Performance Optimization: AI Engineering

Jessica, Postgraduate Student, MBA – Business Analytics, School of Business, Woxsen University

Introduction

In the ever-evolving world of sports, the difference between victory and defeat often comes down to the finest margins. As athletes and teams strive to gain a competitive edge, **AI Engineering** is emerging as a game-changing force in **performance optimization**. By merging advanced data analytics, machine learning algorithms, and real-time sensor technologies, AI enables a deeper understanding of athlete performance, recovery, and strategy.

From personalized training plans powered by predictive analytics to injury prevention through biomechanical analysis, AI transforms raw data into actionable insights. Coaches and sports scientists can now monitor fatigue levels, analyze opponent patterns, and simulate game scenarios with unprecedented precision. As a result, athletes train smarter and compete at levels that were once thought unattainable.

This intersection of **AI and sports science** marks a new era—one where technology doesn't just support athletes, but actively enhances their potential. Welcome to the future of athletic performance, engineered by AI.

The Project

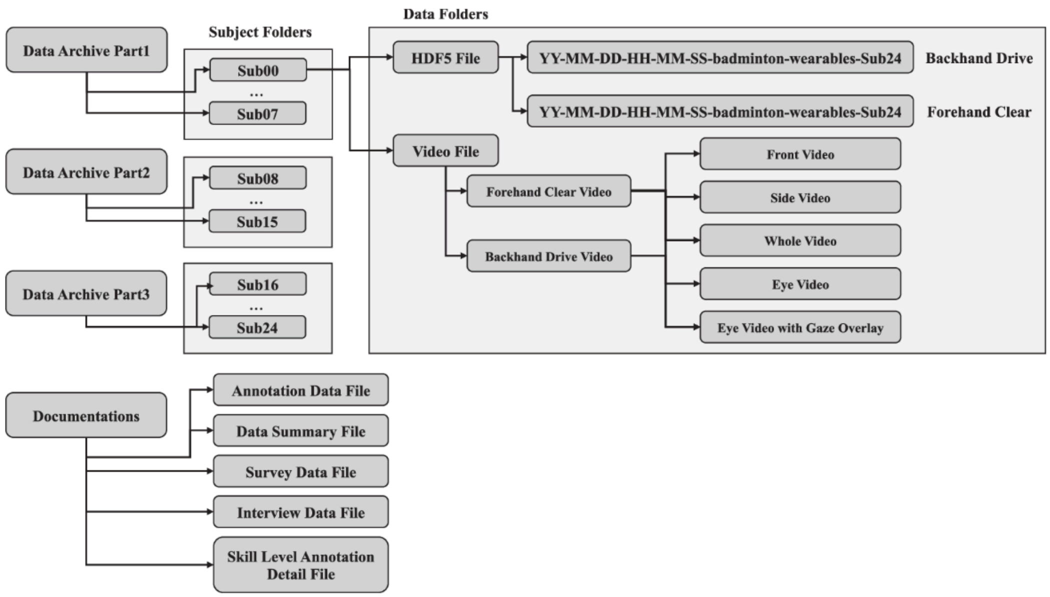
"AI-Driven Performance Optimization System for Elite Athletes"

To develop an AI-powered platform that collects, analyzes, and interprets athlete performance data to deliver **personalized training, injury risk alerts, and tactical insights** to optimize overall athletic output.

Dataset

The **MultiSenseBadminton** dataset is a rich multimodal resource for badminton training and analysis, featuring over 7,700 swing samples from players across skill levels. It includes synchronized data from wearable sensors (eye tracking, EMG, foot pressure, and motion capture), annotated video recordings, and metadata such as stroke types and hitting locations. This dataset is ideal for building AI models that assess skill level, predict injury risks, or give biomechanical feedback.

From: MultiSenseBadminton: Wearable Sensor–Based Biomechanical Dataset for Evaluation of Badminton Performance



Hierarchical folder structure of the dataset.

Each subject in the project has sensor-data HDF5 files labeled with the date of data collection and the participant ID. The HDF5 files can be easily accessed using HDF5 viewer software, and the Python code for reading these files is available on the project's GitHub repository. The HDF5 file contains sensor stream data and Unix time.

The annotation file records the Unix time, stroke number, and annotation value in five levels, providing detailed information about various aspects of badminton strokes. The survey data and data-summary files contain information on all participants, such as demographic information and questionnaire responses, and can be used to understand the participants' characteristics and study findings.

The **MultiSenseBadminton** dataset is categorized into several key data streams, each capturing different biomechanical and performance aspects:

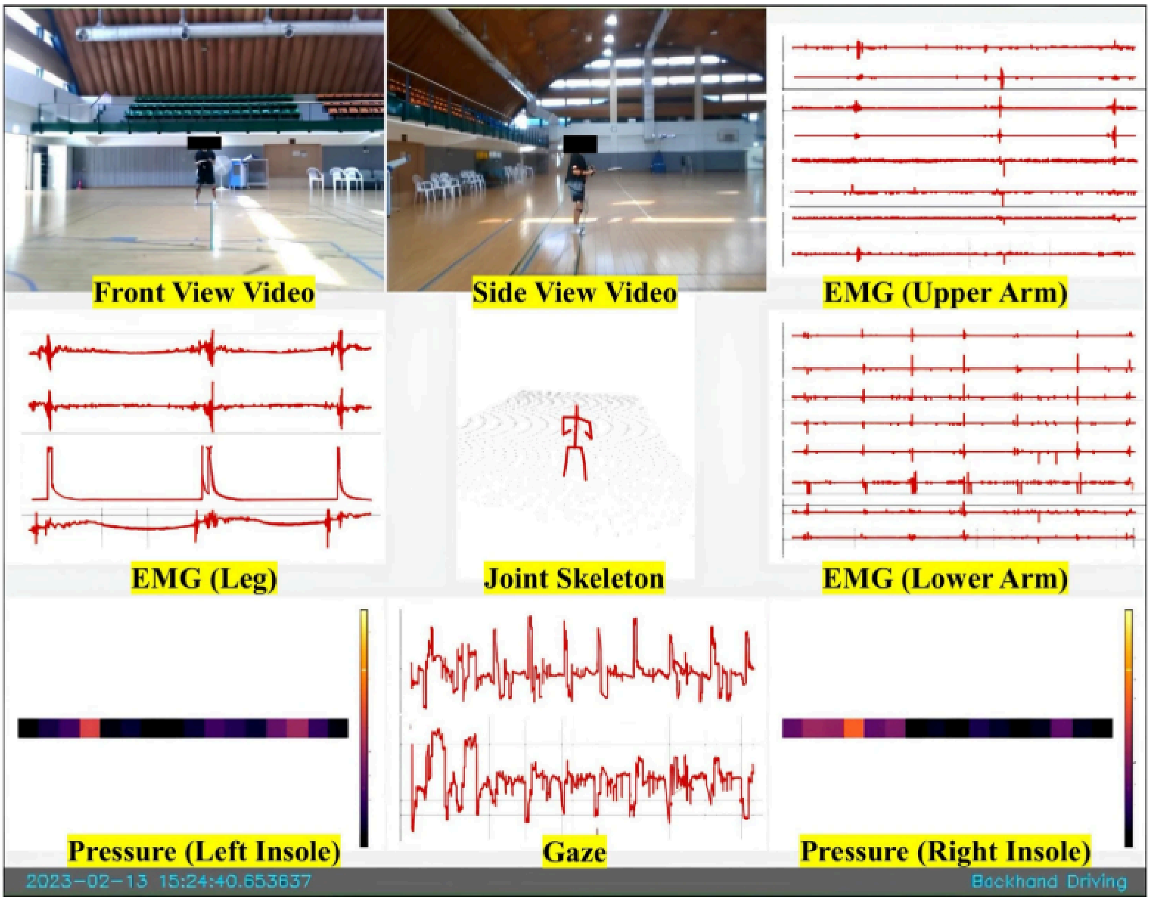
1. **Motion Data** – Captured via inertial sensors on limbs to track body movement during swings.

2. **Electromyography (EMG)** – Muscle activity data from key upper limb muscles.

3. **Foot Pressure Data** – Ground contact and pressure patterns from insole sensors.

4. **Eye Tracking** – Gaze and focus data during play.

5. **Annotations** – Stroke type, player skill level, and hit location.



HDF5 file details

The HDF5 file contains the following information: EMG values of leg, forearm, and arm start and stop time of strokes and calibration, eye gaze, foot pressure data, and motion capture data. Each sensor data structure is composed of “data”, “time s”, and “time str”. The “data” represents sensor data for each channel, “time s” represents Unix time for each data, and “time str” represents the global time for each data. The following subsections introduce the structure and composition of the dataset.

cgx-aim-leg-emg

This dataset encompasses EMG values for the dominant leg, expressed in millivolts (mV), across four distinct channels. Each channel serves to characterize specific muscle information, with the unit being mV: channel 1 is designated for the rectus femoris; channel 2 corresponds to the vastus medialis; channel 3 is allocated for the vastus lateralis; and channel 4 pertains to the biceps femoris.

experiment-calibration

The calibration dataset, with its two channels, relates to EMG data for the arm, forearm, and leg. The first channel marks the beginning and end of calibration. The second channel identifies the calibration pose used, either for gforce, involving three specific arm motions, or leg EMG, with two leg motions. The calibration pose for forearms includes lower arm inward pose and lower arm outward. And the calibration pose for the arm includes the upper arm inward pose. And the calibration poses for the leg include the leg forced pose and squat pose. The primary aim of this calibration is to obtain the maximum EMG value, aiding in the proper normalization and calibration of subsequent EMG measurements.

eye-gaze

This dataset contains two types of data: gaze data and worn data. The gaze data is designed to capture eye-tracking data through the use of pupil-invisible glasses and is artfully constructed into two channels. The first channel precisely maps the X-coordinate of gaze positions, spanning from 0 to 1088. This span aligns with the horizontal resolution of the video, facilitating accurate monitoring of gaze movements across the horizontal axis. The second channel records the Y-coordinate of gaze positions, extending from 0 to 1080, reflecting the vertical resolution of the video. This enables a thorough examination of vertical gaze movements. In addition to these channels, the dataset includes a “worn” column. This column is designed to indicate whether the glasses were worn during the data capture, with a value of 1 denoting the glasses were on and a value of 0 indicating their absence. This binary value provides a clear indication of the presence or absence of the glasses, thereby informing the status of data prediction during the capture process.

gforce-lowerarm-emg

This dataset furnishes EMG values for the lower arm, represented in normalized units ranging between 0 and 250 across eight channels.

gforce-upperarm-emg

Analogous to the gforce lower arm dataset, this collection entails EMG values for the upper arm, comprising eight channels, with data normalized between 0 and 250.

moticon-insole

The Moticon insole data contains five types of data. The first type of data is the Center of Pressure (COP). It is represented by four channels denoting the x and y coordinates of COP for both feet. The first channel denotes the COP x coordinates of the left foot, the second channel denotes the COP y coordinates of the left foot, the third channel denotes the COP x coordinates of the right foot, and the fourth channel denotes the COP y coordinates of the right foot. The second type of data is the acceleration. It captures the linear acceleration of the foot across the x, y, and z directions, expressed in g (gravity). The first channel is the acceleration of x coordinates, the second is the acceleration of y coordinates, and the third is the acceleration of z coordinates. The third type of data is the angular velocity. It captures the angular velocity of the foot across the x, y, and z directions, measured in degree/s. The first channel is the angular velocity of x coordinates, the second is the angular velocity of y coordinates, and the third is the angular velocity of z coordinates. The fourth type of data is the pressure. It captures the pressure maps around the foot in 16 channels, measured in N/cmÂ². The fifth type of data is the total force. It captures the total force of each foot in 1 channel, measured in Newton (N).

pns-joint

This dataset contains four types of data: information relative to joint global positions, joint local positions, quaternions, and Euler angles, encompassing the following parameters for 21 joints. And the order of the joints within the dataset is detailed as follows: hip, right up leg, right leg, right foot, left up leg, left leg, left foot, spine, spine 1, spine 2, neck, neck 1, head, right shoulder, right arm, right forearm, right hand, left shoulder, left arm, left forearm, left hand.

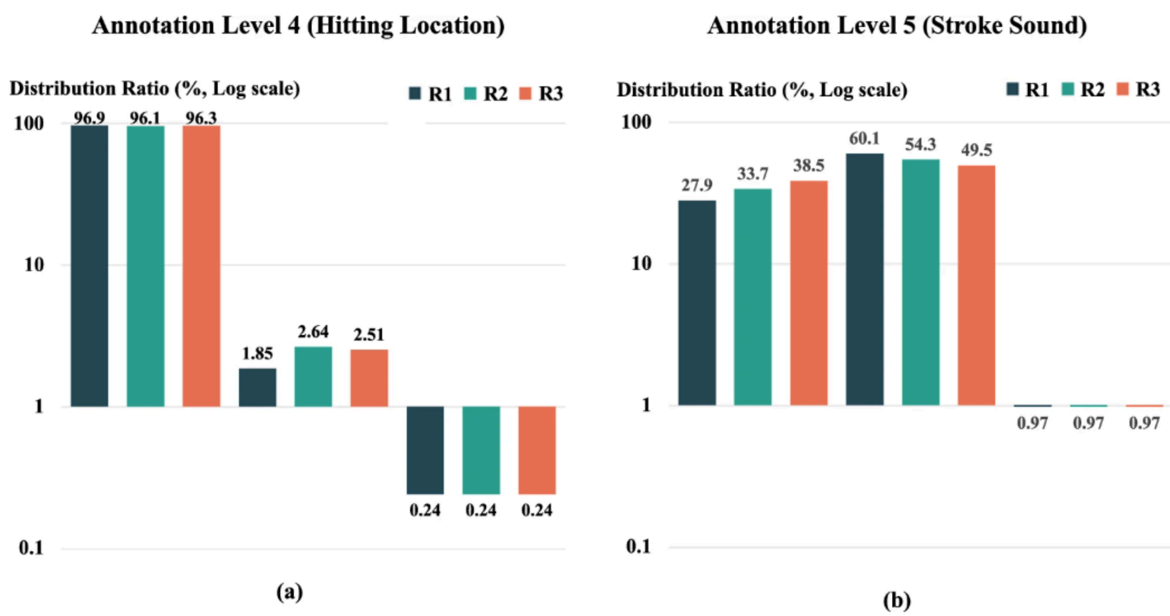
The first type of data pertains to the local position of each joint, measured in centimeters (cm). This data is structured across a total of 63 channels, comprising the x, y, and z coordinates for each joint. The second type of data pertains to the global position of each joint, measured in centimeters (cm). This data is structured across a total of 63 channels, comprising the x, y, and z coordinates for each joint. The third type of data pertains to the Euler angle of each joint, measured in degrees. This data is structured across a total of 63 channels, comprising the x, y, and z coordinates for each joint. The fourth type of data pertains to the quaternion of each joint, measured in degrees. This data is structured across a total of 84 channels, comprising the w, x, y, and z values for each joint.

Table 5 Skill level annotation; # means the number of subjects.

From: MultiSenseBadminton: Wearable Sensor–Based Biomechanical Dataset for Evaluation of Badminton Performance

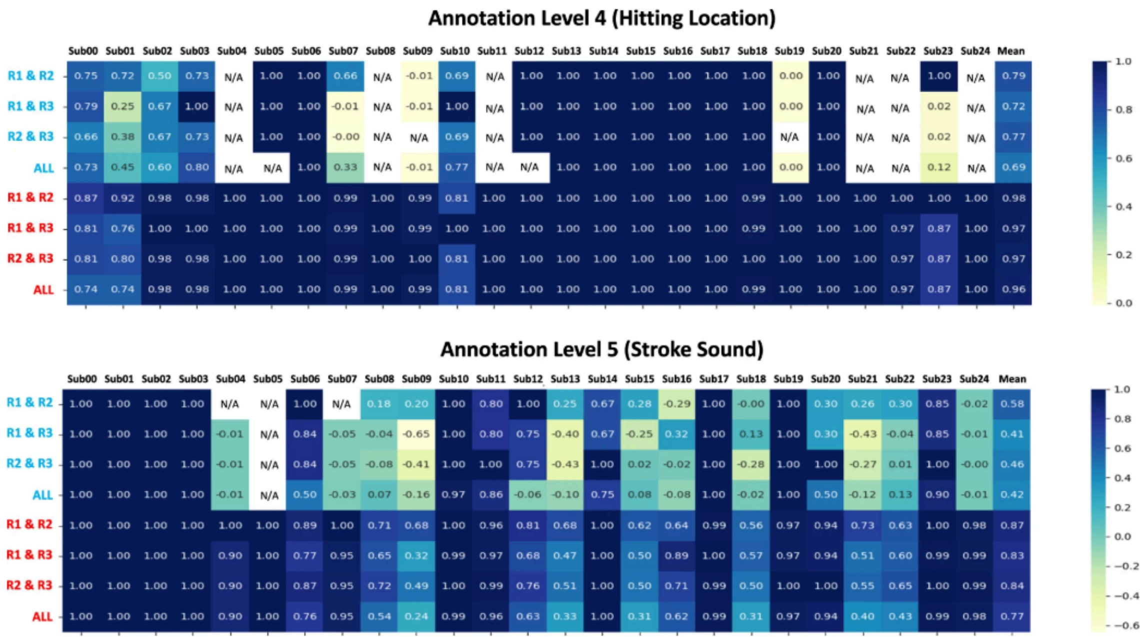
Stroke	# Beginner	# Intermediate	# Expert	IRR (R1&R2)	IRR (R1&R3)	IRR (R2&R3)	IRR (ALL)
Clear	11	8	6	0.70	0.63	0.59	0.64
Drive	11	9	5	0.82	0.82	0.64	0.75

The distribution and frequency of annotations are shown in Fig. 10a,b, where the term “Not Contact” indicates that the participant did not make contact with the ball. The total number of stroke instances is 7,763 and we expressed each label as a percentile ratio.



Distributions and frequencies of Annotation Levels 4 and 5 from three raters (R1, R2, R3): Not Contact indicates the shuttlecock and the racket do not make proper contact. The graph is designed with a logarithmic scale on the y-axis to account for data imbalance.

Below figure displays a heat map of Krippendorff's alpha coefficients and percentage of agreement calculated for three annotation levels measured on an ordinal scale. Annotation Levels 4 and 5 had high agreement percent (0.96 and 0.77, respectively), indicating a good agreement score. However, when considering Krippendorff's alpha value, Annotation Level 5 received a low score of 0.42. This can be attributed to the subjective nature of the task, where raters assessed the quality of a sound as Good, Maybe, or Bad. The final annotation data files for levels 4 and 5 contain both the consolidated annotation values that received agreement from multiple raters and the raw annotation values provided by each individual rater.



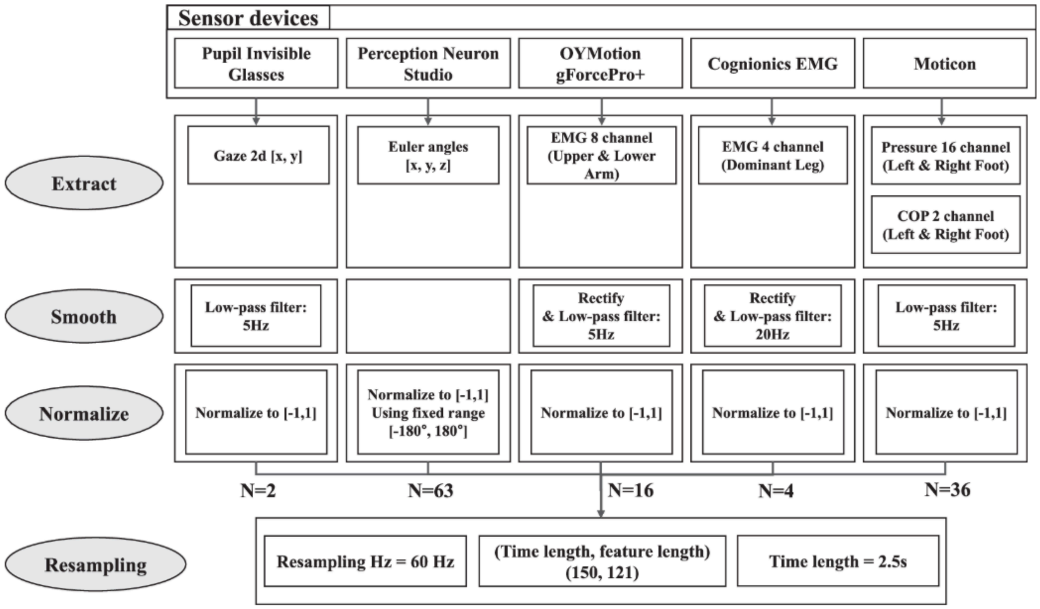
Heat maps of inter-rater reliability measured by Krippendorff's alpha and percentage of agreement for each subject: The first four rows (blue text) indicate Krippendorff's alpha value, while the last four rows (red text) indicate the percentage of agreement. The columns show the respective metrics for each participant, and R1, R2, and R3 represent the raters.

Preliminary learning pipelines

In this section, we provide an initial pipeline for utilizing our multi-modal data, aiding easier use of our dataset for non-expert developers. Classification results for each annotation are demonstrated with an state-of-the-art neural network model. This structure can be tailored to fit individual researchers' unique hypotheses and objectives. Much of the pipeline's foundation is drawn from the ActionSense dataset76, from which we adapted preprocessing and analysis methodologies. The experimental methods and the results for dataset validity are elaborated are provided in the following sections.

Data preprocessing and feature extraction

We derived six types of features from five distinct types of wearable sensors. These features encompassed gaze 2D data obtained from Pupil Invisible Glasses, joint Euler angles extracted from Perception Neuron Studio, EMG data collected from gForcePro + and Cognionics EMG sensors, as well as pressure and center of pressure data acquired from Moticon sensors. To synchronize these features, each sensor was integrated into the main computer's data collection framework. This involved fetching the Unix time and data from each sensor server and transmitting these simultaneously. Based on this approach, the Unix time and sensor data were collected simultaneously for five different sensors. We performed preprocessing for each sensor, and the preprocessing process is summarized in Fig. 12. To facilitate further research in this area, the code used in this study for data preprocessing and classification was published on GitHub as an open-source project.



Data Preprocessing.

In the preprocessing stage, we extracted features from five sensors: 2 channels of 2D gaze data, 63 channels from 21 joint Euler angles, 16 channels for upper and lower arm EMG, 4 channels for leg EMG, and 36 channels from insole sensors, which include both 32 pressure channels and 4 COP channels. To reduce noise and artifacts in the data, we applied a low-pass filter with cut-off frequencies of 5 Hz for the arm EMG, 2D gaze and insole pressure, and 20 Hz for the leg EMG. The EMG value was rectified by taking its absolute value before applying a low-pass filter. Altogether, this process generates a total of 121 data channels, with each channel normalized within the range of $[-1, 1]$. Subsequently, we resampled all channels to a uniform time vector at a 60 Hz sampling rate, employing linear interpolation throughout to ensure consistent temporal alignment across the data. By segmenting each stroke example with a 2.5-second interval, we extracted a total of 7761 stroke instances from 18 participants. This total includes 2607 instances of backhand drive, 2613 instances of forehand clear, and 2541 instances of non-strokes, where non-strokes refer to instances where no stroke was being performed. Utilizing our stroke dataset, we proceeded to classify the five types of annotations we collected; types of strokes, skill level, horizontal location, vertical location, hitting point, and sound.

Network architecture

For our training process, we constructed models using deep learning architectures commonly employed for time-series data: ConvLSTM, Long Short-Term Memory (LSTM), and Transformer. These models were developed to effectively capture and analyze the temporal patterns inherent in our dataset. Our pipeline, implemented in Python 3.9 using Pytorch, utilized these architectures to accurately extract key features from sensor data for sequential information classification. We consistently applied these architectures across the classification of annotation data.

To evaluate the model's performance, we used Accuracy, Balanced Accuracy, and F1 score as metrics, with the Adam optimizer and categorical cross-entropy as the loss function. The model's performance was assessed using hyperparameters: learning rates of 0.0005 and 0.0001, and epochs of 200. We applied early stopping with a patience of 10. The dataset division into train + and test sets involved two validation methods: 10-fold cross-validation, where the dataset is split into ten equal parts, each part used once as a test set while the others serve as training data, ensuring all data is used in both roles; and leave-three-out (LTO) cross-validation, which involves selecting one subject from each skill level - beginner, intermediate, and expert - and using their data as the test set, while data from other subjects form the training set. Detailed reference numbers for subjects used in the LTO cross-validation, which demonstrate the unique combinations of participants from different skill levels–beginner, intermediate, and expert–across ten iterations, are listed in Table 6. This approach tests the model's ability to generalize to new subjects across different skill levels. Additionally, for comparison, we developed a baseline model that predicts the predominant class, serving as a benchmark to assess our models' performance and effectiveness against this basic approach.