School of Electronic and Electrical Engineering
FACULTY OF ENGINEERING

UNIVERSITY OF LEEDS

ELEC5882M MSc Individual Project

Declaration of Academic Integrity

Plagiarism in University Assessments and the Presentation of Fraudulent or Fabricated Coursework

Plagiarism is defined as presenting someone else's work as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance.

Fraudulent or fabricated coursework is defined as work, particularly reports of laboratory or practical work that is untrue and/or made up, submitted to satisfy the requirements of a University assessment, in whole or in part.

Declaration:

- I have read the University Regulations on Plagiarism [1] and state that the work covered by this declaration is my own and does not contain any unacknowledged work from other sources.
- I confirm my consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes.
- I confirm that details of any mitigating circumstances or other matters which might have affected my performance and which I wish to bring to the attention of the examiners, have been submitted to the Student Support Office.
- I confirm that any use of generative AI in the preparation of this report has been properly referenced, as per the instructions.

[1] Available on the School Student Intranet

Student name: AARUSH ARYA

Signed: Aarush Arya

Date: 08/03/2026

Smart Table Tennis Training System: IoT-Based Movement Analytics with Edge Machine Learning

Aarush Arya (202022653), *bdgw0689@leeds.ac.uk*, Dr Craig Evans (Supervisor)

***Abstract*—Objective feedback on table tennis stroke technique conventionally requires a dedicated coach or laboratory instrumentation, neither of which scales to routine practice. This report presents a fully untethered smart racket that recognises strokes on the racket itself. A six-axis inertial measurement unit (Appendix N) embedded in a 3D-printed handle is sampled at 100 Hz, with its accelerometer range extended from ±4 g to ±16 g through a driver-level modification verified against gravity. One-second windows are reduced to 78 log-spectral features and classified by an INT8-quantised (Appendix A) dense neural network (Appendix B) distinguishing four stroke classes and a still-idle class. Training and evaluation use two independent recording sessions with the hardware re-mounted between them, a protocol adopted after data leakage was diagnosed in an earlier single-session result. On the fully held-out session the deployed model achieves 90.91% accuracy, identical in float32 and INT8, with per-class accuracies of 88.6–97.7% and weighted F1 (Appendix C) of 0.93; feature extraction completes in 20 ms within 3 kB of RAM on a Cortex-M4F. Deployment exposed and resolved two measured acquisition defects, and the final on-racket detector classifies once per completed swing with the stroke peak centred in the window, reporting a verdict ≈0.5 s after each stroke. The results demonstrate that real-time, on-device stroke feedback is feasible on sub-£80 hardware.**

***Index Terms*—Bluetooth Low Energy, edge machine learning, embedded systems, human activity recognition, inertial measurement unit, table tennis, TinyML (Appendix O).**

I. Introduction

TABLE tennis outcomes are decided by small differences in stroke technique, contact point, and spin.Coaching feedback is inherently subjective and difficult to scale, as one coach can observe only one player at a time, while objective measurement of racket motion typically relies on high-speed cameras or laboratory equipment that are impractical for routine training. A low-cost racket that senses its own motion and reports objective stroke statistics during ordinary play would give players and coaches automated insight without changing how the sport is practised.

Inertial sensing has repeatedly been shown to support this goal. Blank *et al.* classified eight stroke types at 96.7% accuracy using a racket-mounted IMU with hand-crafted features and an SVM (Appendix D) [1], and deep models have raised recognition performance further: Ferreira *et al.* report a 97.33% F1 score with a ConvLSTM (Appendix E) on wrist-worn data [2], and IMU pipelines have been extended to fine-grained forehand evaluation [3] and cross-sport racquet activity recognition [4]. Two findings from this literature are load-bearing for the present work: strokes played on the same wing are reported as the dominant confusion [2], and a wrist-worn device cannot observe racket-specific information such as blade contact location, which motivates racket-mounted sensing.

Contact location and spin have instead been addressed with blade-level piezoelectric sensing (Appendix F). Row–column electrode arrays reduce an $m\times n$ contact grid to $m+n$ channels and can infer rotation type [5], and ultrathin piezoelectric arrays laminated between blade and rubber discriminate stroke classes from strain distributions [6], [7]. For the present project, the key point is that these approaches rely on a flat strain-mode film spanning the monitored area, and this transduction requirement ultimately motivated the de-scoping decision (Section V).

Prior systems, however, are predominantly host-bound: classification runs offline or on a paired computer, cloud service, or smartwatch [2], [8], and reported accuracies are frequently not evaluated on session-independent data. The gap addressed here is a racket that is simultaneously untethered, performs inference on the embedded device itself, and is evaluated against a genuinely independent recording session. The contributions are: (i) a racket-integrated wireless sensing prototype with a verified power and communications chain; (ii) an on-device, quantised stroke classifier wrapped in a swing-segmented, peak-centred real-time detection pipeline; and (iii) an evaluation and deployment methodology — independent two-session testing, per-window timing instrumentation, and bit-exact off-target replay of the deployed artifact, adopted after diagnosing data leakage in an earlier 100% result. The remainder of the report covers hardware design (Section II), firmware, dataset and deployment (Section III), evaluation (Section IV), and conclusions (Section V).

A. Aims and Objectives

1) Project Aim: The aim, as set out in the scoping document, is to design and evaluate a smart table tennis racket that recognises common stroke types in real time on low-cost hardware and provides immediate, objective feedback usable during normal practice, with blade-level sensing of contact position and spin investigated as an extension of the same platform.

2) Project Objectives:

1) Review the literature on racket and wearable sensing and stroke recognition, identifying the gap in integrated, low-latency, on-device systems.
2) Design and build a racket-integrated prototype combining motion sensing with blade impact sensing, suitable for dataset collection under realistic play.
3) Develop a machine-learning model that recognises stroke types (with hit position and spin as extensions)

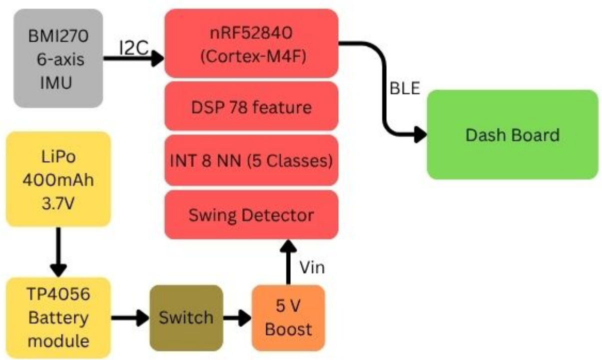

Fig. 1. Hardware block diagram of the smart racket, showing power and data paths separately.

accurately and efficiently enough for real-time use on low-cost embedded hardware.

4) Evaluate the system under realistic playing conditions with an honest, session-independent protocol, documenting limitations.
5) Produce and document all deliverables, including this report, to the required academic standard.

These objectives are revisited individually in Section V-A.

II. Hardware Design

A. System Architecture and Design Constraints

Fig. 1 shows the hardware architecture: a single sensing–compute–radio board powered by a battery rail, all sealed inside a 3D-printed handle. Three constraints drove every component decision. First, *size*: everything must fit a 37×10×100 mm printed handle cavity, so board and battery geometry are hard limits. Second, *mass and rigidity*: added mass alters swing dynamics and a loosely mounted sensor corrupts the inertial data it exists to collect. Third, *untethered operation*: a cable restricts the swing and therefore distorts the very motion being measured, making battery power and a wireless link requirements rather than conveniences. The complete prototype cost is approximately £78 excluding VAT (board £30.99, power subsystem ≈£15, blade and rubber ≈£32, handle printed in-house), grounding the low-cost claim of this report.

B. Microcontroller Selection

The Arduino Nano 33 BLE Sense Rev2 was selected because it integrates the entire sensing–compute–radio chain on one 45×18 mm board: a Nordic nRF52840 SoC (Arm Cortex-M4F at 64 MHz with hardware floating point, 1 MB flash, 256 kB SRAM), an on-die 2.4 GHz radio, and an on-board BMI270 IMU [10]. Flash holds the firmware and model weights. SRAM is the run-time ceiling for tensors, met with large margin (Section IV-C). Alternatives (ESP32, classic Nano, single-board computers) each fail at least one of the three constraints on integration, power or size.

C. Inertial Sensing and Full-Scale Range

The BMI270 [9] is a six-axis MEMS device combining a three-axis accelerometer and three-axis gyroscope with 16-bit signed output per axis, so the configured full-scale range (FSR, Appendix G) defines the scale in counts per g. Sampling was set to 100 Hz, twice the 50 Hz rate used in comparable wrist-worn work [2], providing headroom for the transient content of fast strokes.

During bring-up the accelerometer appeared to saturate: values railed at exactly ±4 g on hard swings, the default FSR, and the vendor Arduino driver hard-locks this range with no exposed API. The fix required two *coupled* edits in the driver source: the range register (4G→16G) and the conversion constant (8192→2048, since $32768/16 = 2048$ counts per g). Changing one without the other scales every reading by a factor of four. The edit was verified physically: a resting axis reads ≈1.0 g. One reproducibility caveat is that this is a local library modification, silently reverted by any library update.

In hindsight, handle-mounted stroke peaks reached only 2.8–5.9 g (single-axis 5.4 g), so ±8 g would also have sufficed; ±16 g retains headroom. Peak angular rate (Appendix J) was 966 °/s (a backhand topspin), within the gyroscope range.

D. Power Subsystem

The power chain is LiPo cell → TP4056 charger → SPDT switch → 5 V boost → MCU V_{IN}. The cell is a 522332 flat pouch (400 mAh, 3.7 V nominal, 5×23×32 mm) selected by geometry: the handle battery bay measured from the STL is approximately 30×22×7 mm, a space into which a cylindrical 18650 cell (ordered earlier) physically cannot fit therefore, form factor, not capacity, was the deciding specification. The TP4056 provides the constant-current/constant-voltage termination profile lithium cells require, plus DW01-based over-charge, over-discharge (≈2.5 V cut-off) and short-circuit protection between cell and load. The boost stage is required: the cell delivers 3.0–4.2 V across its discharge curve against a V_{IN} regulator expecting ≈5 V; a 150 kHz step-up (≈85% efficient, 480 mA against a measured draw below 100 mA) holds the rail as the battery sags. The SPDT switch sits between charger output and boost input, so the sealed handle powers down without opening while the charge path stays live.

One honest limitation is charge rate: the TP4056 default of 1 A is ≈2.5C for a 400 mAh cell, which is aggressive. Charging is therefore supervised, and the programming resistor can be changed to reduce the current to ≈0.5C. Bring-up followed a verify-then-connect discipline: the full chain was built and the boost output measured at 5.0 V with a multimeter before the MCU was ever attached.

E. Mechanical Integration

The electronics are housed in a two-part 3D-printed handle replacing the original, with the two conductors crossing the split line strain-relieved. Critically, the mount defines a *single fixed IMU orientation* held constant across all data collection. The tempting alternative — flipping the board between forehand and backhand recording recording was rejected because it flips the gravity vector and would let the classifier separate strokes by orientation rather than motion, creating a data leak designed out at the hardware stage (Section III-B).

III. Firmware, Dataset and Deployment

A. Acquisition Firmware and Wireless Link

Two firmware builds were produced: an acquisition build that streams the six-axis IMU for dataset capture, and an inference build (Section III-D). Acquisition migrated from USB serial to BLE because wireless operation is a data-validity requirement: a tethered racket cannot be swung representatively. A fixed 26-byte little-endian notification carries [uint16 sequence counter | 6×float32], sent per sample at 100 Hz (≈2.6 kB/s) with a 7.5–15 ms connection interval. The sequence counter makes link integrity *measurable*: the host reconstructs timestamps from the device's own counter, so a lost packet appears as a genuine time gap rather than silently compressed time, and every recording reports its achieved rate and drop count. Measured across all 41 recordings of the final dataset, the link delivered 47,358 of 47,804 expected samples (0.93% loss, effective 99.1 Hz, median inter-sample interval exactly 10 ms, worst gap 50 ms). This measure-rather-than-assume stance had already paid for itself: an earlier silent regression to ≈65 Hz (a side-effect of the slower ±16 g driver path) was caught by per-recording rate checks after collapsing a first real-ball test to 57.8%.

B. Dataset and Experimental Protocol

The dataset comprises four stroke classes (forehand/backhand × drive/topspin) recorded with real ball contact, plus two idle classes (still, moving) capturing non-stroke racket handling. Composition is given in Table I. Recordings were made in two independent sessions, S1 (training) and S2 (held-out test), with the board physically re-mounted between sessions. This protocol exists because an earlier single-session split produced 100% accuracy through data leakage, training and test windows from the same continuous swings and treating the perfect score as a warning rather than a result is itself a finding of this project. Class recording order was interleaved so fatigue does not correlate with the class label. One stated limitation is that inter-stroke dead time within real-ball recordings introduces some label noise, since windows between contacts still carry the stroke label. The idle class partially addresses this by giving such motion a class of its own. Gate-detected stroke events (peak gyroscope magnitude criterion, Section III-D) indicate ≈434 strokes in S1 and ≈165 in S2 at a cadence of ≈1.8 strokes/s.

C. Feature Extraction and Model

The classification pipeline was produced with the Edge Impulse toolchain [11]; the tool choice is justified by its direct support for the target board and INT8 deployment path, and the description here is limited to the design decisions. Six-axis windows of 1000 ms with a 500 ms increase at 100 Hz are reduced by spectral analysis (Appendix I) (FFT length 16, log spectrum, overlapping frames, no pre-filter) to 78 features. Frequency-domain features were chosen over raw time series because strokes are transient events whose discriminative content is spectral shape rather than sample-level alignment, and because the resulting 78-dimensional input keeps the MCU-resident model small. The final classifier is a compact dense network on the 78 features with five outputs (four strokes and still-idle), post-training quantised to INT8. An intermediate six-class variant, with identical DSP and dense 20–10 hidden units trained for 100 cycles at a learning rate of 0.005 on 605 windows with approximately 20% reserved for validation, also included a moving-idle class. On validation, this class absorbed 16.7% of backhand-topspin windows, suggesting that some inter-stroke transition motion still carried stroke labels, and it was therefore removed from the final model.

TABLE I
Dataset Composition Across the Two Independent Sessions. Stroke Counts Are Gate-Detected Events (150 °/s Peak, 400 ms Refractory).

	S1 (train)		S2 (test)	
Class	Dur. (s)	Strokes	Dur. (s)	Strokes
Backhand drive	58.6	88	23.6	31
Backhand topspin	70.6	139	23.5	51
Forehand drive	58.8	91	23.4	36
Forehand topspin	58.8	116	23.6	47
Idle (still)	46.6	—	23.4	—
Idle (moving)*	44.5	—	22.4	—
Total (29+12 rec.)	337.9	434	139.9	165

Durations are wall-clock from device sequence timestamps and therefore include transmission gaps (Section III-A) where sample-count durations are ≈1% shorter. From the received S1 samples the toolchain formed 605 windows, ≈20% of which were reserved for validation. *Note: Idle (moving) was recorded during dataset acquisition but removed from the final 5-class deployed model (Section III-C).

An orientation (Kalman) filter appeared in the initial architecture and was removed since it estimates absolute orientation, which a spectral classifier does not consume, and its smoothing attenuates exactly the transient content that discriminates strokes.

D. On-Device Inference and Stroke Detection

Deployment was treated as an experiment in its own right, and two acquisition defects were found by measurement before the detector was finalised. First, polling the IMU on a rigid 10 ms grid beat against the sensor's 99.84 Hz data-rate clock, silently discarding ≈45% of samples: per-window timestamping showed nominal one-second windows spanning 1.7–1.9 s at 52–60 Hz — the second appearance of the rate-failure class met in Section III-A. The sampler was made sensor-paced and the internal I^2C bus (Appendix H) raised to 400 kHz, restoring ≈100 Hz live; as a permanent guard, windows whose measured span falls outside 850–1200 ms are rejected before inference by the window guard (Appendix K).

Second, timer-driven inference was found to grade *incomplete* motions: a forehand backswing is backhand-direction rotation, and was being labelled before ball contact. The final detector therefore segments swings and classifies each exactly once (Fig. 2): angular rate above 250 °/s opens a swing event and its peak is tracked; the swing completes when the rate stays below 120 °/s for 100 ms; classification

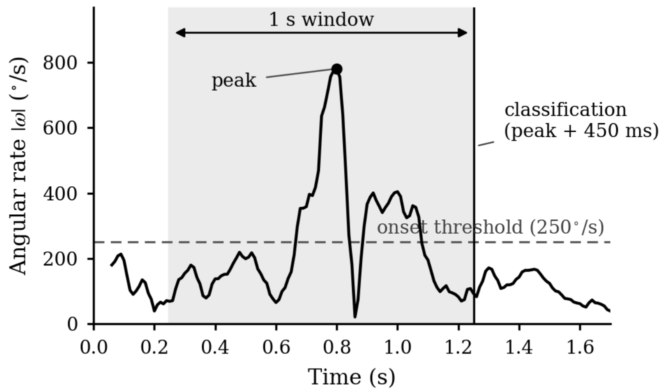

Fig. 2. Swing segmentation on a measured backhand-topspin swing (session S1, peak 780°/s): crossing the onset threshold opens a swing event; classification runs once, 450 ms after the tracked peak, on the trailing 1 s window (peak −550 ms to +450 ms).

runs once, ≈450 ms after the peak, so the entire stroke (backswing, contact and follow-through) sits centred in the 1 s window, mirroring both the training windowing statistics and the peak-centred segmentation of [1]. A 0.75 confidence threshold filters weak verdicts; swing-to-verdict latency is ≈0.5 s. Above this sits the rally state machine (3 s timeout, minimum two strokes); stroke events and rally summaries are sent as compact 8-byte BLE notifications. In the live demonstrator, forehand-topspin verdicts are suppressed at the decision layer (Section IV-D).

E. *Feedback Interface*

A browser dashboard connects over Web Bluetooth and renders live per-class counts, the ordered stroke sequence, and per-rally summaries. The browser route was chosen because it requires no installation, speaks BLE directly, and this MCU has no WiFi.

IV. Results and Evaluation

A. *Classification Performance*

Table II reports the deployed model's full held-out confusion matrix (Appendix L) on session S2 (44 windows per class), recorded after a physical re-mount and sharing no swings with training: **90.91%** accuracy, weighted precision, recall and F1 all 0.93, AUC 0.99. The float32 and INT8 versions score identically on this set — quantisation is lossless at the decision level — so the deployed INT8 network carries the full result. The toolchain's confidence threshold leaves 5.0% of windows undecided ("uncertain") rather than forcing a label; among decided windows, the largest confusions are forehand topspin→backhand topspin (4.5%) and the drive pair (4.5%/2.3%). The intermediate six-class variant's validation accuracy was 89.3%; that a re-mounted, session-independent test exceeds it reflects the removal of the weakest (moving-idle) class rather than optimism in the protocol. The single-session held-out set is modest, so per-class figures should be read as indicative.

TABLE II

Held-Out (S2) Confusion Matrix of the Deployed Five-Class INT8 Model, Row-Normalised %. Overall Accuracy 90.91%; Weighted Precision, Recall and F1 All 0.93; AUC 0.99.

True ↓	Still	BH dr.	BH ts.	FH dr.	FH ts.	Unc.
Still (idle)	**88.6**	0	0	2.3	0	9.1
Backhand drive	4.5	**88.6**	0	4.5	0	2.3
Backhand topspin	2.3	0	**97.7**	0	0	0
Forehand drive	0	2.3	0	**90.9**	0	6.8
Forehand topspin	0	0	4.5	0	**88.6**	6.8
Per-class F1	0.91	0.93	0.97	0.92	0.94	—

Rows: true class, 44 windows each. "Unc.": below the toolchain's confidence threshold (5.0% of all windows), counted as incorrect in the 90.91% figure.

B. *Error Analysis and Frame Sensitivity*

The error structure inverts the literature's prediction. Where prior work reports same-wing strokes as the dominant confusion [2], the six-class variant's validation matrix showed same-side confusion as minor (one pair at 4.2%) against cross-side errors up to 12.5% and the held-out five-class matrix (Table II) replicates the inversion in stronger form: same-side confusion is *absent*, and every residual stroke error is cross-side (4.5%, 4.5% and 2.3%). Two design properties plausibly explain this: the features are magnitude spectra, invariant to a sign flip of an axis, so the approximately mirrored kinematics of the same stroke on opposite wings converge in feature space; and the deliberately fixed single-orientation mount (Section II-E) ensures no orientation shortcut re-separates the wings.

Deployment furnished direct evidence for this mechanism: a live forehand-topspin window scoring 0.000 for its own class as captured reached 0.973 after nothing but a rigid rotation of the sensor frame, replayed bit-exact through the deployed pipeline; and the two ready stances are measurably distinct in the training data (mean resting a_z $+0.50$ forehand-ready versus -0.40 backhand-ready), i.e. stance attitude is itself a learned cue. The feature set is frame-sensitive by construction: rotating the input moves samples across class boundaries.

Motion-similarity error must be separated from sensor error. At ±16 g no recording in the final dataset clips (peak 5.9 g), and in the earlier ±4 g investigation the most-clipped class scored *best* while the least-clipped scored worst, so sensor range is not the dominant error term, and the residual confusions above are attributable to genuine kinematic similarity and labelling noise rather than transduction artefacts.

C. *Embedded and Link Performance*

Table III summarises on-device and link measurements. The spectral feature extraction executes in 20 ms within a 3 kB DSP peak RAM footprint (Appendix M) on the Cortex-M4F, comfortably inside the 200 ms inference stride and the 256 kB SRAM budget. The link figures are those measured across the entire final dataset (Section III-A) rather than a bench test. Live acquisition figures are post-fix measurements from the deployed detector: the sensor-paced sampler sustains 99–100 Hz on the racket, the span guard admits only windows of 850–1200 ms, and a stroke verdict is reported ≈0.5 s after swing completion. Together these are the evidence that the

TABLE III
ON-DEVICE RESOURCE UTILISATION AND WIRELESS LINK PERFORMANCE.

Metric	Measured
DSP (spectral) processing time	20 ms
DSP peak RAM	3 kB
NN inference latency	2 ms
NN peak RAM / flash	1.8 kB / 14.2 kB
Live sampling (post-fix)	99–100 Hz
Window integrity guard	850–1200 ms accepted
Swing-to-verdict latency	$\approx$0.5 s
Dataset sample rate	99.1 Hz effective
Packet delivery	99.07% (446/47,804 lost)
Worst observed gap	50 ms

untethered, edge-inference claim holds in the conditions the system is actually used in.

D. Live Deployment and the Forehand-Topspin Gap

Live testing weeks after data collection reproduced every class except forehand topspin. The fault was isolated by construction. The deployed INT8 pipeline, compiled unmodified for a desktop host and fed the held-out S2 recordings through the firmware's exact windowing, scored forehand topspin at a mean probability of 0.90, thereby testing the artifact, quantisation, DSP, and labels together. Captured live windows, when replayed through the same pipeline, reproduced the device's outputs byte-for-byte, confirming correct on-device inference. An exhaustive rigid-rotation search over live windows still failed to recover the class, indicating that the live forehand-topspin input alone had drifted outside the distribution of the single July session, rather than that any pipeline component had failed. The demonstrator therefore reports four classes, with forehand topspin suppressed at the decision layer; condition-aligned data top-up and retraining are the identified remedy (Section V-C). This negative result is reported deliberately because the testing sequence is what localised the fault.

E. Comparison with Prior Work

Direct comparison needs care: Blank *et al.*'s 96.7% covers eight classes from ten players, classified offline [1]; Ferreira *et al.*'s 97.33% F1 uses a ConvLSTM on wrist-worn data at 50 Hz [2]. Neither performs quantised inference on the racket, and published evaluations are frequently not session-independent, whereas 90.91% here is from a physically re-mounted held-out session — a deployment-realistic estimate traded for a few in-session points, with full wireless operation and microcontroller inference; the final detector also adopts at deployment the peak-centred segmentation applied offline in [1].

V. CONCLUSION

A. Project Aims and Objectives

Against the objectives of Section I: Objective 1 is met — the review identified the same-wing-confusion prediction against which this system's contrasting, predominantly cross-side error structure is interpreted (Section IV-B), and the flat-strain transduction requirement that decided the piezo question. Objective 2 is partially met: the racket-integrated wireless prototype was delivered and used for all data collection, but blade impact sensing was de-scoped on technical grounds. The purchased LDT2-028K elements are vibration-mode sensors with $\approx$61 mm active length, unable to span the $\approx$150 mm blade as the flat-strain row–column method requires [5]. Objective 3 is partially met: an on-device stroke classifier meeting real-time constraints was delivered; hit position and spin follow the piezo de-scope. Objective 4 is partially met: evaluation used realistic play and a session-independent protocol, but was single-participant; multi-player testing requires the ethics process identified during the project. Objective 5 is met through this report and the accompanying deliverables. A justified de-scope is reported here as exactly that, with its evidence.

B. Significant Technical Achievement

Three achievements stand above the rest. First, a fully untethered racket running quantised neural inference on-device, wrapped in a staged detector and rally state machine that turn window classifications into stroke-level feedback. Second, the driver-level full-scale-range fix, including the coupled conversion constant, diagnosed from railed data and verified against gravity. Third, and the one with most transferable value, a measurement-first methodology: genuinely independent evaluation sessions adopted after diagnosing that an earlier perfect score was leakage; link and timing integrity instrumentation that caught two silent sample-rate regressions; and bit-exact off-target replay of the deployed artifact, which localised a live failure to the input distribution by testing every pipeline component with experiments rather than argument.

C. Future Work

Four concrete extensions follow directly from the evidence. First, forehand topspin should be restored to the live demonstrator through a condition-aligned data top-up, consisting of a short recording session in the final mounting configuration added to the existing dataset, followed by retraining. More generally, periodic recalibration against acquisition drift would also be useful, with frame normalisation of the input as a complementary direction. Second, blade-mounted hit-position and spin sensing should be revisited using the correct transduction method, namely flat-strain PVDF film, with either full-blade coverage or the reduced 10×10 cm monitored area used in prior work [5], [7]. Third, the system should be evaluated across multiple participants and skill levels under the required ethics approval. Fourth, although the power subsystem has been specified and verified, battery life during sustained play remains to be measured.

D. Reflection

The decisions I value most from this project were acts of judgement rather than construction. Treating a 100% result as a warning, and rebuilding the evaluation protocol it invalidated, changed how I interpreted every later result. Rejecting the easier flipped-board mounting to keep the gravity vector out of the labels was a hardware decision made purely to protect data validity. When the accelerometer first appeared dead, the obvious explanation was wrong twice: the signal was saturated, not small, and when clipping was identified, quantifying it showed that the most-clipped class actually classified best. This partly challenged the assumption that the sensor was the main accuracy bottleneck. Two things I would do differently are verifying a sensor's transduction mode before purchase, since the piezo de-scope came directly from not doing so, and collecting an idle class from the first session, since relying only on a motion-magnitude gate without a learned still-idle class proved insufficient. The deployment phase reinforced this approach. When live results contradicted every offline result, the crucial step was to test and verify by capturing the exact bytes seen by the device and replaying them elsewhere. The key insight, that the detector was classifying half-finished swings, came from watching it fail in real time. More than anything else, this project changed my default approach from assuming the pipeline worked to instrumenting it so that it had to prove it worked.

Acknowledgment

The author expresses sincere gratitude to supervisor Dr Craig Evans for his continuous support, technical guidance, and encouragement throughout the engineering and development of this project. Special thanks are also extended to the author's father, Mr. Sandeep Kumar Arya, whose early introduction to table tennis sparked the initial interest that inspired this research, and to the author's mother, Mrs. Toolika Johri Arya, whose constant encouragement and efforts guided him towards the feild of science and engineering.

Use of Generative AI

Generative AI tools (Anthropic Claude and Microsoft Copilot) were used, as authorised for this module, to assist with the execution and documentation of this project. Anthropic Claude was utilised for hardware setup guidance, report drafting, and editing. Microsoft Copilot was used for presentation preparation, Bill of Materials (BOM) structuring, data analysis extraction, and scripting in LaTeX. All system design, physical implementation, experiments, and conclusions are the author's own, and all AI-assisted outputs, code, and computed figures were independently reviewed and verified by the author against the project's primary data.

References

[1] P. Blank, J. Hoßbach, D. Schuldhaus, and B. M. Eskofier, "Sensor-based stroke detection and stroke type classification in table tennis," in *Proc. ACM Int. Symp. Wearable Computers (ISWC)*, Osaka, Japan, Sep. 2015, pp. 93–100, doi: 10.1145/2802083.2802087.

[2] N. M. Ferreira, J. M. Torres, P. Sobral, R. Moreira, and C. Soares, "Classification of table tennis strokes in wearable device using deep learning," in *Proc. 14th Int. Conf. Agents and Artificial Intelligence (ICAART)*, vol. 3, 2022, pp. 629–636, doi: 10.5220/0010871100003116.

[3] S. S. Tabrizi, S. Pashazadeh, and V. Javani, "A deep learning approach for table tennis forehand stroke evaluation system using an IMU sensor," *Comput. Intell. Neurosci.*, vol. 2021, Art. no. 5584756, 2021, doi: 10.1155/2021/5584756.

[4] K. Xia, H. Wang, M. Xu, Z. Li, S. He, and Y. Tang, "Racquet sports recognition using a hybrid clustering model learned from integrated wearable sensor," *Sensors*, vol. 20, no. 6, Art. no. 1638, 2020, doi: 10.3390/s20061638.

[5] C. Zhang, Z. Fan, Y. Kang, *et al.*, "Piezoelectric row–column sensing system on table tennis rackets for hit and rotation measurement," *PLoS One*, vol. 20, no. 10, Art. no. e0333735, 2025, doi: 10.1371/journal.pone.0333735.

[6] T. Yamashita and T. Kobayashi, "Smart ping pong racket by ultrathin piezoelectric strain sensor array," in *Proc. Symp. Design, Test, Integration and Packaging of MEMS and MOEMS (DTIP)*, Roma, Italy, 2018, pp. 1–3, doi: 10.1109/DTIP.2018.8394237.

[7] T. Yamashita and T. Kobayashi, "Smart table tennis racket using a rubber mounted ultrathin piezoelectric sensor array," *Sensors and Materials*, vol. 33, no. 3, pp. 1081–1089, 2021, doi: 10.18494/SAM.2021.2954.

[8] Y. Hu, Y. Li, B. Cui, H. Su, and P. Zhu, "Internet of things enabled deep learning monitoring system for real-time performance metrics and athlete feedback in college sports," *Sci. Rep.*, vol. 15, no. 1, Art. no. 28405, 2025, doi: 10.1038/s41598-025-13949-6.

[9] Bosch Sensortec, "BMI270: 6-axis inertial measurement unit," datasheet, Doc. BST-BMI270-DS000-08, 2020. [Online]. Available: https://www.bosch-sensortec.com.

[10] Arduino, "Nano 33 BLE Sense Rev2," ABX00070 product documentation, 2022. [Online]. Available: https://docs.arduino.cc.

[11] Edge Impulse, "Spectral features and neural network classifier documentation," 2024. [Online]. Available: https://docs.edgeimpulse.com.

Supplementary Material

Supplementary material, including full firmware source code, dataset processing scripts, CAD files for the 3D-printed handle, and trained model artifacts, is hosted in a public GitHub repository available at: https://github.com/aaruche/smart-table-tennis-racket.

Appendix A
Float32 and INT8 Quantisation

Float32 (32-bit floating point) is standard computer representation for decimals. **INT8 Quantisation** compresses model parameters (weights and biases) from 32-bit floating numbers down to 8-bit integers (-128 to $+127$). In this project, quantisation drops model size by approximately $4\times$ and speeds up mathematical calculations on the Cortex-M4F microcontroller without loss of stroke classification accuracy (both achieved 90.91% on test session S2).

Appendix B
Dense Neural Network (DNN)

A **Dense Neural Network** (also called a Fully-Connected Network) is a machine learning structure where every neuron in one layer connects to every neuron in the next layer. Here, a compact network receives 78 log-spectral motion features and maps them through hidden layers to output class probabilities for table tennis strokes.

Appendix C
Weighted F1 Score

The **F1 Score** combines precision (how many detected strokes were correct) and recall (how many real strokes were successfully caught) into a single score from 0 to 1. A **Weighted F1 Score** calculates the F1 score for each individual stroke class and averages them while weighting each score by the number of true instances in that class, accounting for minor dataset imbalances.

Appendix D
Support Vector Machine (SVM)

A **Support Vector Machine** is a classic machine learning algorithm that classifies data by finding an optimal decision boundary (hyperplane) between different groups. It was used in foundational literature (e.g., Blank *et al.* [1]) as an offline classifier for table tennis stroke recognition.

Appendix E
ConvLSTM

A **ConvLSTM** (Convolutional Long Short-Term Memory) network combines Convolutional layers (which extract spatial patterns) with LSTM recurrent units (which track timing sequences). Prior literature (e.g., Ferreira *et al.* [2]) used ConvLSTMs to extract temporal patterns from wearable IMUs, though they are computationally heavy for on-racket microcontroller deployment.

Appendix F
Piezoelectric Array Sensor

A **Piezoelectric Sensor** generates an electrical voltage when subjected to physical mechanical pressure or strain. In table tennis research, thin film arrays are laminated under the rubber to measure impact location and ball spin. This feature was descoped in this project because commercial discrete elements (LDT2-028K) measure beam bending rather than flat surface strain over the full blade surface.

Appendix G
Full-Scale Range (FSR)

Full-Scale Range defines the maximum positive and negative acceleration or angular velocity boundaries a sensor can measure before its output saturates ("rails"). In this system, the accelerometer FSR was raised from ± 4 g to ± 16 g in the driver to prevent signal clipping during fast table tennis swings.

Appendix H
Inter-Integrated Circuit (I^2C)

I^2C is a two-wire serial communication bus used to transfer data between the BMI270 sensor and the nRF52840 microcontroller. The bus speed was upgraded from 100 kHz (Standard-mode) to 400 kHz (Fast-mode) to prevent data bottlenecks during real-time 100 Hz motion sampling.

Appendix I
Spectral Analysis

Spectral Analysis converts time-domain signals (raw sensor values over time) into the frequency domain using the Fast Fourier Transform (FFT). Instead of processing raw acceleration points, the system extracts 78 log-spectral features representing the frequency composition and power of swing movements.

Appendix J
Angular Rate

Angular Rate (measured in degrees per second, °/s) represents how fast the racket rotates around an axis as measured by the gyroscope. Peak angular rate (e.g., 966 °/s during topspin) is used by the system to trigger stroke detection windowing.

Appendix K
Window Guard

The **Window Guard** is a safety check in firmware that measures the actual physical time elapsed during a collected 100-sample window. If a timing drop or bus lag causes a window duration to fall outside the acceptable 850–1200 ms interval, the window is rejected prior to inference to prevent corrupting model predictions.

Appendix L
Confusion Matrix

A **Confusion Matrix** is a table used to evaluate machine learning performance by comparing true target classes against model predictions. Rows represent true stroke labels, while columns represent predicted classes, clearly exposing misclassifications (e.g., cross-side forehand/backhand errors).

Appendix M
DSP Peak RAM

DSP Peak RAM refers to the maximum temporary static working memory required by the Digital Signal Processing algorithm while computing FFT frequency spectra. On the Cortex-M4F microcontroller, feature extraction requires a minimal 3 kB RAM footprint out of the 256 kB available.

Appendix N
Inertial Measurement Unit (IMU)

An **IMU** is an electronic sensor module combining a 3-axis accelerometer (measuring linear forces) and a 3-axis gyroscope (measuring rotational speed). The onboard Bosch BMI270 IMU provides the raw 6-axis motion data for stroke analysis.

Appendix O
TinyML

TinyML is the practice of running machine learning inference directly on tiny, low-power microcontrollers (such as ARM Cortex-M microcontrollers) rather than relying on external computers or cloud servers.