IMPERIAL

DESE71003 – Sensing and Internet of Things

Imperial College London

Dyson School of Design Engineering

KiAI: An IoT Based Kendo Training Assistant for Real-Time Motion Analysis

Author:
Raphassit (Beam) Suwiwatchai (CID: 02026376)

Code & Data URL:
https://github.com/beam-su/SIOT-KiAI

Coursework Video URL:
https://youtu.be/nLtwH57EI44

Coursework Video URL (alternatively):
https://github.com/beam-su/SIOT-KiAI

Date: December 12, 2024

KiAI: An IoT Based Kendo Training Assistant for Real-Time Motion Analysis

Sensing and Internet of Things

Raphassit Suwiwatchai 02026376
Dyson School of Design Engineering
Imperial College London, UK
https://github.com/beam-su/SIOT-KiAI
https://youtu.be/nLtwH57EI44

***Abstract*—Kendo, a traditional Japanese martial art emphasising the unity of spirit, sword, and body (*Ki-Ken-Tai-Ichi*), requires practitioners to develop precise coordination and control. However, refining these skills often depends on expert guidance, which may not always be accessible. This means that solo-practitioners lack tools that offers personalised, driven insights, particularly regarding motion smoothness and classification of kendo techniques. This gap highlights the need for a system that can provide actionable feedback while respecting kendo's principles of focus and discipline. This report introduces *KiAI*, an Internet of Things (IoT)-based kendo training assistant that integrates motion sensors, environmental monitors, and real-time video feedback. The system uses MPU-6500 sensor to collect acceleration and gyroscope data at 1Hz, combined with machine learning techniques (Support Vector Classifier) to classify kendo moves and assess performance metrics such as motion smoothness, jerk, and acceleration. Our classification model, trained on four key kendo techniques (*men*, *kote*, *do*, *kamae*), achieved an accuracy of 87.5% on unseen test data. The results demonstrate that IoT and machine learning can effectively analyse and enhance kendo practice, providing users with data-driven feedback to refine their technique. Future developments, such as integrating expert-labelled datasets and exploring competition-compatible tools like image recognition, could further advance kendo training while preserving its core philosophies.**

I. Introduction

KENDO is a Japanese martial art which evolved from the swordsmanship exercises of samurai warriors [1]. Over time, it was adapted into a sport in the Meji era, to use bamboo swords (called *shinai*) and protective armour to enable safe sparring and competition. The term *kendo*, which directly translates to *"The way of the Sword"*, encapsulates not just a martial discipline but a core philosophical pursuit, with its emphasis on the concept of *Ki-Ken-Tai-Ichi*- the unity of spirit, sword, and body [2].

Therefore, the mastery of Kendo is more than just raw physical strength, requiring precise coordination, balance, and rhythm. However, refining these intricate movements can be challenging, particularly without expert guidance, hindering a practitioner's ability to improve.

A. Objectives

Recognising that advancements in Internet of Things (IoT) systems and machine learning can be used to help practitioners gain real-time insights into their techniques, this project introduces *KiAI*, an IoT-based kendo training assistant. *KiAI* enables users to monitor key performance metrics such as smoothness, speed, acceleration, and jerk in their movements, offering actionable feedback to refine their techniques. The primary design objectives are as follows:

Objective	Parameters
Measure and record sword movements using an accelerometer	3-Axis Acceleration (ms^{-2}), 3-Axis Gyroscopic Readings
Measure and record environmental data to reduce noise complaints and mould growths on bamboo/wooden swords	Volume (dB), Humidity (%), Temperature (C°)
Livestream the training session using cameras	ESP32-CAM (fps)
Utilise machine learning to recognise move-sets & analyse the motion smoothness	Move-set Classification Model
Interactive Web Application with data visualisation & insights	N/A
LINE (Japanese messaging app) integration	N/A

TABLE I: Project Key Objectives

Building upon these objectives, a comprehensive IoT system was successfully developed, enabling users to effectively practice and refine their kendo forms. It is to be noted that the sword used is a *bokken* not a *shinai*. This is due the *shinai* being longer in length, which would hit the ceiling of my room.

II. Component 1: Sensing

While the project is roughly split into 2 parts (Sensing & Internet of Things), the first step in the development is to build the system architecture to understand the end-to-end system characteristic.

A. Sensing System Architecture

With the design objectives defined, the first critical component of the system is data sensing and collection architecture. This involves using sensors and microcontrollers to capture real-time motion and environmental data for further analysis to provide actionable insights to users.

Objective	Requirements
Real-time collection of motion, environmental conditions, and video data	High sampling rates for motion data to capture rapid movements
Integration of variables to provide actionable insights into kendo performance	Low-latency streaming for video feedback
Efficient & scaleable data storage & retrieval for analysis & visualisation	Environmental sensing

TABLE II: Sensing System Key Objectives

To ensure efficiency, scalability, and ease of development, the sensors are partitioned into independent modules, with each uploading its data separately to the InfluxDB cloud storage. During development, various database services were considered, however many (such as *ThingSpeak*) impose limitations on upload frequency and incur higher costs, making them unsuitable for capturing the rapid movements inherent in kendo practice. Additionally, InfluxDB's support for plug-ins with third-parties, such as *Grafana*, makes it easily scaleable in data visualisation.

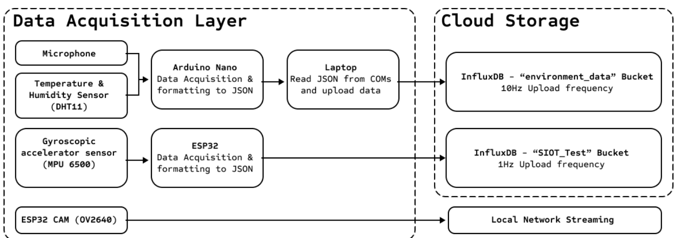


Fig. 1: Data Collection System Architecture

1) Motion Tracking Component: The system's sensing component begins with the MPU-6500, which measures the user's sword motion through acceleration and angular velocity. In the context of this project, the motion data is important in two ways: the identification of user moves (move classification) and the calculation of key performance metrics (e.g. peak acceleration, motion smoothness, jerk).

This sensor is ideal for the application due to its integration of a 3-axis accelerometer with a 3-axis gyroscope to provide 6-axis motion tracking in a compact package. Most importantly, to accurately capture the fast and dynamic characteristics of kendo strikes, its maximum sampling rate of 1kHz ensures real-time collection. Additionally, its low power consumption ($3.5mA$) makes it well-suited for prolonged practice sessions, enhancing its practicality.

Initially the MPU-9250 was also considered as its in-built magnetometer would reduce drifts in the calculations of angles (roll, pitch, yaw). However, since the usage duration of KiAI is relatively short, the accumulated error is minimal and has little impact on the overall output. Ultimately, the MPU-6500 was chosen to minimise power consumption and cost.

On the microcontroller aspect, the ESP32's Wi-Fi connectivity and great energy efficiency allows it to precisely capture quick sword movements at a high frequency before uploading the data to a cloud bucket with minimal latency for longer periods of time. Leveraging the ESP32's processing power, cloud computing resources and bandwidths are further optimised by calculating derived metrics, such as roll and pitch on the edge.

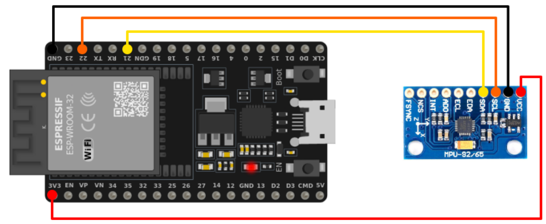


Fig. 2: MPU-6500 Data Collection Pinout Diagram

During system development, the sensing components underwent several iterations, particularly in optimising its sampling rate and determining the number of MPU-6500 sensors to use. While the sensor itself supports $1kHz$ sampling rate, the average tested latency between London (user location) to the InfluxDB data centre (us-east-1) was measured at $0.3s$. This constraint meant that either the ESP32 would have to batch write data or limit its sampling rate to $3Hz$. The latter option was selected for simplicity and reliability, where the sampling rate was further reduced to $1Hz$ to provide an additional buffer for network variability.

Initially, the design also incorporated a second MPU-6500 sensor, one for each hand, to better capture the user's movements. However, during the machine learning classification training—discussed in later section—the performance of this dual-sensor configuration was found to be inferior to a single-sensor setup.

2) Environmental Condition Component: In this sensing component of the system, DHT11 temperature & humidity sensor was used in tandem with a microphone module. While not directly linked to performance, these address noise complaints from neighbours during kendo practice and the risk of mould growth on swords due to high humidity—a problem often faced by Imperial Kendo Club. Here, the data structure being sent to the InfluxDB is humidity (%), temperature ($C°$), and volume (dB). Following the principle of least privilege, the data from the microphone is not recorded as it is not needed.

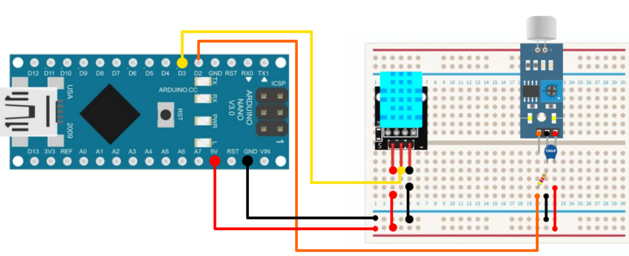


Fig. 3: Environmental Conditions Monitoring Pinout Diagram

For monitoring humidity and temperature, the DHT11 was chosen due to its affordability and sufficient accuracy for the design requirements. While less precise than alternatives like the DHT22, its accuracy and range are sufficient for a typical indoor environments. Additionally, as the data in this application (monitoring the training environment) does not require high-frequency sampling, the DHT11's maximum sampling rate of $1Hz$ [3] is more than adequate in this specific use-case.

On to the microcontroller choice, as the noise sensor module require $5V$ power, the Arduino Nano 33 IoT is an ideal choice for its $5V$ pinout and its built-in Wi-Fi connectivity. However, budget constraints limited the microcontroller choice to what I have on-hand, being Ardiuno Nano Every. As this specific board does not have Wi-Fi capabilities, the board needed to be connected to a computer using micro USB to allow the computer to read the data, parse the data to JSON, and upload the data to be stored on InfluxDB.

Focusing on the noise sensor module, a 104 capacitor ($0.1F$) was placed between the V_{in} and ground pins as a decoupling capacitor to stabilise voltage and reduce power supply noise that could affect sensor readings. Additionally, a $4700k\Omega$ resistor was added between the ground and D_{out} pin to provide pull-down functionality; ensuring that the output signal of the sensor defaults to low to prevent erratic behaviour from floating signals. The sensor's sensitivity was then calibrated iteratively using the module's potentiometer to set the noise volume limit for the system.

3) Video: The video streaming functionality was implemented using an ESP32-CAM board with an OV2640 camera, providing a real-time stream of the practice session over the local network. In this application, the JPEG frame is the data being updated. Since the web app is hosted on the cloud, the camera's local connection acts as a security measure, ensuring that the stream is accessible only within the local network, preventing unauthorised external access.

The video streaming script is designed for scalability and efficiency, dynamically detecting PSRAM availability to optimise performance across different ESP32-CAM variants. With PSRAM, the system utilises additional memory to enable higher JPEG quality, larger frame buffers, and more responsive streaming.

B. Sensing System Analysis

The decision to partition each sensing component offers several advantages, enhancing the system's efficiency and maintainability. The suitability of each component, based on its sensitivity and sampling rate, has been detailed in the previous section.

More importantly, the separation of processes aligns with the concept of modular design, where a complex system is broken down into smaller, manageable, and self-contained units. Not only does this help with the development process in decreasing debugging times and allowing for easy scaleability, it also allows greater ease in characterising the system's logic and complexity. From this, we can build our sensing complexity analysis.

The *Big O Complexity Analysis* shown in Table III confirms the practicality of separating the sensing components. While the dominant processes of all components exhibit linear complexity, the larger input size and higher computational demands of video data ($O(m)$) would overwhelm system resources if all sensors operated on a single microcontroller. This is evident when comparing data sizes: motion and environmental sensors generate relatively small numerical and JSON payloads, whereas video produces continuous high-bandwidth streams. These differences would lead to increased latency and resource contention in a shared system. By distributing tasks across three microcontrollers, the workloads are effectively isolated, allowing for independent optimisation and ensuring the real-time responsiveness necessary for kendo training.

Sensor Data Collection	**State Machine Representation**	**Big O**
Motion	Initialisation, Idle, Data Processing, Data Transmission	$O(n)$
Environmental Data	Initialisation, Data Collection, Data Reception, Data Parsing, Data Transmission	$O(n_1 + n_2)$, where n_1 = JSON length & n_2 = # of fields written to database
Video	Initialisation, Wifi Connection, Web Server Setup, Idle	$O(m)$

TABLE III: Data Collection & Storage Complexity Analysis

III. Component 2: Internet of Things

In the sensing component of the project, all data except video is being actively collected. Additionally, for the environmental data from the noise and the DHT11 sensors there are limited data interaction that could be done. As the data is being gathered indoors, meaning there is limited fluctuations and variations between temperature and humidity, integration of weather-related APIs for forecasting wouldn't add much value to the system. Therefore, these sensors, while essential for monitoring noise levels and storage conditions, provide relatively straightforward data output and require minimal analysis at this stage.

Therefore, the focus shifts to the motion data collected using the MPU-6500 sensor. For practical purposes, an additional Arduino and Python script were written to read and store the data locally as a CSV file for convenience during analysis. Since this data is intended for training the Kendo Move Classification Model, it is crucial that the sampling rate is a multiple of the cloud upload frequency of $1Hz$, mentioned in the previous section, ensuring compatibility and facilitates windowing during time-series analysis.

From this sensing components, there are 4 main variations of kendo strikes we are intending to collect: *men* (Head Strike), *kote* (Hand Strike), *do* (Body Strike), and *kamae* (Guard Stance). Ideally, we would want to collect all of the kendo moves to create a more comprehensive classification model, but due to the limitations of my skill, this was omitted.

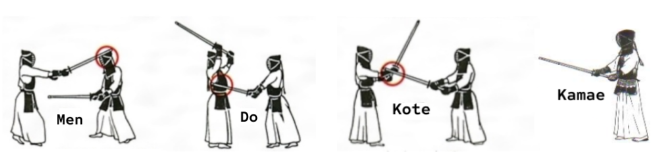


Fig. 4: Four Basic Kendo Moves of Interest [4] [5]

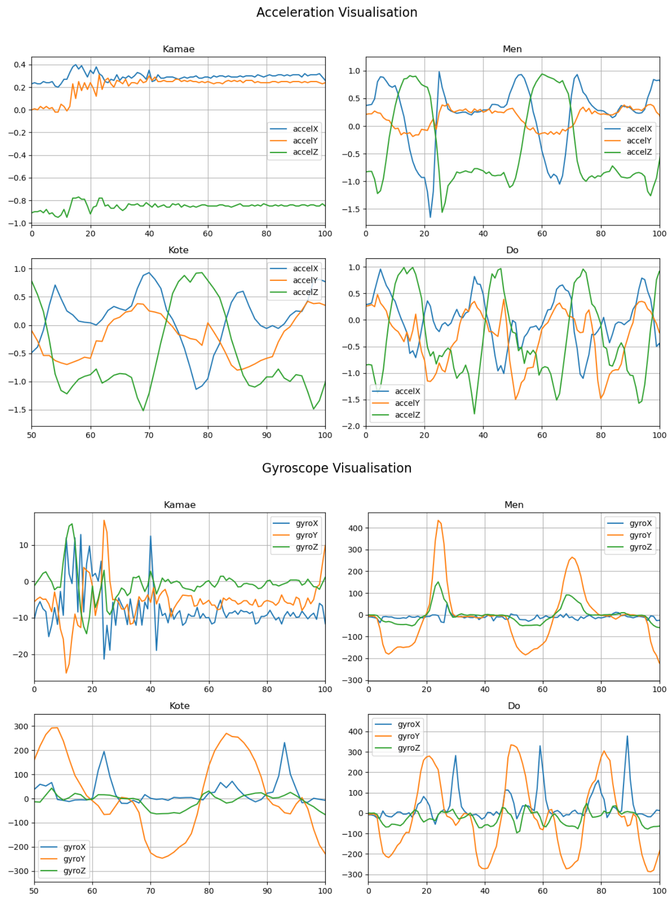


Fig. 5: Time Series Plot of each Kendo Movements (First 100 points)

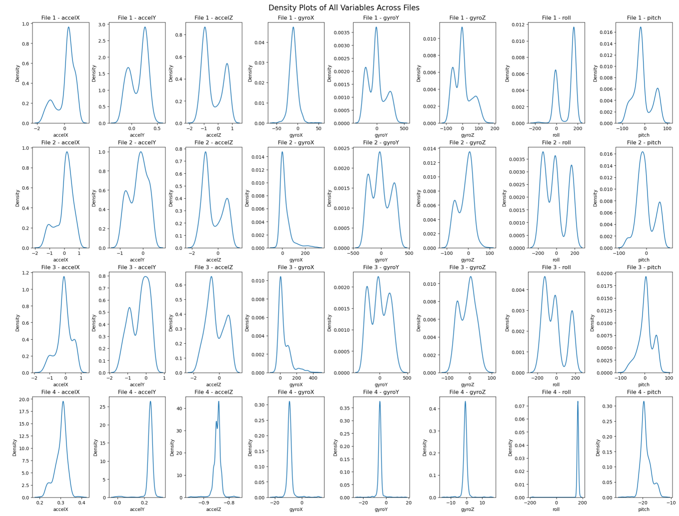


Fig. 6: Dataset Density Plot

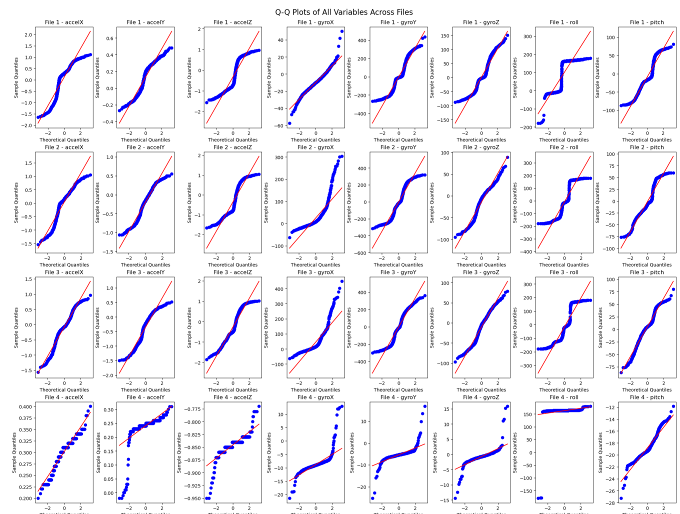


Fig. 7: Dataset Q-Q Plot

As part of the data collection process, four distinct kendo moves were recorded and labelled, with each move performed repeatedly over an approximately two-minute duration. The time-series data, illustrated in Figure 5, reveals unique cyclic patterns for each move, highlighting their distinct characteristics and potential for detailed analysis. For example, the gyroscope readings along the X-axis, which capture rotational speed during a downward strike, are significantly more pronounced in *do* compared to the same motion in *men*, reflecting the variations in execution between these techniques. These cyclic patterns not only provide insights into the subtle differences in kendo moves but also serve as the foundation for two key aspects of the system: the development of a classification model for move identification and the calculation of key performance metrics.

A. Move Classification

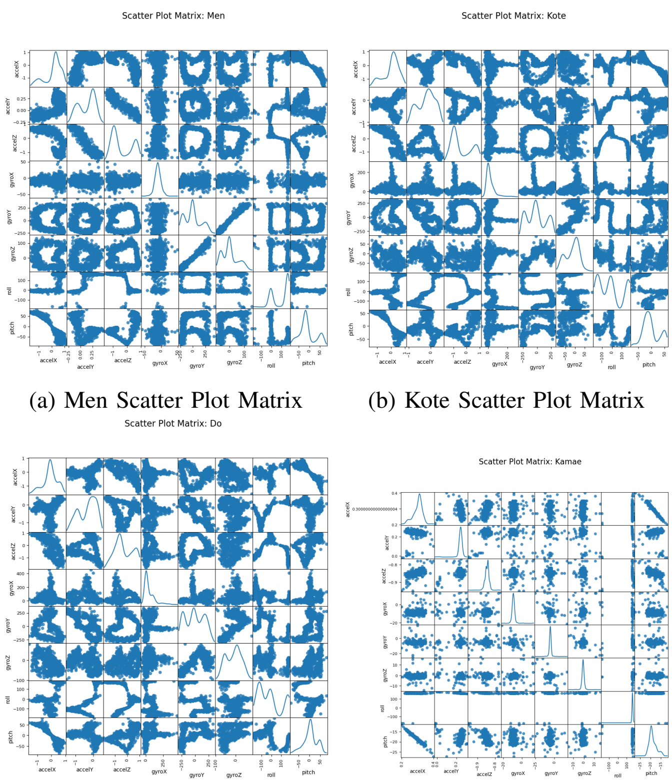
1) Data Distribution Analysis: Having identified that the characteristics of each move is different and can be used for classification, before further analysis are made, the first step is to determine whether the data set is normally distributed as it influences the choice of preprocessing techniques, feature engineering, and machine learning techniques. To build a comprehensive analysis density plot, and Q-Q plots, and histograms (See Appendix C) were made of all variables.

While some plots in Figure 6 and 7 suggest a degree of normality, it is difficult to confirm visually. This aligns within our expectations, as the dataset—comprising basic swing drills—may naturally exhibit a normal distribution due to the consistent and smooth motions involved. However, the presence of heavy tails reflects the biomechanics of reduced control during more forceful swings, introducing deviations from normality. To validate this, the Kolmogorov-Smirnov test, a non-parametric statistical test used to compare the observed distribution of a variable against a reference distribution (in this case, normality), was applied to all variables. The results indicated that none of the variables in the dataset followed a normal distribution.

This immediately limits the choice of machine learning algorithms and statistical methods. For example, we can no longer use Linear Discriminant Analysis (LDA) to characterise swing segments. Although normalising the data might address this assumption, it is not advisable in this case, as the dataset's tails capture critical features of kendo movements.

These tails often represent the peaks and extreme values in accelerometer readings, which are essential for identifying the distinct dynamics of a swing, such as the speed and force at the peak of motion. Normalising the data could suppress these nuances, potentially diminishing the model's ability to effectively classify and analyse kendo techniques.

2) Pairwise Relationship Analysis: The scatter plot matrix is a valuable tool for analysing pairwise relationships, revealing patterns like correlations, non-linear trends, and redundancies. These insights guide feature engineering decisions, such as transforming, combining, or removing variables, and help in selecting modelling techniques suited to the data's structure.



(a) Men Scatter Plot Matrix (b) Kote Scatter Plot Matrix

(c) Do Scatter Plot Matrix (d) Kamae Scatter Plot Matrix

Fig. 8: The Scatter Plot Matrix of all Kendo Moves

From Figure 8, *men* shows dynamic variability with strong non-linear relationships, such as the elliptical shape between $gyroY$ and $gyroZ$, indicating rotational coordination. However, in general the circular pattern indicate smooth movements. *Kote* exhibits tighter distributions in linear acceleration, suggesting controlled motion, while *Do* displays broader variability in rotational features like $gyroX$ and $gyroZ$ reflecting the body rotation. Lastly, *Kamae*, being a static posture, shows weak feature interactions, serving as a base line. Strong correlations, such as $gyroY$ vs. $gyroZ$, and distinct cyclic patterns across movements suggest key features for classification.

Interestingly, the differences in data distribution between *kote* and *do* highlight potential shortcomings in the training dataset. The broader distribution observed in the *do* movements suggests variability in execution, likely reflecting my relative lack of expertise in performing this technique compared to simpler movements like *men* and *kote*. This underscores the importance of gathering training data from more experienced kendoka in future work to ensure consistency and accuracy in motion representation, ultimately improving the model's reliability and generalisability.

3) Feature Extraction: As we are using raw sensor data, a sliding window approach was used to classify the time-dependent kendo movements. This data segmentation method into overlapping window ensures that each window captures the temporal dynamics of the system. Key statistical features—including the mean, standard deviation, maximum, minimum, skewness, and kurtosis—were extracted from each sensor axis for every segment, providing a comprehensive representation of the motion patterns. This structured approach enhances the ability of the classification model to accurately distinguish between different kendo movements (See Appendix 17 for featured engineered histogram).

4) Feature Scaling: With an expanded dataset, data scaling was crucial to address the sensitivity of machine learning algorithms to feature magnitudes. Without proper scaling, features with larger values could dominate the model, overshadowing those with smaller ranges, potentially leading to biased predictions. For this project, two scaling techniques were evaluated: MinMaxScaler and RobustScaler. While StandardScaler was initially considered, it was excluded due to its assumption of data normality, which does not align with the distribution of the sensor data. Ultimately, MinMaxScaler was selected based on empirical experimentation, as it ensures all features are scaled to a specified range, making it particularly effective for algorithms sensitive to feature magnitudes while preserving computational efficiency.

5) Classification Model: To identify and classify different kendo moves, a supervised learning approach was implemented using a Support Vector Classifier (SVC) with a radial basis function (RBF) kernel. Reflecting on the dataset characteristics mentioned in previous sections, SVC was chosen for its robustness to handle non-linear decision boundaries, which is useful in recognising the subtle differences in Kendo moves. As the features exhibit complex relationship and overlapping patterns, RBF kernel is well-suited as it maps the input features into a higher dimensional space, enabling the SVC to better identify a more precise hyperplane that can separate the classes effectively, even when they are not linearly separable in the original feature space.

Moreover, the SVC's ability to generalise well with smaller dataset ensures reliable performance while its relative computational efficiency compared to deep learning model makes it an optimal choice for iterative development and testing. This efficiency allows for scaleability, allowing the model to be deployed on-cloud at a relatively low cost while enable future incorporation of additional training data to improve its classification accuracy.

6) Classification Results Analysis: To evaluate the performance of the classification model, a new unseen test dataset was recorded, encompassing all four kendo moves (*Do*, *Kamae*, *Kote*, and *Men*). The results of the evaluation are summarized in Table IV.

	Precision	Recall	F1-Score	Support
Do	1.00	0.75	0.86	4
Kamae	1.00	1.00	1.00	4
Kote	0.88	0.78	0.82	9
Men	0.78	1.00	0.88	7
Accuracy			0.88	24
Macro Avg	0.91	0.88	0.89	24
Weighted Avg	0.89	0.88	0.87	24

TABLE IV: Test Classification Report

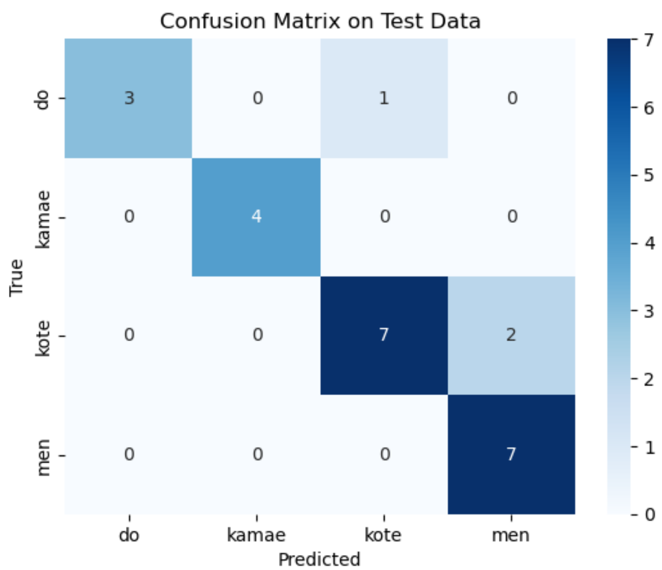


Fig. 9: Test Data's Confusion Matrix

The model demonstrates strong performance across most moves, with particularly high precision and recall for *Kamae*, indicating that its motion characteristics are distinct and easily recognizable. Similarly, the *Men* move achieved perfect recall, meaning all true *Men* instances were correctly identified. However, its slightly lower precision (78%) suggests some misclassifications where other moves were incorrectly predicted as *Men*. For *Do*, the model achieved perfect precision but struggled with recall, highlighting instances where actual *Do* moves were missed. The *Kote* move had the largest support (99 instances) and achieved balanced performance with a weighted F1-score of 82%.

Overall, the classification accuracy of 87.5% demonstrates the model's strong ability to generalize to unseen data. However, creating the test dataset posed significant challenges. Since it included all four kendo moves, every data point had to be meticulously hand-labelled by cross-referencing timestamps with individual frames of the drill recording. This labour-intensive process limited the size of the test dataset (160 data points), making it smaller than ideal and potentially impacting the reliability of performance evaluation. A larger, automated labelling process in future iterations would help paint a better understanding of the model's performance.

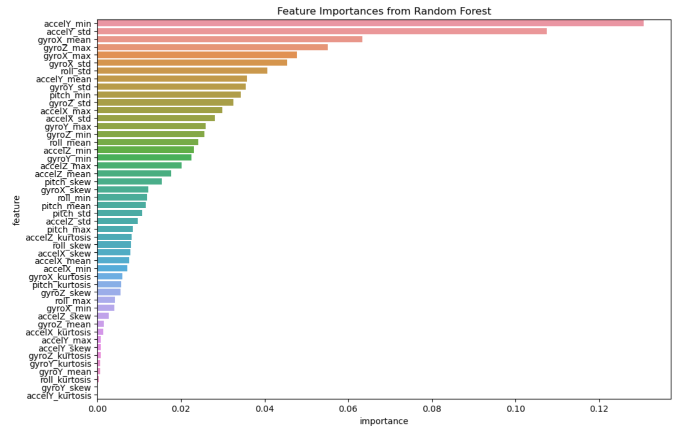


Fig. 10: Feature Importance

Random Forest Classifier was then applied to rank the feature importances, to better understand which features contribute the most to differentiating kendo moves. As shown in Figure 10, the most influential features are primarily related to the Y-axis acceleration ($accelY_{min}$, $accelY_{min}$) and the standard deviation and maximum values of the gyroscope readings ($gyroX_{mean}$, $gyroZ_{max}$). These results align with the dynamics of kendo moves as the Y-axis acceleration reflects the downward force and the variability of motion. For example, a sharper drop in Y-axis acceleration is indicative of the more linear trajectory of *men* strikes, while *do* exhibits more gradual acceleration patterns due to its diagonal motion.

For gyroscopic features, The prominence of $gyroX_{mean}$ and $gyroZ_{max}$ highlights the rotational dynamics involved in kendo moves. For instance, the rotational speed around the X-axis is more pronounced during wrist-driven movements like *kote*, while the Z-axis rotation becomes significant in diagonal strikes like *do*. These findings demonstrate how the classifier can effectively captures the unique characteristics of kendo strikes, thus validating our feature engineering decisions and the implementation of SVC.

Typically, the next step is to identify and remove features with low importance for dimensionality reduction. This step is important in reducing computational overhead and reduces the risk of over-fitting. However, from testing with an unseen mixed dataset (a dataset containing all types of moves), the accuracy of 87.5% is satisfactory and require no further refinement.

B. Key Performance Metrics

To improve the user's form, three key performance metrics were identified and analysed. The first metric is the acceleration of the sword, which reflects the intensity of the strike and provides insights into the power behind the swing. The resultant acceleration is calculated as:

$$a_{resultant} = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{1}$$

The second metric is jerk, which evaluates the abruptness of motion and infers the smoothness of the user's movements. High jerk values indicate a lack of control or unnecessary force transitions, and it is calculated as:

$$J = \frac{\Delta a}{\Delta t} \quad (2)$$

Finally, the most critical performance metric is the motion smoothness, which indicates refined technique and control. Smoothness in this case is quantified as the percentage of time points with "smooth" movements, defined as those where jerk value remained below a specified threshold. This metric provides an intuitive measure of the user's motion fluidity and control.

$$Smoothness = \frac{\sum Smooth\ \ Durations}{Total\ \ Time} \times 100 \quad (3)$$

C. Cloud Infrastructure & User Interactions

Having derived the key insights from the sensors, the next step is to convey said information effectively to the user.

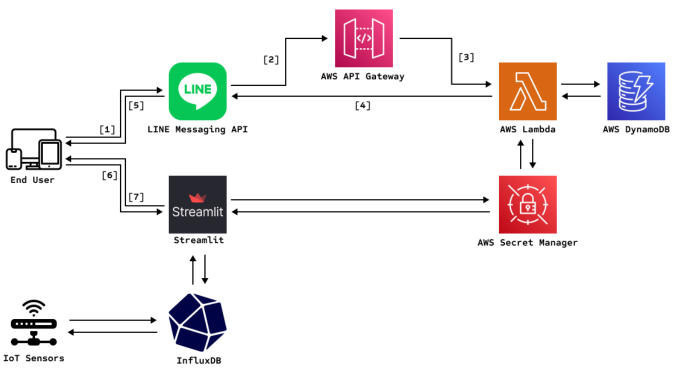


Fig. 11: Cloud Solution Architecture

The process in which the user can access are as follow:

[1] User sends a message to a KiAI channel on LINE messaging app.
[2] The message is received and checked via Amazon API Gateway.
[3] The edge-optimised Amazon API Gateway passes valid requests via a proxy integration to AWS Lambda. At this stage, AWS Lambda compares the request body with the 'x-line-signature' request header to confirm that the request was sent from the LINE Platform, checks the user's data in the AWS Dynamo Database, and listens for keywords for commands (e.g. 'practice' results in returning user's dashboard URL).
[4] AWS Lambda then returns the URL to KiAI practice dashboard in the correct format expected by the LINE Messaging API.
[5] LINE Messaging API receives the request and forwards the URL to the KiAI practice dashboard back to the user.
[6] Through the link, the user can access the KiAI dashboard hosted online using Streamlit.
[7] Real-time data and key performance metrics are displayed to the user on the online dashboard.

1) Feature Selection Rationale: Before discussing the specifics of the dashboard, it is important to first highlight the rationale for integrating the LINE API with an AWS back-end into the project.

Firstly, this integration addresses the interoperability and scaleability requirements. As the number of users grows, the service would require a centralised platform for efficient user management. The use to LINE API eliminates the need for a traditional login page, leveraging LINE account IDs as a unique identifiers to streamline the user experience by enabling direct access to their personal dashboard. Moreover, this approach simplifies the back-end logic, as LINE IDs can directly map to corresponding URLs or data entries in the DynamoDB and InfluxDB, ensuring efficient data retrieval and management. It also provides a foundation for future developments, enabling the system to deliver insights and notifications directly to users via LINE messages, reducing reliance on dashboards and further enhancing usability. Additionally, LINE have a *richmenu* feature which creates a selection menu for the users on the app, allowing the user to communicate with the bot without needing to type. While this feature was not entirely implemented in this project due to the limited timeframe, the capability is there and is relatively easy to scale for greater user experience.

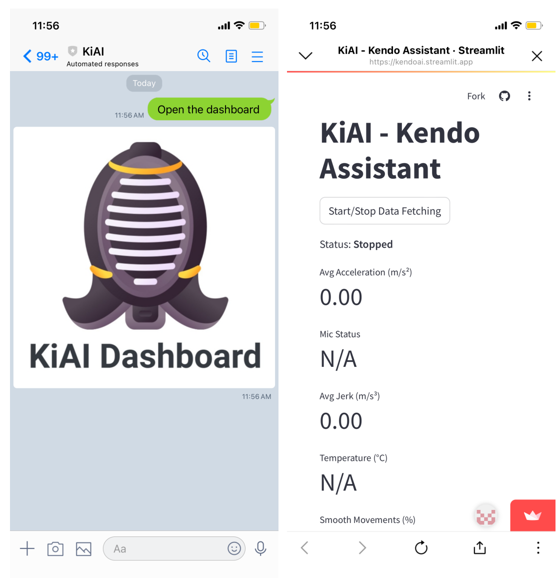


Fig. 12: User Interface on LINE (IOS)

Secondly, the inclusion of AWS Secrets Manager significantly enhances security across all processes. While personal kendo data may not be highly sensitive, tokens and credentials required for data communication must be securely stored and dynamically retrieved at runtime. This approach mitigates the risks of unauthorized access or hard-coded vulnerabilities in the scripts.

Finally, the server-less architecture ensures cost efficiency and flexibility. Costs are incurred only when requests are processed, making this design highly economical. Moreover, its modularity facilitates future enhancements, such as integrating additional APIs or third-party services, without requiring substantial changes to the existing infrastructure.

2) AWS Design Considerations: In addition to feature selection, several key considerations guided the design and implementation of the back-end infrastructure. One of the most

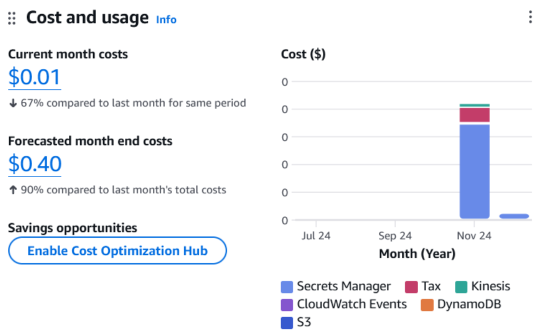


Fig. 13: Cost Breakdown of AWS Services in November

critical aspects was the optimisation of AWS and InfluxDB regions. As the system operates in real time, it was essential to balance low latency, cost efficiency, and service availability.

For most AWS services, including Lambda, API Gateway, DynamoDB, and Secrets Manager, the London (eu-west-2) region was selected to minimise latency, as the primary user base is located in the UK and nearby regions. Hosting these services closer to the users ensures faster response times and enhances the overall user experience.

In contrast, InfluxDB is hosted in the US East (us-east-1) region due to limited cloud availability in closer regions. As InfluxDB does not currently provide services in the UK, the US East region is both geographically close and the most well-supported. While this choice introduces slight network latency for data transmission, the trade-off is justified by the robust feature set and cost efficiency offered in the US East region, making it ideal for handling high-frequency time-series data.

As shown in Figure 14, the KiAI dashboard displays the key information and analyses discuss in previous sections, being, average acceleration, average jerk, movement smoothness, move classification, and environmental data. As these are the most important data to the users in understanding their swings, it is strategically placed at the top to ensure that the most relevant information is immediately accessible and easy to read. The column layout further prioritises information by aligning metrics in descending order of importance, guiding the user's focus intuitively from left to right. Given the high-frequency, real-time nature of data writing and reading, a start-stop button was implemented as a cost-saving measure. Since AWS and InfluxDB services operate on a demand-based pricing model, this feature ensures the system minimises additional costs when not actively in use, making the solution both economical and user-friendly.

Beneath the key metrics section, time-series plots of accelerometer and gyroscope readings provide detailed insights into motion data. These plots allow user to see changes in bokuto orientation and movement dynamics, helping them refine their technique. Of course, the plot is interactive and the users are free to zoom-in and out for further analysis. Additionally the dashboard includes a live video stream of the practice session, enabling real-time monitoring of form. These integration of motion data and visual feedback offers a more comprehensive approach to improving kendo performance. All of the data displayed are stored in the session state. This means that even after the user press the stop button, the numbers and plots would remain the same, allowing the users to freely explore and analyse their training session.

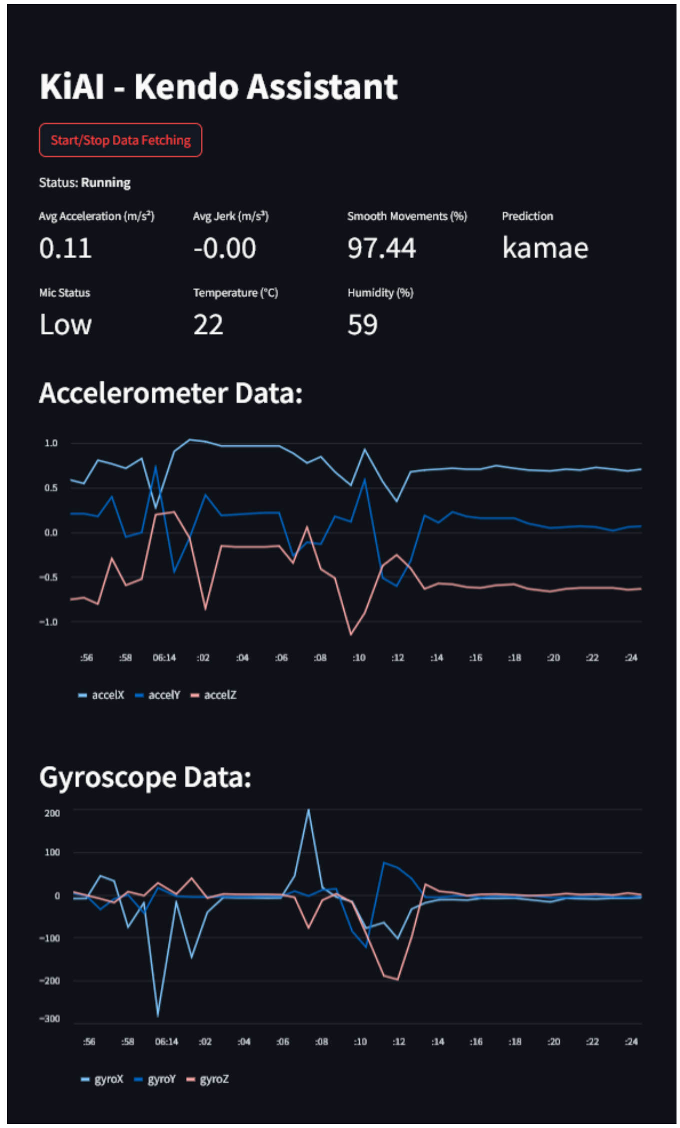


Fig. 14: KiAI Dashboard (Firefox)

One notable advantage of using Streamlit to host the dashboard, in addition to its deployment simplicity, is its built-in screen recording capability. This eliminates the need to develop additional JavaScript functionalities for recording practice sessions, significantly reducing development complexity and time while offering users a better experience.

IV. Project Development Discussion

Overall, this project was able to achieve all of the design objectives set in the beginning, where the system successfully used the data from the IoT devices to give meaningful insights to the users. This comes in the form of the move classification model, derivation and display for key performance metrics, environmental data insights, and the video stream.

A. Actuation?

Initially, the project aimed to incorporate physical actuation as an additional feature to provide users with real-time feedback on their movements, eliminating the need to constantly check the dashboard. One proposed implementation was a wrist-mounted display that would indicate whether the user's motion aligned with the ideal trajectory for a specific kendo move. However, this concept conflicted with the core philosophy of kendo, which emphasises maintaining focus and concentration on oneself in the present moment. Such a form of interaction would be counterproductive, as it could distract practitioners from the essence of their practice. Therefore, the final design prioritised a dashboard that could be reviewed after practice or monitored by a second person, aligning the *KiAI* system better with kendo's principles.

B. Security Concerns

While several measures were implemented to enhance the system's security—most notably the use of AWS Secrets Manager and adherence to the "Principle of Least Privilege"—there remain areas requiring improvement. One significant issue is that certain sensitive tokens are still hardcoded within the ESP32 and Arduino scripts.

Although it is technically feasible to integrate these microcontrollers with AWS Secrets Manager by connecting to AWS IoT Core and invoking Lambda functions to retrieve secrets securely, this approach was not pursued due to the time constraints of the project (MQTT protocol needed, as discussed in the later subsection). This issue must be addressed in future developments by implementing secure methods for storing and retrieving credentials, such as utilizing AWS Secrets Manager or alternative secure storage solutions.

Another potential security risk is the livestreaming of the camera to the local server. Not only does the lack of authentication allows anyone on the network to access the video feed, but it also uses unsecured HTTP making the stream extremely vulnerable to eavesdropping and *man-in-the-middle* attacks. While this risk was identified during the development process, there were many unidentified sources of error in the network settings that prevented the encryption of the video feed from ESP32 (This is due to the lack of technical/programming knowledge on my part).

C. Client-Server Protocols

In this project, the system uses HTTP rather than MQTT, a common choice for IoT devices. The primary reason for this decision is that InfluxDB natively supports HTTP-based writes and queries, eliminating the need for an additional MQTT broker. This simplifies the system architecture, reduces development complexity, and minimises potential points of failure. Direct communication via HTTP also reduces latency by avoiding intermediary layers, ensuring efficient data transmission.

Additionally, the system operates at a maximum data upload rate of 10Hz, which is relatively low frequency compared to real-time IoT applications. As such, there is no immediate need for the lightweight, continuous connection provided by MQTT. More importantly, the current setup involves multiple microcontrollers sending data to distinct InfluxDB buckets, focusing solely on one-way data transmission without requiring bidirectional communication. Given these requirements, HTTP provides sufficient performance, reliability, and simplicity for this application.

D. Future Developments and Improvements

While the proposed *KiAI* system works well by having an IoT device stuck to the sword, this would only be viable during practices when the actual International Kendo Federation rules doesn't apply. During competition, it is critical that the equipment used are not modified in anyway or contain any electronics. This is one of the reason that Kendo does not use sensors seen in fencing. This means, that insights regarding the users techniques during competition would need to be from image recognition. An interesting avenue to continue development.

The second area for improvement is training data collection and labelling, which can be divided into two key aspects: capturing more techniques and distinguishing between good and bad movements.

First, to gain deeper insights into the user's technique, it is essential to collect data that goes beyond simply identifying the type of attack being performed. A larger, more comprehensive dataset, including labelled instances of various *kata* (fixed patterns designed to teach the fundamental elements of kendo), would enable the classification model to better understand the nuances of the user's actions. For instance, in competition, a commonly used technique is *harai waza*, where the practitioner swiftly sweeps the opponent's weapon aside with a quick wrist twist before delivering a strike [6]. This movement is both rapid and intricate, requiring precise timing and control, making it difficult to identify without sufficient training data. By building a robust dataset, the model could reliably detect and classify such advanced techniques, enhancing its usefulness for users seeking to improve their skills.

The second aspect is the ability to classify movements as correct or incorrect. While the model might recognize the type of move being performed, without sufficient training data, it risks reinforcing poor techniques. For example, an improper wrist motion during *harai waza* could result in ineffective strikes or unnecessary openings for the opponent. To mitigate this, it is imperative to involve kendo experts in the data labelling process. Their expertise ensures that the training dataset captures the correct execution of techniques, enabling the model to discern proper form and provide meaningful feedback on whether a user's movements align with kendo principles.

This process is particularly challenging for novices (like me), as accurately labelling such data requires detailed knowledge of the phases of an attack, such as normal stance, attack initiation, seizing an opportunity, striking, and follow-through [7]. Expert assistance is crucial to achieve this level of granularity, ensuring that the dataset accurately reflects the complexities of kendo techniques. A well-curated dataset will not only improve the model's classification accuracy but also help users develop correct habits and refine their overall technique.

The third area for improvement is the integration of more sensors to improve the classification model. Existing researches found correlations between the grip strength and the moves being done [8], meaning further improvements could include pressure sensors on the sword grip itself for more data-points.

E. Project Management

Throughout the project, I kept track of the essential tasks through the use of Gantt Chart (Appendix B), which provided a clear roadmap for major milestones and deadlines. While this approach worked well in keeping the major tasks on-time, there were some deviations made to the plan. For instance, the initial design thought of in week 1 included the integration of piezoelectric pressure seonsors onto the kendo armour to register strike events. However, due to the armour being owned by the Imperial Kendo Club and being loaned to me; any modifications - even those unlikely to cause damage - would be inappropriate. While this realisation came relatively early in the developmental process, the components were already ordered. This resulted in this project costing more than initially planned as well as pushed the timeline back as I had to made (small) changes to the project. Although this adjustment extended the hardware procurement timeline, my mitigating strategy was to re-prioritise tasks and use that timeframe to work on the LINE messaging feature and the AWS back-end as to not wait time. This experience underscored the importance of flexibility in project planning and the need for contingency measures to address unforeseen challenges effectively.

V. Personal Reflection

This project provided an invaluable opportunity for personal growth, challenging me to step out of my comfort zone and tackle unfamiliar areas of system design. With limited prior experience in back-end development, I had to quickly adapt and take the initiative to learn new skills, particularly in integrating different AWS services and handling time-series data. The iterative nature of the project not only deepened my technical understanding but also reinforced the importance of problem-solving and persistence in addressing complex challenges. A major point of improvement I wish to further develop is in programming. While I was able to achieve most aspects of the project by myself, there are still many parts that I failed to implement, even with help from generative AI. An example of this is the use of static credentials on the Arduino scripts. Regardless, through this project, I have gained greater confidence in my ability to design and implement robust IoT systems for problems I find meaningful.

VI. Conclusion

To conclude, the KiAI system successfully demonstrates the potential of IoT and machine learning technologies in enhancing kendo training by providing real-time insights into user performance. By leveraging a compact and efficient sensing architecture, the system captures key motion metrics such as acceleration, jerk, and smoothness, enabling detailed analysis of kendo techniques. The integration of machine learning, specifically an SVC model with RBF kernel, facilitates accurate classification of moves with an overall accuracy of 87.5%. This performance validates the robustness of the chosen features and model design.

Despite these achievements, the project also highlights areas for further improvement. The current reliance on manual data labelling and the limitations of a single-sensor setup point to opportunities for refining the dataset and expanding the system's capabilities. Future developments could focus on incorporating image recognition for competition scenarios, automating data labelling with expert collaboration, and expanding classification to include technique quality.

KiAI not only bridges the gap between traditional martial arts training and modern technology but also respects the core philosophy of kendo by emphasizing focus and post-practice analysis. This balance ensures that the system serves as a valuable tool for practitioners, paving the way for further innovations in digital sports training.

VII. Acknowledgements

I would like to extend my sincere gratitude and thanks to Dr. David Boyle for his lectures and guidance throughout this module and project. Additionally I would like to also thank all of the graduate teaching assistants from the Systems & Algorithms Lab at Imperial College London for their advice during the tutorial sessions.

VIII. AI Usage Declaration

I would like to formally acknowledge the use of generative AI tools, specifically GitHub Copilot, in this project. The use of generative AI was limited to debugging and optimisation of back-end code (AWS services related) integration. At no point in the development process were any sensitive credentials been shared with large language model. This includes all scripts used to create the classification model as well as the Arduino scripts.

References

[1] A. Bennett, *Kendo: Culture of the Sword*. University of California Press, 2015. [Online]. Available: https://books.google.co.uk/books?id=RaXOCQAAQBAJ

[2] U. of Tsukuba. (2017, February) The establishment of kendo techniques. Accessed: November 29, 2024. [Online]. Available: https://budo-world.taiiku.tsukuba.ac.jp/en/2017/02/06/

[3] A. Electronics, "Dht11 humidity and temperature sensor technical data sheet," 2020, accessed: November 20, 2024. [Online]. Available: https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf

[4] K. Gokui. (n.d.) The rules of kendo. Accessed: November 30, 2024. [Online]. Available: https://www.kendo-gokui.com/

[5] K. Jogja. (2008) Fundamentals of kendo ii: Kamae. Accessed: November 30, 2024. [Online]. Available: http://www.kendojogja.com/2008/10/fundamentals-of-kendo-ii-kamae.html

[6] A. J. K. Federation, *Nippon Kendo Kata*. Tokyo, Japan: All Japan Kendo Federation, 1977, instruction manual on the fundamental forms of Kendo practice.

[7] ——, *The Official Guide for Kendo Instruction*. Tokyo, Japan: All Japan Kendo Federation, 2008, comprehensive guide on Kendo techniques and principles.

[8] K. Jeong, A. M. Tan, T. Asai, K. Koda, and F. K. Fuss, "Pressure sensors for measuring the grip pressure during kendo attacks: Assessment of laterality and evidence of the five phases of attack," *Sensors (Basel, Switzerland)*, vol. 23, no. 3, p. 1189, 2023. [Online]. Available: https://doi.org/10.3390/s23031189

APPENDIX

A. Complexity Analysis of MPU-6500 Data Collection

State Machine Representation	Big O Analysis
Initialisation (Setup Phase)	Wi-Fi Connection: $O(n)$, Sensor Initialisation: $O(1)$
Main Loop	Sensor Updates: $O(1)$, Calculations: $O(1)$, HTTP Data Transmission: $O(n)$
Overall Complexity	$O(n)$

TABLE V: Complexity Analysis of MPU-6500 Data Collection

B. Project Gantt Chart

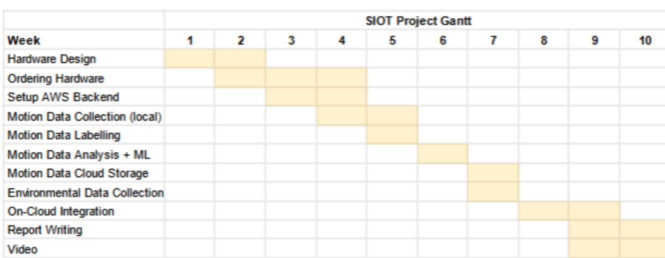


Fig. 15: Project Gantt Chart

C. Histogram of Kendo Move Variables

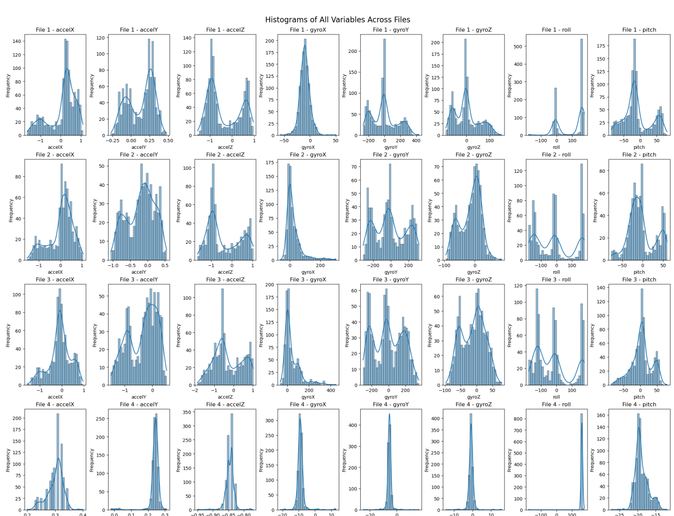


Fig. 16: Histogram of Test-Train Dataset

D. Histogram of Variables after Feature Engineering

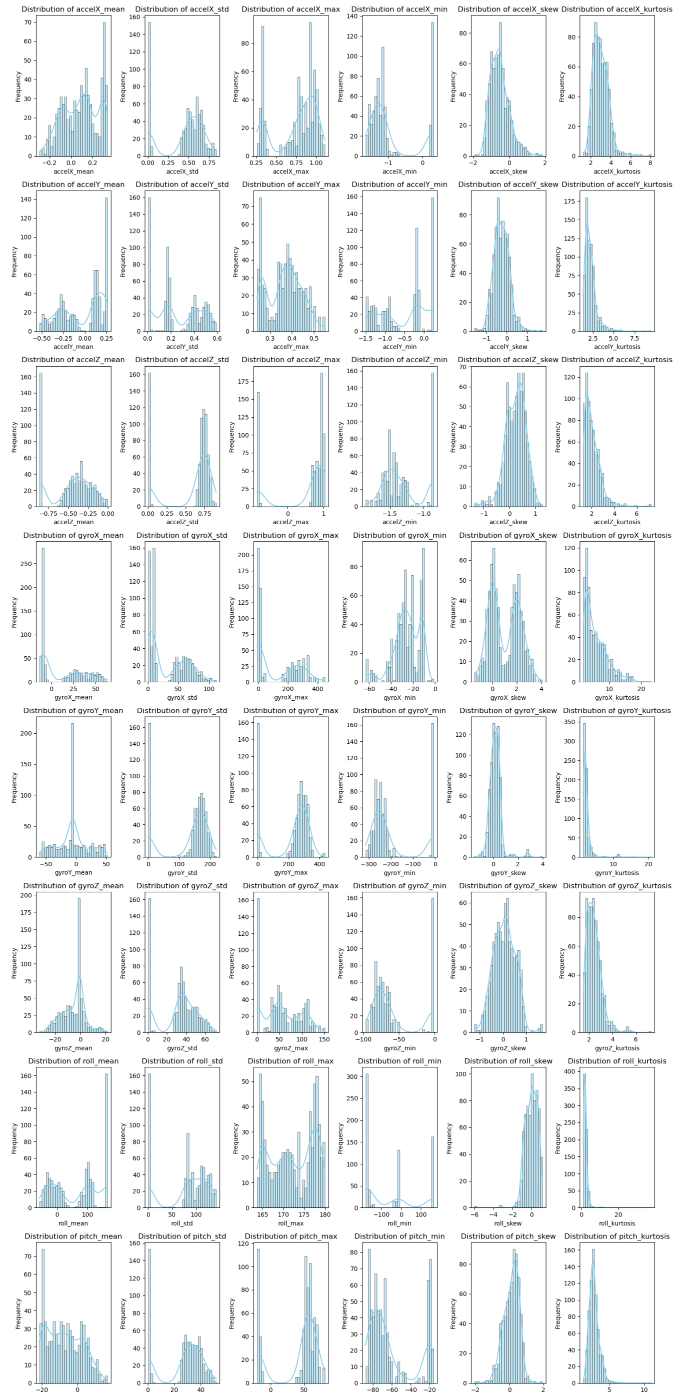


Fig. 17: Histogram of all variables after feature engineering

E. Sensor Setup on the Sword

Fig. 18: Sensor Setup on Bokuto

F. Sword Sensor Setup Closeup

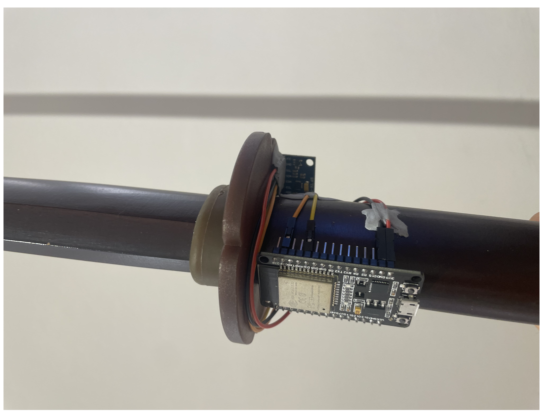


Fig. 19: Motion Sensor Closeup

G. Environmental Sensor Setup

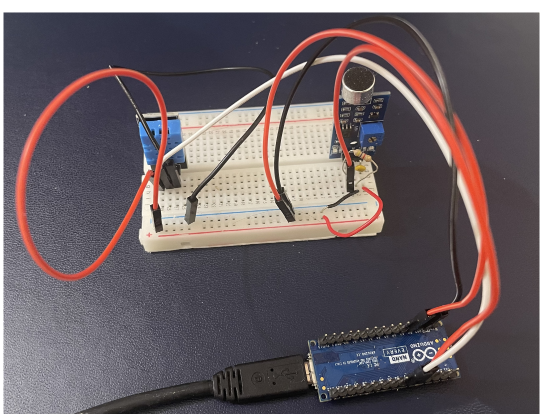


Fig. 20: Environmental Sensor Setup

H. ESP32-CAM

Fig. 21: ESP32-CAM and Camera