CatMotion
Cat Behavior Classification
Final Demo

Lin Zhan, Junrui Zhao

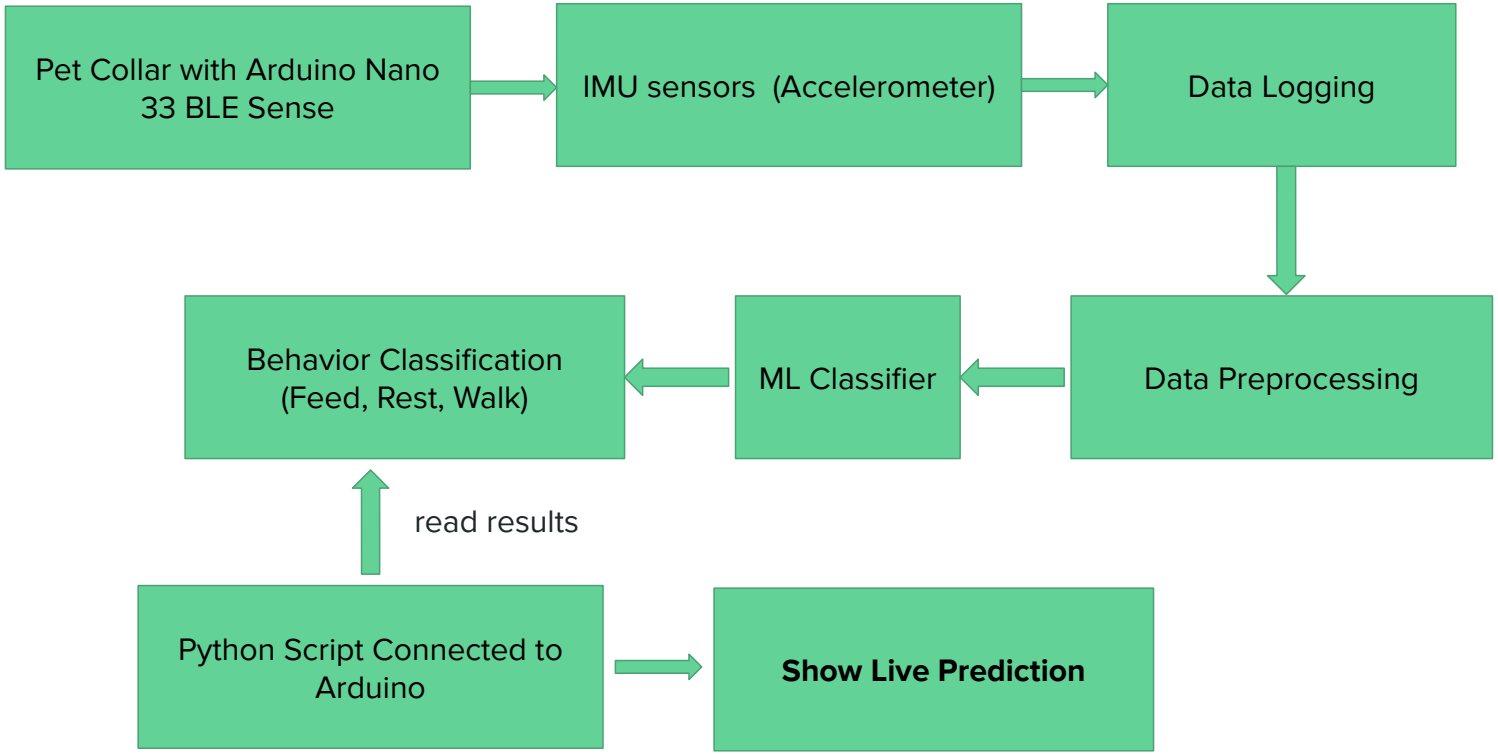
Project Overview

- Goal: Develop a wearable device that classifies cat behaviors (e.g., rest, walk, feed) using motion sensor data.
- Type: New product: a cat-mounted smart collar system.
- Motivation: Catch early changes that may signal cat health issues.
- Objective: Enable continuous, real-time cat behavior tracking for activity analysis and health monitoring.
- Target users: Cat owners (especially busy owners), vets, and rescue groups who need continuous activity and behavior tracking.

Why use Embedded ML? BLERP model

- B - Bandwidth
  - Our system processes raw IMU data locally on the Arduino Nano 33 BLE Sense.
- L - Latency
  - Immediate recognition of cat activities, enabling future extensions like alerting or live tracking.
- E - Economics
  - The Nano 33 BLE Sense is optimized for low-power operation, allowing the collar to run for extended periods and thus reducing the overall cost.
- R - Reliability
  - The collar continues to classify behaviors offline, without depending on a network.
- P - Privacy
  - No cloud uploads of pet activity, preserving user and pet privacy.

Block Diagram



Methods - Data collection

- Dataset: public + self-collected
- Public dataset: Domestic cat accelerometer data calibrated with behaviors (Dryad repository).
- Self-collected data using Arduino Nano 33 BLE Sense recording 3-axis accelerometer readings at 40 Hz.

Methods - ML models

- Knowledge Distillation
- Train a strong Random Forest (teacher) on extracted feature vectors
- Teacher outputs soft class probabilities (not just hard labels)
- Train a lightweight neural network (student) via knowledge distillation
- Loss = (1 − α) · cross-entropy (hard labels) + α · distillation loss (match teacher distribution with temperature T)
- Export the trained student as TensorFlow Lite with INT8 quantization for deployment

Methods - Embedded systems implementations

Our system consists of an Arduino Nano 33 BLE Sense running a custom .ino sketch and a Python script on PC using bleak to receive BLE notifications.

- Arduino collects 3-axis accelerometer data continuously at the model's required sampling rate
- Maintains a sliding inference window (predicts every 0.5 seconds)
- Buffers all predictions inside a 4-second smoothing window
- Applies majority voting across the 4-second window to generate a stable, smoothed behavior label
- Sends the final 4-second predicted label to the laptop via BLE notifications (and optionally Serial)
- Python script (BLE/bleak) receives, displays, and logs predictions in real time

Results

- Overall performance: Accuracy 74.7%, with AUC 0.93 and weighted F1 0.78 ➔ good separability.
- Strong classes: Feed 91.7% (F1 0.87), Rest 81.3% (F1 0.77) ➔ reliable for the most common behaviors.
- Main error case: Walk ↔ Rest confusion (Walk➔Rest 28.8%) — expected for IMU-only signals

Validation

Model — Model version: Quantized (int8)

Last training performance (validation set)

ACCURACY	LOSS
74.8%	0.66

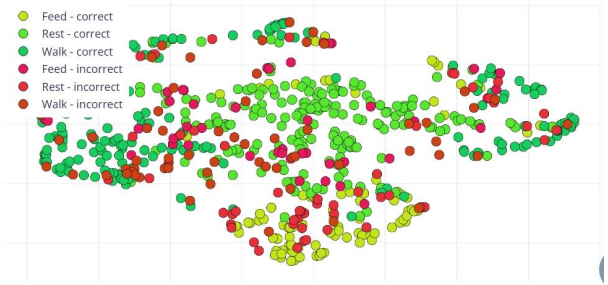
Confusion matrix (validation set)

	FEED	REST	WALK
FEED	80.6%	9.7%	9.7%
REST	18.9%	71.7%	9.4%
WALK	7.8%	17.6%	74.5%
F1 SCORE	0.71	0.74	0.78

Metrics (validation set)

METRIC	VALUE
Area under ROC Curve	0.91
Weighted average Precision	0.76
Weighted average Recall	0.75
Weighted average F1 score	0.75

Data explorer (full training set)



Testing

Results — Model version: Quantized (int8)

ACCURACY
74.70%

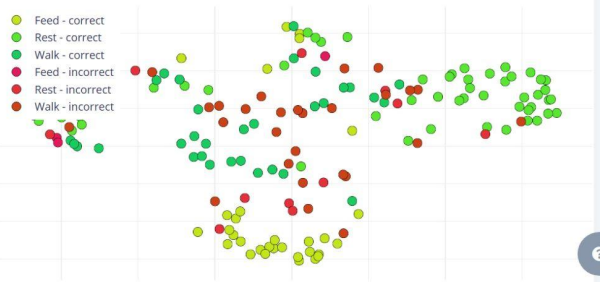
Metrics for rf-cat-block

METRIC	VALUE
Area under ROC Curve	0.93
Weighted average Precision	0.80
Weighted average Recall	0.78
Weighted average F1 score	0.78

Confusion matrix

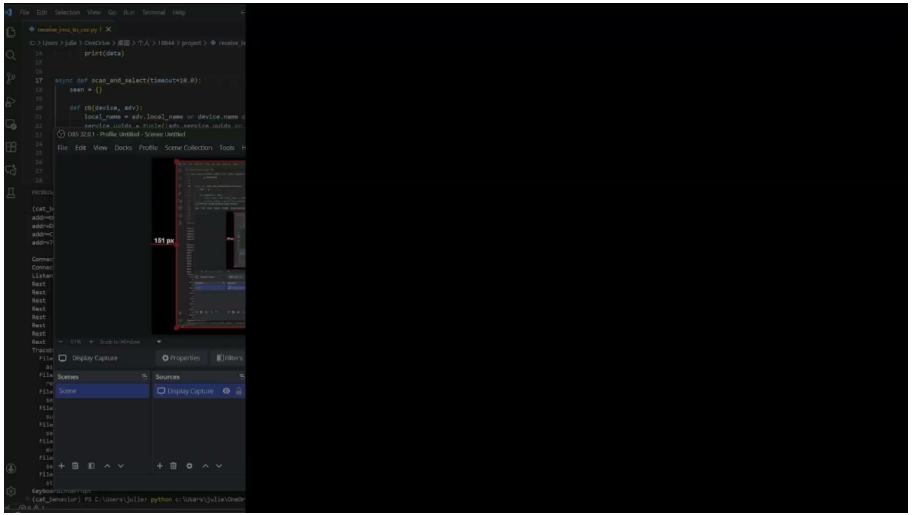
	FEED	REST	WALK	UNCERTAIN
FEED	91.7%	0%	5.6%	2.8%
REST	6.3%	81.3%	3.1%	9.4%
WALK	4.5%	28.8%	59.1%	7.6%
F1 SCORE	0.87	0.77	0.72	

Feature explorer



Deployment

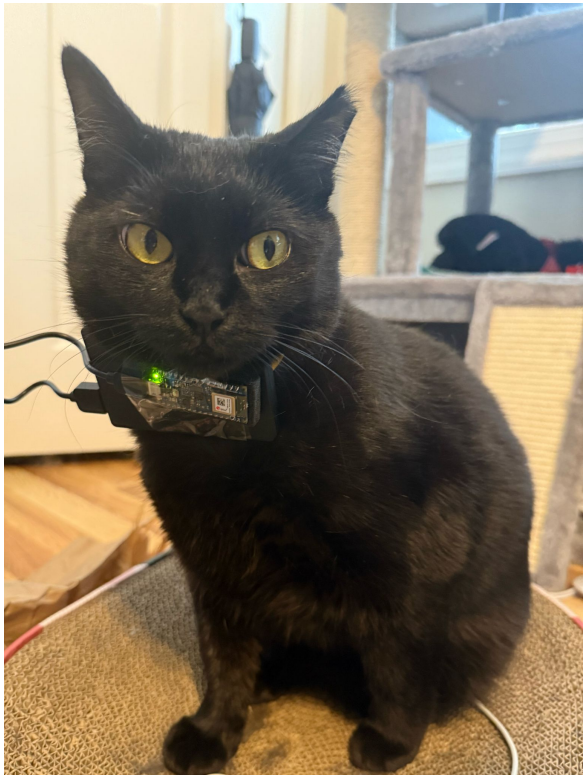
	SPECTRAL FEATURES	RF-CAT-BLOCK	TOTAL
LATENCY	4 ms.	2 ms.	**6 ms.**
RAM	2.0K	-	**2.0K**
FLASH	-	-	**-**
ACCURACY			**74.70%**

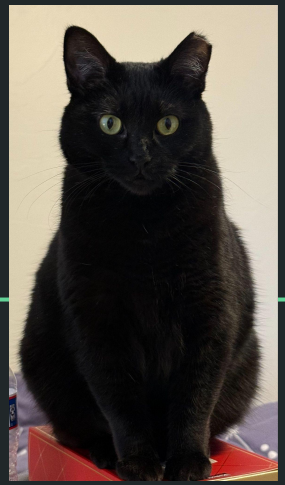


Challenges

- **Rest vs. slow walk**: “Rest” is not pure stillness, so it overlaps with slow walk. This is a hard boundary case that our model may not fully resolve.
- **Cross-cat variation** (different cat size) may hurt generalization. we currently cannot test on many different cats/breeds/sizes, so it is unclear whether the model stays steady when the cat body size and collar fit change a lot.
- **Behavior transitions and mixed windows.** Real life is continuous; many windows contain two behaviors, which makes precise separation and clean classification harder.

Live Demo





Thank you