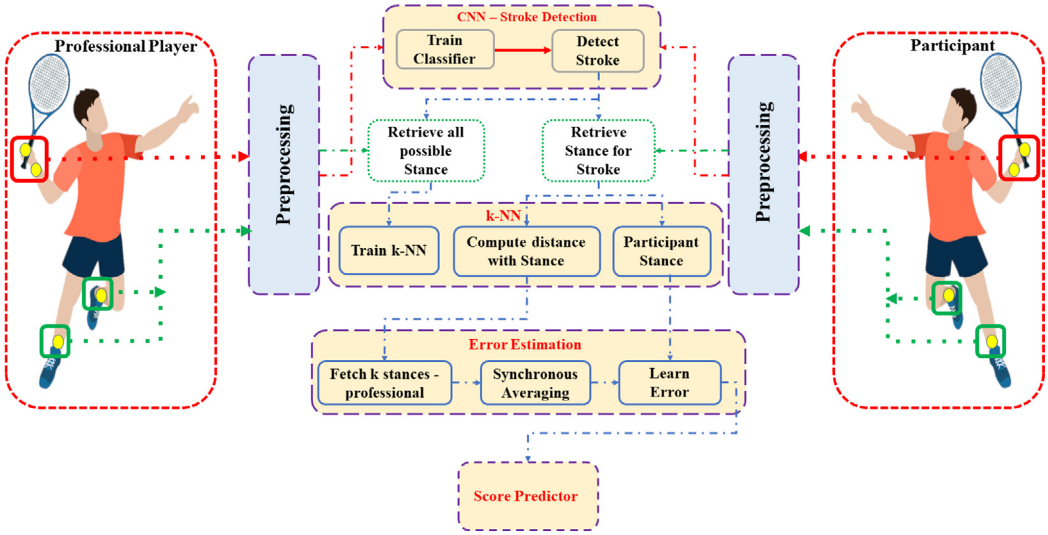


Fig. 4. *DeCoach Architecture*: The proposed framework consists of four major modules: *stroke classification*, *error learning for stance retrieval*, *error metrics computing* and *score prediction*.

3.2. Overview

We exhibit and discuss the overall architecture of the *DeCoach Framework*. In Fig. 4 depicts the overview of the proposed framework, and each module of the overall framework is enlightened in algorithms 1, 2 and 3. We categorize the proposed framework into four main modules: *stroke classification*, *error learning for stance retrieval*, *error metrics computing* and *score prediction*. Since badminton involves coordinated movements and both upper and lower limbs, it is essential to capture the movements of all the limbs. First, we collected the data from 4 and 7 participants in controlled and uncontrolled environment settings, respectively. The first component of the proposed work involves using the wearables in the upper limb to train a classifier where we aim to predict the stroke played by the player. Then, once the player's stroke is determined, we use the stroke information to determine the possible ways in which a player could use their lower limbs to execute the shot. We propose a *distance-based error learning for stance retrival* module to extract such possible stances to address this problem, where we employ instance-based template matching for ideal stance learning. Finally, we compare the ideal stances learned using the professional player's data with other participants. Furthermore, we believe that the error learned here between the professional player and other players can be used for scoring. We investigated various error metrics that can be useful to determine players' performance based on the professional player's performance and can be used as a good representation of the error made by the player. We employed feature selection on the various error metrics obtained to eliminate correlated error metrics. The new feature space of uncorrelated error metrics will refer to as an error matrix. Consequently, we propose a deep learning-based regressor convolution neural network (CNNs) to predict an overall score based on their performance. Besides, we also demonstrate the robustness of the proposed methodology by introducing another professional player's data.

Algorithms 1, 2 & 3 represent the pseudocodes of the *DeCoach* framework modules, where the P and Q refers to the professional and participant data, respectively, P_y and Q_y refers to the professional's and participant's activities labels for each limb data respectively, R refers to the handcrafted score associated with each stroke, *Score* refers to the predicted score obtained from the scoring module. The subscript *(UL & LL)* refers to upper and lower limbs, respectively.

3.2.1. Stroke classification module

We designed a *time-based decay scheduler learning rate convolution neural networks (CNNs)-based classifier* shown in Fig. 5, for recognizing **twelve** different types of badminton strokes discuss in Table 6. We implemented and compared several traditional machine learning algorithms such as random forest (RF), multilayer perceptron (MLP), decision tree (DT) and support vector machine (SVM). The proposed deep learning-based classifier comprises two components: the feature extraction component, followed by the classification component. The feature extraction component is responsible for hierarchically extracting high-level features from the raw data. We employed CNNs layers followed by max-pooling layers for the feature representation learning and followed by a softmax layer responsible for classifying the strokes. Each convolution layer 1D comprises the following operations in the sequence mentioned: convolution, rectilinear activation function, dropout and time-based decay learning rate schedule (where scaling factor = 1) shown in Eq. (2).

Algorithm 1 Pseudocode of Stroke Classification Module

INPUT: Acquire labeled data from P_{UL}, & Q_{UL} for classification module
OUTPUT: Retrieve all possible stances w.r.t the successful strokes
for label ∈ unique P_y and Q_y for upper limbs
  for epoch = 1 to e, total epochs **do**
    for each CNNs layers, learning rate = LR (equation (2))
    $\mathbf{C}_{UL}$ = training CNNs (P_{UL}, Q_{UL})
  # Forward Pass and Update weights
    end for
  end for
  $\mathbf{C}_{UL}$ ← **Retrieved all possible stance**
  return C_{UL}

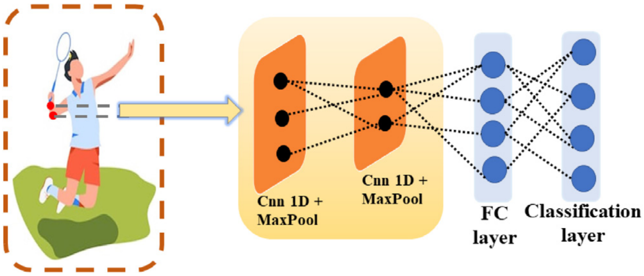


Fig. 5. *Stroke Classification Module:* comprises of CNNs layers to detect 12 different strokes using the hand and palm (upper limbs) dataset.

Mathematically, X_H represent the input matrix of participant hand data, Y represent the kernel size of the layer and then Eq. (1) represent the steps of the CNNs model operation. A Convolution layer is a product of one matrix of learnable parameters and another matrix called kernel producing an activation map, as shown in Eq. (1).

$$f[X, n] = [X_H * Y][X, n] = \sum_i \sum_j Y[i, j] X_H[X - i, n - j] \tag{1}$$

$$\text{LR} = \textit{Base Learning Rate} + \textit{scaling factor} * \textit{decay} * \textit{epoch} \tag{2}$$

3.2.2. Error learning for stance retrieval module

To compute the ideal stance that the participant players could play, we leverage the lower limb data of the professional. We compute the euclidean distance between each player and the professional player where X_p represent the professional player, and X_L represents participant player leg data. The motivation behind employing the euclidean distance metric as it shows better results for instance-based learning across a vast range of problems [35]. We extracted k closest data instances from the professional's data for each player data instance. Moreover, to extract k closest data instances, we employed the broadcasting function for arithmetic operations. The broadcasting function's advantage is handling different dataset size problems while performing arithmetic operations. We further performed synchronous averaging of the k retrieved data instances. The notion is that performing synchronous averaging causes a filtering effect on the retrieved k stances of the professional and removes the noise, leaving behind the ideal stance's core pattern. During the retrieval of k instances, the labels of the professional's and the intermediate/novice player's data were matched. By employing distance-based approach, we obtained the newly predicted values where X_k represent the synchronous averaging of the participant player data. Finally, we hypothesize that the obtained pattern is the ideal stance for playing any stroke based on the stance is retrieved from the professional's data.

Mathematically, *error learning module* help us to learn the distinct feature mapping functions of the anchor player *(P)* and corresponding player's *(Q)* data.

$$\textit{Euclidean Distance}\,(P, Q) = ||P - Q||_0 = \sqrt{\sum_{i=1}^{n} (P_i - Q_i)^2} \tag{3}$$

$$\begin{aligned} X_D &= \textit{Euclidean Distance}(P, Q): \quad \text{computing euclidean distance} \\ X_K &= \textit{Synchronous Average}(X_D): \quad \text{averaging } K \text{ values} \end{aligned} \tag{4}$$

Table 1
Error metrics.

Error metrics	Advantages of the metrics	Equation
Root Mean Square Error [RMSE]	Penalizing extreme errors	$\sqrt{mean(e_t^2)}$
Mean Square Error [MSE]	Provides linear-quadratic loss function	$mean(e_t^2)$
Mean Absolute Error [MAE]	Avoids mutual cancellations of the positive and negative errors	$mean(\|e_t\|)$
Normalized Mean Square Error [NMSE]	Normalizes the outlier values and reduces the variance in the resultant data	$\frac{MSE}{variance}$
Median Absolute Error [MdAE]	Preserves the same unit of measurement and median is more resistant to outliers	$median(\|e_t\|)$
Relative Median Square Percentage Error [RMdSPE]	Normalized by the actual data and are dimensionless	$\sqrt{100 * median * \frac{\|e_t\|}{X_{LL}}}$
Normalized (by SD) RMSE [NRMSE_sd]	Unbiased estimator that incorporates model bias and its variance	$\frac{RMSE}{SD}$
Sum of Squared Error [SSE]	Removes negative notations	$\sum e_t^2$
Euclidean Distance (L2-norm) [ED]	Removes negative notations and normalizes the distance	$\sqrt{SSE}$
Normalized (by mean) RMSE [NRMSE_m]	Unbiased estimator that incorporates prediction model bias and its means	$\frac{rmse}{e_t}$
Mean Percentage Error [MPE]	Measures the bias	$100 * mean(\frac{e_t}{X_{LL}})$
Mean Normalized Bias [MNB]	Measures the relative average forecast error	$mean(\frac{e_t}{X_{LL}})$
Median Square Percentage Error [MdSPE]	Removes the negative or positive outliers notation	$\frac{rmse}{e_t}$

Algorithm 2 Pseudocode of Error Learning for Stance Retrieval Module

INPUT: Acquire data from multiple limb positions from P_{LL} and Q_{LL}

OUTPUT: Error metrics matrix

1: **for** label $\in$ unique P_y for lower limbs

2: Extract data: P_{LL} *and* Q_{LL}

3: $X_D \leftarrow (K$ = 5 - 100) from (P_{LL} and Q_{LL} [label = (P_y)])

4: $X_K \leftarrow$ Average of k closest sample instances of X_D

5: $Error_{LL} \leftarrow$ *append* [*Compute Error*(Q_{LL}, X_K)]

6: end for

7: **return** $Error_{LL}$

Algorithm 3 Pseudocode of Score Prediction Module

INPUT: Acquire data from distance-based error learning module and handcrafted score

OUTPUT: Predicted automated score

1: E = Extract Uncorrelated Error Metrics ($Error_{LL}$) > **0.90**

2: **for** epoch = 1 to e, total epochs **do**

3: **for** each CNN regressor layers

4: Score = training Reg CNN (E, R)

\# Forward Pass and Update weights

5: **end for**

5: **end for**

8: **return** Score

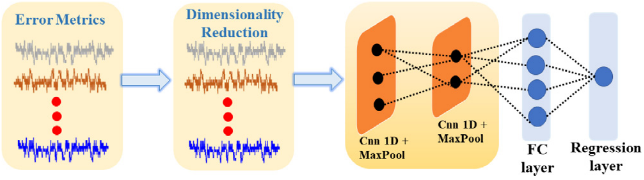


Fig. 6. *Score prediction module:* adopted a CNNs-based regressor for predicting and estimating the score with respect to the handcrafted score.

Table 2
Mathematical notation.

Symbol	Denotation
$\sqrt{x}$	Square Root of x variable
$\lvert x \rvert$	Modulus (MOD) of x variable
$\sum x$	Sum of x variable
x^2	Square of x variable

3.2.3. Error metrics

Following the computation of the ideal stance for each shot played by the participant players, we compute the error metrics [34,36]. The detailed explanation of each error metric is listed in Table 1 along with the mathematical equations. To obtain the error metrics for each activity, we preserve the labels and compute the error respective to the activities by implementing standard matrix operations on the resulting data from the distance-based model described in Section 3.2.2. Our experiment extracted the closest k instances from professional players to other participants discussed in the above section. By performing distance-based, we obtained the newly predicted values. For learning the error between the newly predicted values and actual data, we calculated 13 error metrics listed in Table 1 and mathematical notations listed in Table 2. Lastly, we plotted the probability density function plots for each stroke considered in the study to measure the error during the data sessions.

Let E_t represent the error between the actual participant data X_L and newly predicted values X_k. Error Metrics (EM) represents the handcrafted error metrics matrix shown in Table 1, where n represent number of data instances.

$$
\begin{aligned}
&e_{MSE} = e(1)_{MSE}, e(2)_{MSE}, e(3)_{MSE}, ., ., e(n)_{MSE} \\
&\vdots \\
&e_{MdSPE} = e(1)_{MdSPE}, e(3)_{MdSPE}., ., e(n)_{MdSPE} \\
&EM = \sum_{i=1,j=1}^{n} [e_{MSE} \cdots e_{RMSE} \cdots e_{MdSPE}]
\end{aligned}
\tag{5}
$$

3.2.4. Score prediction module

We design a convolution neural network (CNNs)-based regressor shown in Fig. 6 to predict each participant's score for the handcrafted error vector features. We assign handcrafted scores to each stroke played by the participants' which is shown in Table 5. We define the handcrafted score as referring to the ideal stance, which is discussed in Table 6. The motivation behind developing a deep regressor convolution neural network (CNNs) is that the ability to extract high-level features from input feature space hierarchically gives a cutting-edge advantage over the traditional regression algorithms. This motivated us to design an automated deep score predictor pipeline. Our deep regressor model comprises three main components. The first component is the handcrafted computed error metrics used as the input feature space, followed by the dimensionality reduction block, which helped us to extract the most uncorrelated error features with a correlation (greater than 0.90) to obtain better feature representation from the input data. The third component is the deep feature extraction layers, comprising two 1D convolution layers (responsible for hierarchically extracting high-level features from the input space), followed by a flattened layer to reduce the features' dimensionality. Furthermore, the regression component consists of one fully connected layer, followed by the regressor layer consisting of 1 neuron as the output layer.

Let f_i, y, E_t, $f_i(E_t)$ and Y represents feature selection correlation score, activities, error, score associated with each error metric and kernel size explained in Eq. (6), respectively.

$$
\begin{aligned}
&I = (f_i(E_t) > 0.90) - \text{ represent the feature selection procedure} \\
&f[X, n] = [I * Y][X, n] = \sum_i \sum_j Y[i, j] I[X - i, n - j]
\end{aligned}
\tag{6}
$$

Table 3
Players detail collected in a controlled environment setting (where RH and LH refers to right-hand and left-hand dominant player).

Player	Expertise	Dominant Limb
PP-1 - Male	Professional Player	LH
PP-2 - Male	Professional Player	RH
IP - Male	Intermediate Player	RH
NP - Male	Novice Player	RH

4. Experimental result & analysis

This section comprises of Sections 4.1, 4.2, 4.3 and 4.4. We discuss in-detail data collection setup, data preprocessing techniques, overall experimental setup including computing resource details and evaluation metrics employed for each module.

4.1. Dataset collection

We conducted our in-house data collection setup by employing four Shimmer3 IMU Unit [37] wearable devices which were placed on the player's dominant wrist, dominant palm, left and right leg positions. The shimmer devices were equipped with 3 axis low noise accelerometer, 3 axis high noise accelerometer, 3 axis gyroscope and 3 axis magnetometer. The racket's and sensor weights varied from approximately 75 grams and 23–30 grams, respectively. We ensured that the players were comfortable playing the shots while sensors were attached to different body positions. Moreover, each device orientation was maintained consistently for each player and collected 30 iterations of each stroke by the players throughout the data collection sessions. Although the explanation of the stances in Table 6 may look vague, all the players played the strokes and stances as a player would typically play. As playing becomes much easier compared to explaining the stances with words, we did our best to explain the stances in Table 6, along with associated scores with each stroke and stance performed by the player described in Table 5. The data distribution of each class of the BAR dataset is shown in Fig. 7. The activity labels for the dataset were assigned by an observer using the recorded video camera [38]. We obtained the best angle view from where all the sessions can be captured effortlessly so that the number of cameras required is low. Our goal was to maintain a simple and convenient data collection setup. We appointed two annotators to assign ground-truth labels to the collected dataset (both have a background in ML and badminton), where the first annotator assigned the activities labels, and the second annotator cross-validated the assigned labels. Each activity's start and end times were noted during the data collection sessions, and the labels were assigned based on that. In this way of assigning the labels, we felt that there might be an error at the start and end of each activity; however, since the sampling frequency was 512 Hz, the error may be very minute and does not affect the classification and scoring tasks. Furthermore, the high-speed (micro-seconds) activities level and most strokes are composite of jerk and flick actions. Such experience encouraged us to collect 30 iterations of each stroke at a high sampling frequency. For our in-house dataset, we considered two different environment settings:

- **Controlled and Uncontrolled Environment Settings**: In the controlled setting, we collected the datasets from 4 players (4 males; age: 26–30 years) in an enclosed sports premises, detail shown in Table 3. We selected the players based on their experience and proficiency in the game for the controlled environment setting. Moreover, we have categorized the four players into three levels of expertise: *professional*, *intermediate*, and *novice*. We examined that the gameplay of the volunteers varies based on their experience of the badminton game. As a result, we discovered that both professional players play the game very frequently and demonstrate high game skills compared to other players. We validated our assumption based on the recorded videos analysis of the data collection sessions. We also collected the dataset from more diverse players where we collected data in an uncontrolled environment setting, from 7 players (3 females and 4 males; age: 24–28 years) in an open playground, detail shown in Table 4. In the uncontrolled setting, we randomly selected the players without prior knowledge of the players' proficiency.

4.2. Data preprocessing

We highlighted and discussed the data processing techniques employed for our experiment, and we first dropped the missing values from the raw data. Further, we normalized the 48 raw features and employed a sliding windows technique with 50% overlap on the labeled data. The reason behind employing the sliding overlapping windows technique is that the body-worn IMU sensors provide continuous time-series data, and a segment of the time-series data represents an activity performed. Furthermore, we considered the 3 axis data of wide-range accelerometer, low-noise accelerometer, gyroscope and magnetometer sensors from four body sensors positions as raw features. In Fig. 7 shows the number of data instances for each activity in the *BAR* dataset. We examined various window sizes, and we achieved the best results of window size of **64**. We employed the majority voting for data labeling for each window segment within each window. Further, we

Table 4
Players detail collected in an uncontrolled environment setting (where RH and LH refers to right-hand and left-hand dominant player).

Player	Dominant Limb
P-1 - Male	RH
P-2 - Female	RH
P-3 - Male	RH
P-4 - Female	RH
P-5 - Female	RH
P-6 - Male	RH
P-7 - Male	RH

Table 5
Score table for the players.

Point Score	Stance and Footwork
0	Stance and Stroke performed incorrectly
1	Played the stroke correctly but not correct stance performed
2	Performing the ideal stance while playing the stroke
3	Follow-up after the stroke played
4	Stance and Stroke performed correctly

Table 6
Detailed description of the strokes and stances considered for this study.

Label	Racket Position	Stroke	Stance
1	Forehand	Service	Subtle leg movement
2	Backhand		
3	Forehand	Clear Lob Overhead	Step back with a slight jump
4	Backhand		
5	Forehand	Clear Lob Underarm	Step sideways
6	Backhand		
7	Forehand	Net Shot Underarm	Lunge like forward steps
8	Backhand		
9	Forehand	Drop Shot Overhead	Stand and deliver or slight jump or slight lunge
10	Backhand		
11	Forehand	Smash Overhead	Heavy movements and jumps
12	Backhand		

Table 7
Hyper-parameters of Classification and Scoring modules.

Hyper-parameters	Values
No. of maximum convolution layers	1,2,3
No. of filters in convolution layers	256, 200, 200
Convolution filter dimension	35×1, 25×1, 20×1
No. of maximum fully connected layers	1
No. of neurons in fully connected layers	512, 1
Batch size	128
Dropout rate	0.55
Learning rate	0.0003
Max number of epochs	300, 150

applied a median filter for the deep learning model and shallow learning techniques instead of the kalman filter because of comparatively higher computational complexity. We noticed a comparable increase of 2.3% to 3% for shallow learning algorithms after using a median filter. The reason behind employing filtering techniques is to eliminate the artifacts and errors such as motion artifacts, synchronous errors, sensor malfunctions, etc. from the raw sensory dataset to obtain a better feature representation of the performed activities.

4.3. Experimentation setup

The experiment is conducted on a Linux server integrated with Intel i7-6850K CPU, 4x NVIDIA GeForce GTX 1080Ti GPUs and 64 GB RAM. All the codes of data preprocessing, traditional machine learning and deep learning algorithms are implemented with Python. Especially for deep learning, Keras libraries are used. We used a training and testing split of 80%–20% and a training, testing, and validation split of 60–20%–20% for the traditional machine learning and CNNs-based

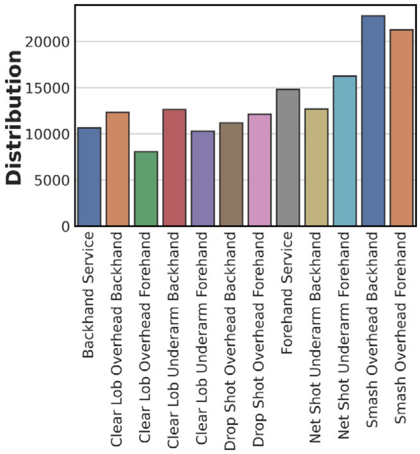


Fig. 7. No. of data instances in BAR dataset.

Table 8
F1 Score for strokes classification module.

Combinations	Train Dataset	Validation Dataset	Testing Dataset
All players	92.50%	86.20%	86.27%
PP and IP	92.61%	83.98%	83.97%
PP and NP	93.75%	87.71%	88.02%
IP and NP	97.41%	93.16%	93.66%

Table 9
F1 Score of optimized strokes classification module for all players.

CNN Layers	Train Dataset	Validation Dataset	Testing Dataset
1-Layer, 1-Fully Connected	86.55%	84.41%	84.72%
2-Layers, 1-Fully Connected	90.43%	88.02%	88.49%
3-Layer, 1-Fully Connected	95.71%	89.36%	89.09%

classifier algorithms, respectively. Furthermore, the validation dataset is use to fine-tune the hyperparameters of the *DeCoach* modules shown in Table 7. During the data collection sessions, the volunteers experimented by following rules and regulations and maintaining the standard badminton court setup. The data collection sessions are recorded through an action camera [38].

4.4. Evaluation metrics

For evaluation for the *DeCoach framework*, we computed six evaluation metrics; *accuracy*, *precision*, *recall*, *F1 Score*, *informedness* and *markedness* for the stroke classification module. We computed two regression metrics for the scoring prediction module:- *mean squared error (MSE)* and R^2-*Score*, where N is the number of data instances. We proposed a *success rate* metric shown in Eq. (9) for the players' performance during the data collection sessions.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(X_{actual\ value} - X_{predicted\ value})^2 \tag{7}$$

$$R - Squared\ Score = 1 - \sum_{i=1}^{N}\frac{(X_{actual\ value} - X_{predicted\ value})^2}{(X_{actual\ value} - mean(X_{actual\ value}))^2} \tag{8}$$

$$Success\ Rate = \sum_{i=1}^{N}\frac{Total\ of\ number\ of\ strokes\ obtained\ a\ perfect\ score}{Total\ of\ number\ of\ strokes} * 100 \tag{9}$$

5. Result & discussion

This section discusses and highlights the results and insights accomplished from the proposed *DeCoach* framework. We summarize the results and discussion into three subcategories: *classification module*, *error learning for stance retrieval module* and *score prediction module*.

Table 10
Comparison to the baseline classification state-of-the-art algorithms.

Models	Accuracy	Precision	Recall	F1 Score	Informedness	Markedness
Anik et al. [12]	58.78%	58%	58%	58%	57.4%	57.47%
Steels et al. [11]	87%	86.5%	86.5%	86.5%	86.58%	86.58%
Wang et al. [29]	75.89%	75.11%	75.10%	75.11%	75.59%	75.23%
DeCoach	**95.21%**	**95%**	**95%**	**95%**	**95.21**%	**95.21**

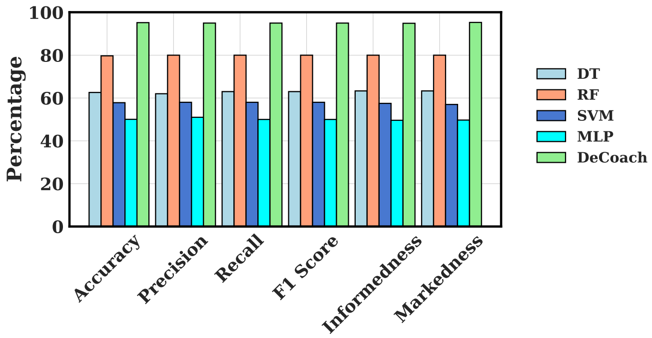


Fig. 8. Comparison of traditional machine learning and *DeCoach classification module*.

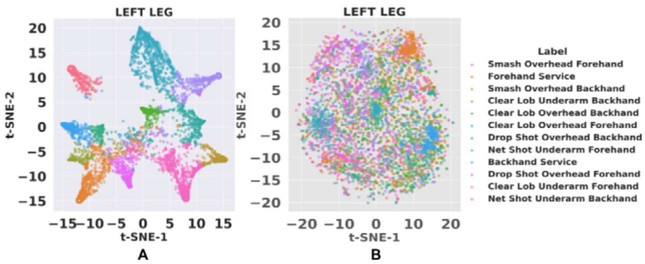


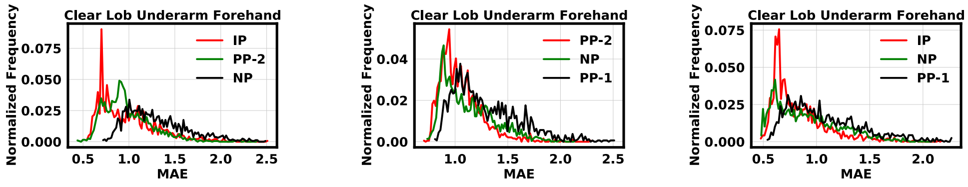
Fig. 9. ***(A)*** and ***(B)***: t-SNE plots represents the learned feature representation from the classification module and raw features representation of the NP in 2D space, respectively.

5.1. Stroke classification module analysis

This subsection highlights and discusses the results obtained from the stroke classification module. We designed a deep learning-based classifier to detect 12 different shots for the classification module when all the players' data are considered. We implemented four traditional machine learning algorithms; decision tree (DT), random forest (RF), support vector machines (SVM) and multilayer perceptrons (MLP); the results are shown in Fig. 8. Furthermore, we cross-validate the stroke classification module by employing six different classification evaluation metrics: *accuracy, F1 score, precision, recall, informedness and markedness*. We noticed that overall, deep learning has an improvement of approximately $\approx$ **15%** when all the player's data is compared to the traditional machine learning algorithms listed above. We believe that the reason behind such a phenomenon is due to the high frequency and complex micro activities available in the *BAR* dataset. Due to the shallow feature extractor architectures of the DT, RF, SVM and MLP algorithms, it cannot learn the complex feature representations from the raw data. In contrast, the convolution neural networks (CNNs) algorithm successfully learns, differentiates, and captures those erratic movements due to their ability to extract low and higher-level features and high convergence rates compared to the traditional algorithms.

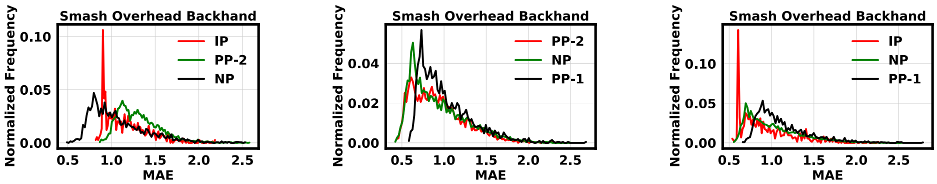
We investigated the hyperparameter tuning technique to reduce the number of parameters employed for the classification module. The classification results for the optimized proposed classification module are reported in Table 9. By employing the following hyper-parameters listed in Table 7, we achieved the best results for our classification module. Moreover, to obtain better results without compromising the model accuracy, we tried varying the number of convolution layers between one to three layers to maintain better accuracy. Such an experiment aims to optimize the classification pipeline, which reduces the training time and computational resource requirements. Fig. 9 represents the distinct features learned in a 2D space from the classification module. The learned features are separable and differential for each activity compared to the simple raw feature representation shown in Fig. 8 (B). It also strengthens the effectiveness of the classification module by learning the fast, erratic and indefinite patterns of each limb.

We performed *cross-person validation* approach by employing PP-2 and IP as the anchor player (AP). We highlighted those results in Table 8. We noticed the highest accuracy with IP and NP in combination. Such a phenomenon is that



(a) (b) (c)

Fig. 10. The subfigures (a, b, c) exhibit the error metrics in *pdfs* plots for **Clear Lob Underarm Forehand Stroke**. Subfigure (a) highlights the error of the players w.r.t the PP-1 (AP), (b) highlights the error of the players w.r.t the IP (AP) and (c) highlights the error of the players w.r.t the PP-2 (AP).



(a) (b) (c)

Fig. 11. The subfigures (a, b, c) exhibit the error metrics pdfs plots for **Smash Overhead Backhand Stroke**, where (a) highlights the errors w.r.t the PP-1 (AP), (b) highlights the errors w.r.t the IP (AP) and (c) highlights the errors w.r.t the PP-2 (AP).

we noticed that both NP and IP have similarly played the shots during the data collections. Their styles were also very symmetrical to each other. Finally, we performed an analysis based on comparing our proposed classification pipeline with different available state-of-the-art works; the results are shown in Table 10. Our three layers of convolution neural network followed by a max-pooling layer of classification pipeline outperform all the available state-of-the-art algorithms by $\approx$ **9%**. We found that our proposed classification module outperform [11], the proposed classification algorithm. The proposed algorithm comprises two layers of 2D-CNNs, fully connected and one batch normalization layer. We obtained **F1 score** of 86.5% by employing the same hyper-parameters employed in their study. The reason behind decrements in performance is the following reasons: (i) proposing a time-based decay learning rate scheduler where the learning rate converges faster and stabilizes the overall model performance (ii) employing max-pooling layers help us to capture sharp and low-level features from the data.

5.2. Error learning for stance retrieval module analysis

Furthermore, for the error learning for stance retrieval module, we employed a distance-based K-NN algorithm examining K values ranging from 5 to 100. We observed that (K = 25) obtained the best representation of instance-based pairwise results for our experiments. Our motivation for employing a distance-based error metrics algorithm is to obtain the best feature representation of players' limbs with respect to the PP. Furthermore, to quantify and estimate the error between the PP and the players, we calculated thirteen different errors shown in Table 1. Finally, we plotted probability density function (*pdf*) plots for each error metric to exhibit the results and insights. Furthermore, to evaluate the proposed scoring pipeline, we execute and assume three different case scenarios where: ***(a) PP-1***, ***(b) IP*** and ***(c) PP-2*** are referred as the anchor player (AP). The motivation behind such case studies is to demonstrate the *generalizability, scalability and robustness* characteristics of the error learning module.

Fig. 10, *clear lob underarm forehand stroke* is examined; in Figs. 10 (a, b, c), we observe notable trends, one of which is that the IP has a high likelihood of getting a high error, particularly in the case (a) compared to the PP-2 and NP. The phenomena behind such occurrence are because the IP played this particular stroke with high precision and accuracy compared to the other players. We also have recorded ground-truth videos of the data collection sessions to bolster our hypothesis. Furthermore, another notable trend we observe is that the PPs in both cases (a) and (c) tend to get a low error of high frequency which means that both the PPs have a high likelihood of getting low errors. Moreover, in case (b), we observed that the error *pdf* plots for all the players have a high percentage of overlapping errors. Such phenomena are because the players have almost the same intensity and style of playing this particular stroke. Lastly, we observed that the IP is highly likely to get a high score for the clear lob stroke.

Fig. 11, the stroke considered is the *smash overhead backhand stroke*, and is supposed to be one of the toughest and most challenging strokes to be played in a game. In Figs. 11 (a and c), the NP for both cases has the same tendency of obtaining low error with high likelihood, which means the NP will obtain a low score for this particular stroke. We can see that the NP was outmaneuvered while playing this particular stroke to support our hypothesis. Another trend we can

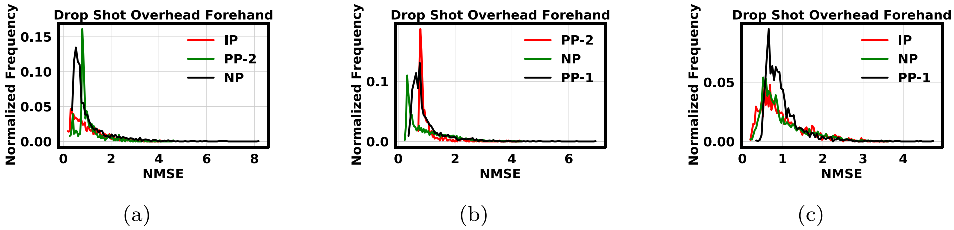


Fig. 12. The subfigures (a, b, c) exhibit the error metrics pdfs plots for **Drop Shot Overhead Forehand Stroke**, where (a) highlights the errors w.r.t the PP-1 (AP), (b) highlights the errors w.r.t the IP (AP) and (c) highlights the errors w.r.t the PP-2 (AP).

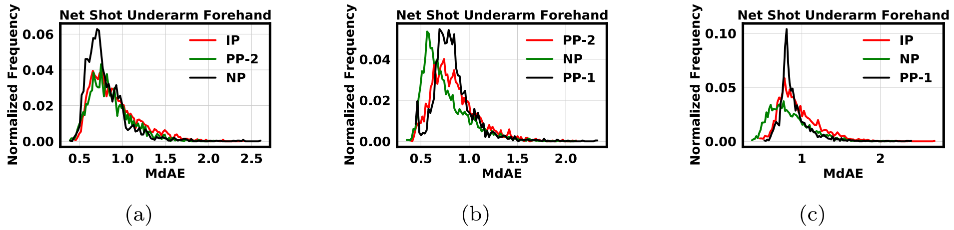


Fig. 13. The subfigures (a, b, c) exhibit the error metrics pdfs plots for **Net Underarm Forehand Stroke**, where (a) highlights the errors w.r.t the PP-1 (AP), (b) highlights the errors w.r.t the IP (AP) and (c) highlights the errors w.r.t the PP-2 (AP).

observe is that both the PPs have slightly high errors in both the cases (a) and (c) compared to PP-1 and PP-2 players, respectively. The reason behind such occurrence is that the PP-1 and PP-2 players are left-handed and right-handed, respectively. Another trend we noticed in case (b) is that the PP-1 player has a high likelihood of error; the reason behind such phenomenon is that the PP-1 player and the IP is left-handed and right-handed dominant, respectively. Moreover, we noticed that during the data collection sessions, the IP and PP-2 player had played the stroke with high intensity and precision compared to the PP-2 player and the NP. The motivation behind the case studies is to highlight and showcase the adaptability and robustness of the distance-based error metrics framework. Further, we considered another stroke known as *net underarm forehand stroke* in Figs. 13 (a, b, c), which is slightly difficult to play as compared to the other strokes. Furthermore, we observed that during the data collection sessions, we noticed that the IP played this particular stroke with low precision and accuracy compared to the other players, and coincidentally the NP played the stroke accurately, and we can also notice that both the PPs have a low error compared to the IP, as the player played the stroke exceptionally well which can be seen in cases (a) and (c). Furthermore, to validate our hypothesis, we can see that case study (c) shows that both players have high errors with respect to the IP. Another trend we noticed is that in case (b) the PP-1 player has a high occurrence of errors; the reason behind such phenomena is that the PP-1 player is a left-dominant player, and the framework is unable to learn the error accurately in this case.

Moving forward, in Figs. 12 (a, b, c), we consider another stroke, *drop overhead forehand stoke*. By looking at case (c), we noticed a common trend in which the IP is prone to high error with a high likelihood compared to other players. The reason behind such an occurrence is that the IP outplayed the PP-1 for this particular stroke during our data collection video analysis. Another notable trend we can observe is that in case (a) for the IP and case (b) for the PP-1, the lower error is higher for both the players, respectively. One of the reasons behind such phenomena is that the PP-1 is a dominant left-hand player, and the IP is a right-hand dominant player. Moreover, particularly for this stroke, the player has to lunge forward his/her dominant leg or slight jump to play the stroke, so this can be one of the reasons for such phenomena. Another reason is that our video analysis observed that the PP-1 did not move around to perform the stroke. In contrast, the IP played the stroke with high precision and outplayed the other players. This further strengthens our hypothesis that the error metrics can be a good error representation approach for scoring a player during the game.

5.3. Score prediction module analysis

We propose a deep learning-based regression-based CNNs score prediction module to predict the score based on the players' performance. We employed traditional machine learning regression algorithms such as random forest regression, support vector machine regression, linear regression, generalized additive models as our baseline for score prediction. Moreover, we observed that the R^2 scores and *MSE* errors are insignificantly lower which is shown in Table 13. Such occurrence is due to the complexity of high speed and complex badminton activities. Another reason might be the simple architecture of the traditional algorithms. As a result, the algorithms are unsuccessful in capturing the high variance error features from the error input space. The score predictor CNNs-based algorithm has shown an improvement of $\approx$ **40%** compared to all the traditional regression algorithms. To develop our score predictor pipeline, we feed the same error

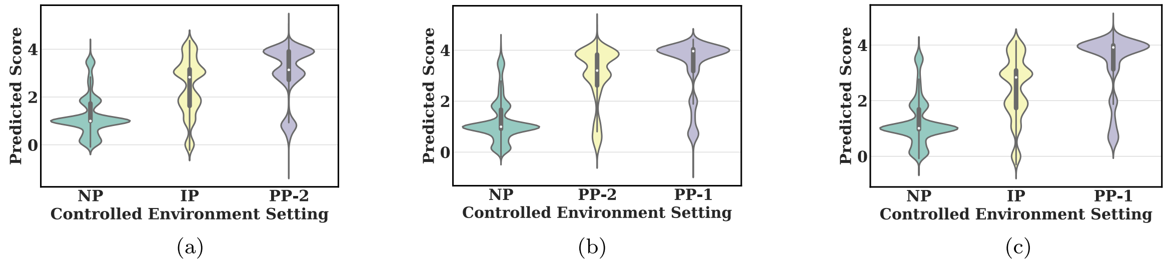


Fig. 14. Subfigures (a, b, c) demonstrate the violin plots for predicted score. In (a) highlight the predicted score of the players with respect to PP-1 (AP). Similarly, (b) highlight the predicted score of the players with respect to IP (AP) and (c) highlight the predicted score of the players with respect to PP-2 (AP).

Table 11
Success rate (%) of the players collected in a controlled environment setting.

Player	Success Rate(%)
PP-1	86.11%
PP-2	76.38%
IP	58.33%
NP	30.55%

Table 12
Success rate (%) of the players collected in an uncontrolled environment setting.

Player	Success Rate(%)
P-1	68.05%
P-2	58.05%
P-3	80.83%
P-4	45.61%
P-5	44.38%
P-6	74.72%
P-7	77.16%

Table 13
Comparison the traditional regression & CNNs algorithms for scoring module.

Algorithms	R^2
Linear Regression	0.1254
Linear Generalized Additive Model	0.2521
Expectile Generalized Additive Model	0.4212
Support Vector Regression	0.3476
Random Forest Regressor	0.4876
CNN Regressor	**0.8884**

vectors obtained from the distance-based error metrics pipeline as our input features to predict a score. We conducted our experiment for both data collection scenarios: controlled and uncontrolled environment settings. Our motivation for conducting our data collection in an uncontrolled environment is to validate our proposed scoring pipeline and predict the players' proficiency without prior knowledge of the players' skills in the game. Lastly, we plotted violin plots to exhibit the raw and obtained predicted score. The width of the violin plot represents the probability of the occurrence of the events. In our case, we plotted the raw and predicted score obtained by each player, where the width represented the probability of score obtained by each player and plotted on a scale of [0–4].

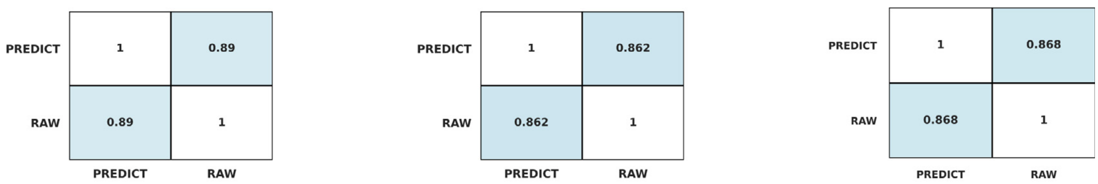
Figs. 14, we show the predicted score for each player by the proposed scoring predictor module for the players played in the controlled environment setting. We measure the performance of the automated scoring pipeline by computing two metrics, i.e. *RMSE* and R^2 *Score*, which are shown in Table 14. Interestingly, we observed that *RMSE* and R^2 *Scores* are low for the PP-1 and −2 compared to other combinations of players'. Such phenomenon occurred due to the effect of the dominant hand on both of the players, i.e. PP-1 is a left-hand player whereas PP-2 is a right-hand player. We are confident that our scoring pipeline can capture such phenomena and successfully be used to distinguish the proficiency and dexterities of the players. Moreover, by looking at the score predicted Figs. 14, we can undoubtedly say that NP performance is lower compared to the IP, PP-2 and PP-1. Moreover, to support our assumption, in figures (a, b, c), we noticed that the violin plots skewed towards lower values for the NP. In contrast, the PP-1 and PP-2 and IP values skewed

Table 14
Results of scoring module for controlled environment setting players.

Player	Mean Squared Error	R-Square Error
PP-1 & NP	0.2063	0.7941
PP-1 & IP	0.2154	0.7851
PP-1 & PP-2	0.3613	0.7615

Table 15
Results of scoring module for uncontrolled environment setting players.

Player	Mean Squared Error	R-Square Error
PP-1 & P-1	0.1417	0.8884
PP-1 & P-2	0.4777	0.7046
PP-1 & P-3	0.4947	0.6311
PP-1 & P-4	0.5750	0.4498
PP-1 & P-5	0.2585	0.8525
PP-1 & P-6	0.4407	0.6645
PP-1 & P-7	0.3477	0.7977



(a) (b) (c)

Fig. 15. Subfigures (a, b, c) demonstrate the bivariate correlation between the predicted score and handcrafted score for NP, IP and PP-2 players, respectively w.r.t PP-1 (AP).

towards higher values. It proves our hypothesis of collecting the dataset based on the players' proficiency in the controlled environment. Furthermore, in Figs. 14, we show three figures (a, b, c) in which we assumed three scenarios (same as our error metrics discussed above). The motivation is to determine the dependency and generalizability of the proposed scoring pipeline. In all three figures, the NP is most likely to get a low score than the other players. PP-1 and PP-2 show the same trend for the IP, i.e. IP performance was comparatively lower than the PP-1 and PP-2 but better than the NP. The performance of both PPs shows high scoring ability among the other players. Furthermore, to validate our scoring framework, we plotted the handcrafted score shown in Fig. 17 obtained by each player during the game. We observed that PP-1 tends to achieve a high score compared to the other players, and similarly, the NP is a vice-versa scenario. We noticed that the handcrafted scores for both the PPs skewed towards higher values and NP skewed towards lower values. We noticed the same trend in the predicted score shown in Fig. 14. Moreover, by referring to the predicted score, we can rank the players according to the score predicted in the game. Additionally, we noticed that the PP-1 and PP-2 achieved the highest score compared to the other players.

Moving forward, we evaluated the scoring predictor module by using *bivariate correlation matrix* between the predicted and handcrafted scores shown in Figs. 15. Furthermore, the correlation coefficient matrix shows that the raw and predicted scores are closely coupled in a robust linear relationship. It also illustrates the dependency and robustness of our automated score pipeline. Therefore, we are confident that our scoring pipeline can estimate the player's performance and rank the players based on their predicted scores. Moreover, we also propose a new success rate metric to re-evaluate the score predictor, the equation shown in (9). We defined the success rate based on the number of strokes played by the players correctly and precisely. The motivation behind defining a success rate metric is to showcase the highest probability of a player performing the perfect coordination of the limbs. In our case, we calculate the success rate for each player shown in Table 11. We noticed that both the PP-1 and PP-2 players have the highest probability of obtaining a high success rate than other players. Moreover, we can say that both the PPs in our case have the highest probability of obtaining a high predicted score from the proposed score predictor than the other players in the controlled environment setting.

Lastly, we collect a new sub-set dataset from volunteers under an uncontrolled environment setting. The motivation of the analysis is to investigate and evaluate our scoring pipeline, through which we are trying to predict the players' proficiency without any prior knowledge about their gameplay style and techniques. We compute two metrics for evaluation, i.e. *RMSE* and R^2 *Score* which are shown in Table 15. Furthermore, by looking at the values for each player for predicted values in Fig. 16 and the handcrafted score values in Fig. 18, we noticed that P-3, P-6 and P-7 shows similar trends of achieving high scores compared to the other players and similarly in both the plots we noticed the P-3, P-6 and P-7 values skewed towards higher values compared to other players. Moreover, both the players had similar styles and techniques during the data collection sessions and similar trends we found while assigning the handcrafted score to

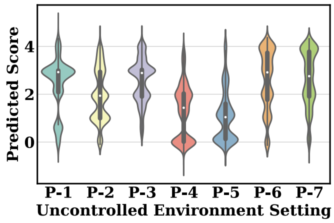


Fig. 16. Predicted score plot for the players' dataset collected in an uncontrolled environment setting w.r.t PP-1 (AP).

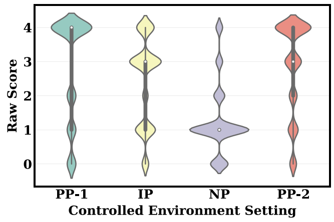


Fig. 17. Violin plots of handcrafted score corresponds to the players collected in a controlled environment setting.

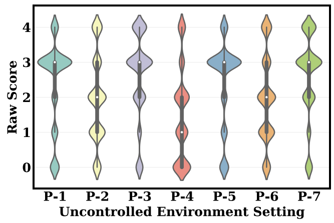


Fig. 18. Violin plots of handcrafted score corresponds to the players collected in an uncontrolled environment setting.

the respective players. Furthermore, we noticed that P-4 is the most inexperienced and incompetent player compared to other players; we observed the same trend through our data collection video analysis and also can be seen in both the plots shown in Figs. 16 and 18. To bolster our findings, we computed success rate metrics for each player performed in an uncontrolled environment setting results shown in Table 12. We obverse the same trend in the success rate value for P-3, P-6 and P-7 obtained the highest percentage of performing the stroke and stance correctly, whereas P-4 obtained the lowest success rate. Another phenomenon we encountered during our experimentation is that for P-3, the predicted score and R^2 *Score* are lower than the handcrafted score and success rate values. However, during our data collection sessions, we noticed that the player style was similar to P-1 and P-7. We believe that such occurrence is that PP-1 is left-handed, whereas P-3 is right-handed. Such occurrences might be one of the reasons for such phenomenon. We believe that comparing the players' of different gameplay styles (dominant limbs), we observed inconsistency and adverse effects on the obtained results and findings. Lastly, we are confident that the *DeCoach* framework can be employed to estimate and predict the performance of the players during the game.

6. Conclusion

In this work, we propose *time-based decay learning rate scheduler convolution neural networks (CNNs)-based classifier* for detecting 12 different strokes in *badminton* game. The proposed classifier outperformed the traditional machine learning algorithms by $\approx$ **15**%. We showcase an improvement of $\approx$ **8**% accuracy compared to the state-of-the-art algorithms. Furthermore, by leveraging the classified stroke label information, we designed a distance-based error learning metrics module to learn the ideal stance from the professional and further used it to learn the error between the professional player's ideal stance and the intermediate/novice player's stances. We have also identified the error metrics that can be used as the scoring metric for the game of badminton. Furthermore, through the *pdf* plots, we strengthened our hypothesis that the error metrics are a good representation of scoring the strokes and stances in the badminton game and be used as profiling metrics of the players' performance. After validating our hypothesis, we applied a feature selection approach to the computed error metrics to remove the redundancy error features in the computed error metrics. We collected *badminton activity recognition* (BAR) dataset in two different settings: controlled environment (prior knowledge of the competency of the players) and uncontrolled environment settings (no prior knowledge of the competency of the players). Our experiment showed that the professional players played the shots accurately and precisely compared to the other players. We adopted a CNNs-based regressor pipeline to build a robust scoring module that shows how the player played and predicts the performance for each player by capturing those distinct variations of their limbs. We obtained a high R^2 score of **88.84**% to predict each player's performance in the game. We compared the CNNs-based regressor pipeline to the traditional regression algorithms and showed that the proposed pipeline outperformed by $\approx$ **40**%. We proposed a *success rate* to determine the participant's performance during the data collection sessions. Furthermore, to determine the *DeCoach framework* robustness, generalizability and scalability characteristics, we proposed a *cross-person validation approach*. Lastly, to evaluate the scoring module, we employed a subset *BAR* dataset collected in an uncontrolled environment setting where we determined the participants' performance without any prior knowledge of their proficiency in the game. Lastly, through our extensive evaluation, we are confident that the proposed *DeCoach* framework can capture and quantify those minutes' erratic and distinct patterns from each limb and enable us to predict the player's performance. *Badminton Activity Recognition* [39] dataset and codes[1] are released.

7. Limitations & future work

In this study, we propose to assess the players based on the style of playing the shots played during the game. In order to achieve our goal, we devise a scoring methodology that can be employed to determine the players' performance during the game. However, one of the limitations of the scoring methodology is that we only consider the lower limb movements for the distance-based error learning module. Additionally, it also does not address the other factors that could affect a player's performance, such as the player's height, strength, swing speed, racket style, dominant limb, posture and many more. Moreover, the proposed error learning module has high computation overhead cost associated.

In future, we would like to investigate the player's body posture while playing a shot by collecting data from the torso position and exploring relevant more shift-invariant feature extraction mechanisms like RNNs/LSTMs, etc. We would also like to explore more sophisticated techniques such as policy-gradient reinforcement learning, imitation learning, etc., for building a model-based unsupervised scoring framework. Furthermore, we would like to investigate and expand our study to limb-based and user-based levels to determine the error. Also, in future, we would like to investigate more sophisticated approaches such as sample learning (paired sample), unsupervised (clustering approach) and deep metrics-based approach and compare them to the existing proposed framework. Besides, we would like to explore cross-player variations, cross-racket-based sports, badminton doubles and sensor device-related heterogeneity. Furthermore, we want to investigate the temporal correlation between the players' dominant upper and lower limbs. Lastly, we would like to deploy the proposed framework on a resource-constrained edge device for real-time inference.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is partially supported by the NSF CAREER Award # 1750936 and U.S. Army Grant # W911NF2120076. In addition, the authors would like to thank all our volunteers who provided the datasets playing in the badminton courts and open playground in the RAC (Retrievers Activities Center) at the University of Maryland Baltimore County.

References

[1] S. Ramasamy Ramamurthy, N. Roy, Recent trends in machine learning for human activity recognition—A survey, WIREs Data Min. Knowl. Discov. 8 (4) (2018) e1254.
[2] I. Ghosh, S.R. Ramamurthy, N. Roy, StanceScorer: A data driven approach to score badminton player, in: 2020 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops, IEEE, 2020, pp. 1–6.
[3] A. Chakma, A.Z.M. Faridee, N. Roy, H.M.S. Hossain, Shoot like Ronaldo: Predict soccer penalty outcome with wearables, in: 2020 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops, 2020, pp. 1–6.
[4] Actigraphcorp, https://www.actigraphcorp.com/.
[5] A.Z.M. Faridee, S.R. Ramamurthy, H. Hossain, N. Roy, HappyFeet: Recognizing and assessing dance on the floor, in: Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications, ACM, 2018, pp. 49–54.
[6] Zepp, http://www.zepplabs.com/en-us/.
[7] Coollang-Asia, http://www.coollang-global.com.
[8] Actofit, https://actofit.com/.
[9] C.T. Kiang, C.K. Yoong, A. Spowage, Local sensor system for badminton smash analysis, in: 2009 IEEE Instrumentation and Measurement Technology Conference, IEEE, 2009, pp. 883–888.
[10] J. Lin, C.-W. Chang, T.-U. Ik, Y.-C. Tseng, Sensor-based badminton stroke classification by machine learning methods, in: 2020 International Conference on Pervasive Artificial Intelligence, ICPAI, 2020, pp. 94–100, http://dx.doi.org/10.1109/ICPAI51961.2020.00025.

[1] https://github.com/indrajeetghosh/DeCoach.

[11] T. Steels, B. Van Herbruggen, J. Fontaine, T. De Pessemier, D. Plets, E. De Poorter, Badminton activity recognition using accelerometer data, Sensors 20 (17) (2020) 4685.
[12] M.A.I. Anik, M. Hassan, H. Mahmud, M.K. Hasan, Activity recognition of a badminton game through accelerometer and gyroscope, in: 2016 19th International Conference on Computer and Information Technology, ICCIT, IEEE, 2016, pp. 213–217.
[13] N.A. binti Rahmad, N.A.J. binti Sufri, M.A. bin As' ari, A. binti Azaman, Recognition of badminton action using convolutional neural network, Indonesian J. Electr. Eng. Inf. (IJEEI) 7 (4) (2019) 750–756.
[14] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[16] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
[17] A. Ghosh, S. Singh, C. Jawahar, Towards structured analysis of broadcast badminton videos, in: 2018 IEEE Winter Conference on Applications of Computer Vision, WACV, IEEE, 2018, pp. 296–304.
[18] G. Thomas, R. Gade, T.B. Moeslund, P. Carr, A. Hilton, Computer vision for sports: Current applications and research topics, Comput. Vis. Image Underst. 159 (2017) 3–18.
[19] E. Wu, T. Nozawa, F. Perteneder, H. Koike, VR Alpine ski training augmentation using visual cues of leading Skier, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2020.
[20] L. Shao, Research on sports training decision support system based on improved association rules algorithm, Secur. Commun. Netw. 2021 (2021).
[21] C. Zalluhoglu, N. Ikizler-Cinbis, Collective sports: A multi-task dataset for collective activity recognition, Image Vis. Comput. 94 (2020) 103870, http://dx.doi.org/10.1016/j.imavis.2020.103870.
[22] S. Liwei, Research on classification and recognition of badminton batting action based on machine learning, J. Intell. Fuzzy Systems 37 (5) (2019) 6241–6252, http://dx.doi.org/10.3233/JIFS-179206.
[23] L. Citraro, P. Márquez-Neila, S. Savare, V. Jayaram, C. Dubout, F. Renaut, A. Hasfura, H.B. Shitrit, P. Fua, Real-time camera pose estimation for sports fields, Mach. Vis. Appl. 31 (3) (2020) 16, http://dx.doi.org/10.1007/s00138-020-01064-7.
[24] W.-T. Chu, S. Situmeang, Badminton video analysis based on spatiotemporal and stroke features, in: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, ACM, 2017, pp. 448–451.
[25] T. Jaitner, W. Gawin, Analysis of badminton smash with a mobile measure device based on accelerometry, in: ISBS-Conference Proceedings Archive, 2007.
[26] H. Zhao, S. Wang, G. Zhou, W. Jung, TennisEye: tennis ball speed estimation using a racket-mounted motion sensor, in: Proceedings of the 18th International Conference on Information Processing in Sensor Networks, ACM, 2019, pp. 241–252.
[27] S.-T. Choe, W.-D. Cho, J.-H. Kim, K.-H. Kim, Reducing operational time complexity of k-NN algorithms thin; clustering in wrist-activity recognition, Intell. Autom. Soft Comput. 26 (4) (2020) 679–691.
[28] A.-M. Sevcenco, K.F. Li, K. Takano, Collection and classification of tennis swings using a virtual racket, in: 2012 Fourth International Conference on Intelligent Networking and Collaborative Systems, IEEE, 2012, pp. 47–54.
[29] Z. Wang, M. Guo, C. Zhao, Badminton stroke recognition based on body sensor networks, IEEE Trans. Hum.-Mach. Syst. 46 (5) (2016) 769–775.
[30] L. Kong, D. Huang, J. Qin, Y. Wang, A joint framework for athlete tracking and action recognition in sports videos, IEEE Trans. Circu. Syst. Video Techn. 30 (2) (2020) 532–548, http://dx.doi.org/10.1109/TCSVT.2019.2893318.
[31] I. Ghosh, DeepMinton: Analyzing stance and stroke to rank badminton players, UMBC Stud. Collect. (2020).
[32] L. Pappalardo, P. Cintia, P. Ferragina, E. Massucco, D. Pedreschi, F. Giannotti, PlayeRank: data-driven performance evaluation and player ranking in soccer via a machine learning approach, ACM Trans. Intell. Syst. Technol. (TIST) 10 (5) (2019) 59.
[33] H.S. Hossain, M.A.A.H. Khan, N. Roy, SoccerMate: A personal soccer attribute profiler using wearables, in: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops, IEEE, 2017, pp. 164–169.
[34] A. Botchkarev, Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology, 2018, arXiv preprint arXiv:1809.03006.
[35] N. Lopes, B. Ribeiro, On the impact of distance metrics in instance-based learning algorithms, in: Iberian Conference on Pattern Recognition and Image Analysis, Springer, 2015, pp. 48–56.
[36] B.B. Nair, M. Patturajan, V. Mohandas, R. Sreenivasan, Predicting the BSE sensex: Performance comparison of adaptive linear element, feed forward and time delay neural networks, in: 2012 International Conference on Power, Signals, Controls and Computation, IEEE, 2012, pp. 1–5.
[37] ShimmerSensing, http://www.shimmersensing.com/.
[38] Akasotech, https://www.akasotech.com/.
[39] I. Ghosh, S. Ramasamy Ramamurthy, A. Chakma, E. Dey, Z. Hasan, N. Roy, Badminton Activity Recognition (BAR), IEEE Dataport, 2020, http://dx.doi.org/10.21227/n1e0-7c60.