Robust Mouse Behavior Recognition using FPS-Aware Multi-Scale Features and Ensemble Gradient Boosting

Nguyễn Đỗ Trọng Nghĩa
Mã sinh viên: 23020125
Email: 23020125@vnu.edu.vn

Nguyễn Thành Long
Mã sinh viên: 23020104
Email: 23020104@vnu.edu.vn

Nguyễn Quốc Minh
Mã sinh viên: 23020116
Email: 23020116@vnu.edu.vn

Tóm tắt nội dung*—Bài toán phát hiện hành vi chuột tự động từ dữ liệu theo dõi chuyển động là một thách thức quan trọng trong nghiên cứu hành vi động vật và khoa học thần kinh. Báo cáo này trình bày hệ thống hoàn chỉnh để nhận dạng hành vi xã hội của chuột từ dữ liệu video được ghi nhận trong cuộc thi MABe 2025 trên nền tảng Kaggle [1]. Hệ thống đề xuất kết hợp mô hình học máy ensemble gồm LightGBM, XGBoost và CatBoost với kỹ thuật trích xuất đặc trưng FPS-aware. Đặc biệt, phương pháp đảm bảo tính nhất quán của đặc trưng giữa các video có tốc độ khung hình khác nhau bằng cách chuẩn hóa tất cả cửa sổ thời gian và độ trễ theo fps thực tế của từng video. Kỹ thuật stratified sampling được áp dụng để huấn luyện hiệu quả trên tập dữ liệu lớn trong khi vẫn duy trì phân phối lớp cân bằng. Hệ thống xử lý cả hành vi cá thể đơn lẻ và tương tác giữa các cặp chuột, tạo ra dự đoán theo frame-level và chuyển đổi thành các sự kiện liên tục với start/stop frame. Kết quả thực nghiệm cho thấy phương pháp đạt điểm 0.449 trên tập test công khai của cuộc thi. Mô hình: GitHub repository

I. Giới thiệu

A. Bối cảnh vấn đề

Nghiên cứu hành vi xã hội của động vật, đặc biệt là chuột phòng thí nghiệm, đóng vai trò quan trọng trong khoa học thần kinh, dược phẩm và tâm lý học hành vi. Truyền thống, việc ghi nhận và phân tích hành vi được thực hiện thủ công bởi các chuyên gia, đòi hỏi thời gian dài và dễ phát sinh sai sót do yếu tố chủ quan. Với sự phát triển của công nghệ thị giác máy tính và học sâu, việc tự động hóa phân tích hành vi từ video đã trở thành hướng nghiên cứu tiềm năng.

Cuộc thi MABe 2025 (Multi-Agent Behavior Challenge) trên nền tảng Kaggle đặt ra thách thức phát hiện và phân đoạn hành vi chuột từ dữ liệu theo dõi tọa độ các bộ phận cơ thể. Dữ liệu được thu thập từ nhiều phòng thí nghiệm với các điều kiện ghi hình và cấu hình tracking khác nhau, tạo nên độ đa dạng cao về chất lượng và định dạng đầu vào.

B. Động lực và mục tiêu

Động lực chính của thực nghiệm này bắt nguồn từ nhu cầu xây dựng hệ thống nhận dạng hành vi có khả năng tổng quát hóa tốt trên dữ liệu đa dạng. Các thách thức chính bao gồm: Sự khác biệt về tốc độ khung hình (fps) giữa các video; Mất cân bằng lớp nghiêm trọng do một số hành vi hiếm gặp; Dữ liệu tracking chất lượng khác nhau giữa các phòng thí nghiệm; Yêu cầu dự đoán theo frame-level với độ chính xác cao.

Mục tiêu của bài tập lớn là thiết kế và triển khai pipeline hoàn chỉnh bao gồm tiền xử lý dữ liệu, kỹ thuật feature engineering phù hợp, mô hình học máy mạnh mẽ và cơ chế hậu xử lý thông minh để đạt được hiệu suất cao trên tập test công khai.

C. Đóng góp chính

Những đóng góp chính của thực nghiệm bao gồm:

1) **FPS-aware feature engineering**: Tất cả đặc trưng thời gian được chuẩn hóa theo fps thực tế của video, đảm bảo tính nhất quán ngữ nghĩa giữa các video có tốc độ khác nhau.
2) **Stratified subset sampling**: Kỹ thuật lấy mẫu có trọng số theo phân tầng giúp huấn luyện nhanh trên tập con dữ liệu mà vẫn giữ được phân phối lớp.
3) **Ensemble với early stopping tự động**: Kết hợp nhiều mô hình boosting với cơ chế tự động lựa chọn metric và patience dựa trên tỷ lệ lớp.
4) **Robustness handling**: Xử lý các trường hợp biên như video không có dự đoán hoặc các đoạn dự đoán chồng lấn.

II. Các công trình liên quan

A. Phân tích hành vi động vật tự động

Lĩnh vực phân tích hành vi động vật đã chứng kiến nhiều tiến bộ trong thập kỷ qua. Các phương pháp truyền thống dựa vào các đặc trưng thủ công như khoảng cách giữa các bộ phận cơ thể, vận tốc và gia tốc [2]. Gần đây, các mô hình học sâu như LSTM và Transformer được áp dụng để học biểu diễn tự động từ chuỗi thời gian tọa độ tracking [3], [4].

Tuy nhiên, các phương pháp học sâu thường đòi hỏi lượng dữ liệu huấn luyện lớn và tài nguyên tính toán đáng kể. Trong bối cảnh cuộc thi với thời gian và tài nguyên hạn chế, các thuật toán gradient boosting như LightGBM [5] và XGBoost [6] vẫn là lựa chọn phổ biến nhờ khả năng huấn luyện nhanh và hiệu suất tốt trên dữ liệu dạng bảng.

B. Feature engineering cho dữ liệu chuyển động

Trích xuất đặc trưng từ dữ liệu theo dõi chuyển động là bước quan trọng ảnh hưởng trực tiếp đến hiệu suất mô hình [2], [8]. Các nghiên cứu trước đây tập trung vào các đặc trưng hình học (khoảng cách, góc), động học (vận tốc, gia tốc, độ cong quỹ đạo) và thống kê thời gian (mean, std, min, max trong các cửa sổ trượt).

Tuy nhiên, vấn đề chưa được chú ý đúng mức là sự không nhất quán giữa các video có fps khác nhau. Không chuẩn hóa fps có thể khiến đặc trưng trở nên vô nghĩa khi cùng số frame ứng với độ dài thời gian khác nhau. Nếu không chuẩn hóa, một cửa sổ 30 frame có thể tương ứng với 1 giây ở video 30fps nhưng chỉ 0.5 giây ở video 60fps, dẫn đến đặc trưng mất ý nghĩa thời gian. Thực nghiệm này giải quyết vấn đề bằng cách scale tất cả tham số thời gian theo fps thực tế.

C. Xử lý mất cân bằng lớp

Do đặc thù dữ liệu hành vi, một số hành vi hiếm gặp dẫn đến phân phối lớp mất cân bằng nghiêm trọng. Những phương pháp như oversampling [9], undersampling, hoặc sử dụng các metric như AUPRC [10] thường được áp dụng. Thực nghiệm trong báo cáo sử dụng sự kết hợp của nhiều chiến lược như tự động tính class weights, và lựa chọn metric early stopping dựa trên phân phối lớp.

D. So sánh với các phương pháp trước

Phương pháp đề xuất có một số ưu điểm nổi bật so với các tiếp cận trước đây trong các cuộc thi tương tự [3], [4], [8]:

- Khả năng khái quát hóa cao nhờ đặc trưng FPS-aware và chiến lược sampling hợp lý [11].
- Hệ thống tự động thích ứng với từng hành vi và từng mô hình thông qua early stopping.
- Cân bằng tốt giữa chi phí huấn luyện và hiệu năng thông qua stratified subset sampling.
- Hậu xử lý robust đảm bảo submission luôn hợp lệ và nhất quán.

III. Phương pháp đề xuất

A. Tổng quan hệ thống

Hệ thống được xây dựng theo pipeline gồm các thành phần: tiền xử lý dữ liệu, trích xuất đặc trưng, huấn luyện mô hình boosting dạng ensemble, suy luận hành vi frame-level, và hậu xử lý để tạo sự kiện liên tục.

Pipeline gồm 5 giai đoạn chính:

1) ***Tiền xử lý và lọc dữ liệu***: Loại bỏ các video có điều kiện không phù hợp (chuột ngủ), chuẩn hóa tọa độ theo pixel-per-cm.
2) ***Feature engineering***: Trích xuất đặc trưng hình học, động học và thời gian với chuẩn hóa fps.
3) ***Model training***: Huấn luyện ensemble gồm nhiều mô hình boosting với stratified sampling.
4) ***Inference và thresholding***: Dự đoán xác suất frame-level, áp dụng thresholding và làm mượt.
5) ***Post-processing***: Chuyển đổi thành sự kiện, xử lý chồng lấn, bổ sung dự đoán cho video trống.

B. Tiền xử lý và lọc dữ liệu

1) Lọc dữ liệu huấn luyện: Một số video từ môi trường "lights on"của lab MABe22 cho thấy chuột chủ yếu ngủ và không có hành vi liên quan. Những video này được loại bỏ để tránh gây nhiễu cho mô hình.

2) Chuẩn hóa tọa độ: Tọa độ các bộ phận cơ thể được quy đổi từ pixel sang centimet dựa trên tham số "pixel-per-cm" của từng video nhằm đảm bảo tính nhất quán giữa các điều kiện ghi hình khác nhau.

3) Xử lý body parts khác nhau: Các video có thể theo dõi số lượng bộ phận cơ thể khác nhau. Một số video theo dõi đầy đủ hơn 20 điểm, trong khi số khác chỉ theo dõi 5-8 điểm cơ bản. Hệ thống tự động nhóm dữ liệu theo body_parts_tracked và xử lý riêng từng nhóm, đảm bảo mỗi mô hình chỉ làm việc với các bộ phận có sẵn.

C. Trích xuất đặc trưng

Hệ thống trích xuất đặc trưng được thiết kế để bao phủ toàn diện các khía cạnh hành vi với cơ chế thích ứng FPS làm nền tảng. Các đơn vị vật lý được sử dụng thống nhất bao gồm: khoảng cách (d) tính bằng centimet (cm), thời gian (t) tính bằng giây (s), vận tốc (v) tính bằng cm/s, $(\dot{x}, \ddot{x})$ là đạo hàm bậc 1 và đạo hàm bậc 2 của x, $\mathbf{p}$ là tọa độ của 1 điểm, giá trị ϵ sử dụng để tránh chia cho 0.

1) FPS-aware scaling: Đây là đóng góp kỹ thuật quan trọng nhất của phương pháp. Toàn bộ tham số liên quan đến thời gian (rolling window, lag, EWM span) được scale theo fps thực tế của video nhằm duy trì ý nghĩa vật lý và đảm bảo tính nhất quán giữa các video.

Với một cửa sổ chuẩn w_{ref} được định nghĩa tại $f_{ref} = 30$ fps (tương ứng 1 giây), cửa sổ thực tế w tại tốc độ f được tính:

$$w = \max\left(1, \text{round}\left(w_{ref} \cdot \frac{f}{30}\right)\right) \tag{1}$$

Ví dụ: Nếu video thực tế có $f = 60$, cửa sổ $w_{ref} = 30$ sẽ tự động scale lên 60 frame để vẫn đại diện cho 1 giây.

2) Đặc trưng cho hành vi đơn (single mouse): Hàm `transform_single` tạo ra các nhóm đặc trưng sau:

a) Đặc trưng hình học cơ bản: Bao gồm bình phương khoảng cách giữa tất cả các cặp bộ phận cơ thể d_{ij}^2. Tỷ lệ giãn dài (Elongation E) được tính bằng tỷ lệ bình phương khoảng cách:

$$E = \frac{||\mathbf{p}_{\text{nose}} - \mathbf{p}_{\text{tail}}||^2}{||\mathbf{p}_{\text{ear}_L} - \mathbf{p}_{\text{ear}_R}||^2 + \epsilon} \tag{2}$$

Góc định hướng cơ thể θ_{body} được xác định bởi vector từ đuôi đến mũi.

b) Đặc trưng vận tốc: Vận tốc tức thời v_t (cm/s) được tính dựa trên độ dịch chuyển (displacement) với độ trễ τ đã được scale theo FPS:

$$v_t = \frac{||\mathbf{p}_t - \mathbf{p}_{t-\tau}||}{\tau / f} \tag{3}$$

c) Đặc trưng thời gian đa tầng: Thống kê (Mean, Std, Min, Max, Range) của tọa độ và vận tốc trong các cửa sổ $w \in \{5, 15, 30, 60\}$ frames. Chỉ số hoạt động (Activity Index $\mathcal{A}$) trong cửa sổ w:

$$\mathcal{A}_w = \sqrt{\text{Var}(\Delta x)_w + \text{Var}(\Delta y)_w} \tag{4}$$

d) Đặc trưng độ cong quỹ đạo: Độ cong κ (Curvature) và tốc độ quay (Turn Rate) được tính từ vận tốc và gia tốc, đảm bảo tính bất biến thời gian:

$$\kappa = \frac{|\dot{x}\ddot{y} - \dot{y}\ddot{x}|}{||\mathbf{v}||^3}, \quad \text{TurnRate}_w = \sum_{t}^{t+w} |\Delta\theta| \tag{5}$$

e) Đặc trưng trạng thái: Vận tốc được phân loại thành 4 trạng thái rời rạc (Dừng, Chậm, Trung bình, Nhanh) với các ngưỡng (cm/s) được scale theo FPS. Đặc trưng bao gồm tỷ lệ thời gian ở mỗi trạng thái và số lần chuyển trạng thái trong cửa sổ.

f) Đặc trưng dài hạn: Để nắm bắt ngữ cảnh, hệ thống tính Rolling mean/std trên cửa sổ 120-240 frame, EWM với span 60-120 frame, và khoảng cách tích lũy (Cumulative Distance):

$$D_{cum} = \sum_{i=t-180}^{t} ||\Delta\mathbf{p}_i|| \tag{6}$$

g) Đặc trưng grooming micro: Để nhận diện hành vi chải chuốt, các chỉ số vi mô được tính toán:

$$R_{decouple} = \frac{v_{\text{nose}}}{v_{\text{body}} + \epsilon}, \quad J_{head} = \text{std}_{w=30}(\theta_{\text{head}}) \tag{7}$$

Trong đó $R_{decouple}$ là tỷ lệ tách biệt vận tốc đầu/thân và J_{head} là độ rung (jitter) của hướng đầu.

h) Đặc trưng asymmetry tương lai-quá khứ: Sử dụng Symmetric KL Divergence (D_{sym}) giữa phân phối vận tốc quá khứ (P) và tương lai (F) để phát hiện sự kiện:

$$D_{sym} = \frac{1}{2}\left[\frac{\sigma_P^2}{\sigma_F^2} + \frac{(\mu_P - \mu_F)^2}{\sigma_F^2} + \frac{\sigma_F^2}{\sigma_P^2} + \frac{(\mu_F - \mu_P)^2}{\sigma_P^2} - 2\right] \tag{8}$$

3) Đặc trưng cho tương tác cặp (pair interaction): Hàm `transform_pair` tạo thêm các đặc trưng mô tả mối quan hệ giữa hai chuột:

a) Đặc trưng khoảng cách giữa cá thể: Tập hợp bình phương khoảng cách giữa tất cả các cặp bộ phận của hai chuột (cross-individual distances).

b) Đặc trưng định hướng tương đối: Góc ϕ giữa vector hướng di chuyển của hai chuột, chỉ báo trạng thái đối mặt hay quay lưng:

$$\cos\phi = \frac{\mathbf{v}_A \cdot \mathbf{v}_B}{||\mathbf{v}_A|| \cdot ||\mathbf{v}_B||} \tag{9}$$

c) Tốc độ tiếp cận: Sự thay đổi của bình phương khoảng cách mũi-mũi (d_{nn}^2) theo thời gian:

$$v_{appr} = d_{nn}^2(t) - d_{nn}^2(t - \tau) \tag{10}$$

Giá trị âm chỉ báo chuột đang tiến lại gần nhau.

d) Phân loại khoảng cách: Khoảng cách trung tâm-trung tâm (d_{cc}) được chia thành 4 mức: Rất gần (< 5cm), Gần (5-15cm), Trung bình (15-30cm), Xa (> 30cm).

e) Đặc trưng thời gian tương tác: Thống kê (Mean, Std, Min, Max) của khoảng cách trong các cửa sổ. Chỉ số cường độ tương tác (Intensity Index):

$$I_w = \frac{1}{1 + \text{Var}(d_{cc}^2)_w} \tag{11}$$

f) Đặc trưng động học mũi-mũi: Bao gồm khoảng cách mũi-mũi tức thời, độ thay đổi khoảng cách, và tỷ lệ thời gian ở gần (< 10cm) trong các khoảng lag khác nhau.

g) Velocity alignment: Correlation giữa độ lớn vận tốc của hai chuột ở các offset thời gian khác nhau, chỉ báo hành vi đồng bộ.

h) Đặc trưng leading/chasing: Chỉ số dẫn đầu L_A (chiếu vận tốc lên trục nối tâm) và điểm số rượt đuổi C_{chase}:

$$C_{chase} = \max(0, -v_{appr}) \times L_{target} \tag{12}$$

Công thức này kết hợp tốc độ tiếp cận và vị trí dẫn đầu của đối phương để xác định hành vi rượt đuổi.

D. Ensemble và chiến lược huấn luyện

1) Stratified subset sampling: Kỹ thuật này cho phép mô hình học trên tập dữ liệu lớn với chi phí thấp nhưng vẫn duy trì phân phối lớp nhờ sampling theo phân tầng.

2) Wrapper nâng cao với early stopping tự động: Wrapper bổ sung validation set động, lựa chọn metric thích hợp và điều chỉnh patience theo độ hiếm của lớp.

3) Cấu hình ensemble: Hệ thống sử dụng 5–8 mô hình boosting (LightGBM, XGBoost, CatBoost), kết hợp nhiều mức độ sâu và số cây khác nhau. Các mô hình nâng cao chỉ được dùng khi GPU khả dụng.

4) Xử lý lỗi và fallback: Pipeline được trang bị cơ chế fallback thông minh khi gặp lỗi GPU hoặc lỗi huấn luyện, nhằm đảm bảo quá trình train không bị gián đoạn.

E. Suy luận và hậu xử lý

1) Thresholding: Dự đoán xác suất được làm mượt theo thời gian, áp dụng ngưỡng riêng cho từng hành vi và chuyển đổi thành các chuỗi sự kiện liên tục. Các sự kiện ngắn bất thường được loại bỏ.

2) Robustness handling: Hệ thống loại bỏ sự kiện không hợp lệ, xử lý chồng lấn và bổ sung dự đoán cho các video không có kết quả, đảm bảo submission luôn tuân thủ định dạng yêu cầu.

F. Khả năng tái lập và quản lý SEED

Toàn bộ pipeline được kiểm soát bởi SEED cố định để đảm bảo tính tái lập. Mọi thành phần có yếu tố ngẫu nhiên (StratifiedShuffleSplit, LightGBM, XGBoost, CatBoost) đều được truyền cùng SEED này, đảm bảo kết quả hoàn toàn giống nhau giữa các lần chạy.

IV. Thiết lập thí nghiệm và Kết quả

A. Môi trường thực nghiệm

Phần cứng: GPU P100

Phần mềm:

- Python 3.11.13
- pandas 2.x, numpy 1.x
- scikit-learn
- LightGBM, XGBoost, CatBoost
- Polars

Cấu hình:

- USE_GPU được tự động phát hiện dựa vào môi trường Kaggle
- Stratified sampling với n_samples=1,500,000 frame cho baseline models
- Validation size=0.10, validation cap ratio=0.25
- SEED=1234 cố định cho mọi thành phần ngẫu nhiên

B. Dữ liệu

Training set:

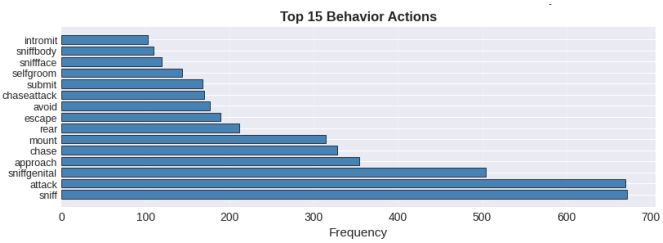
- Số lượng video: 8789
- Số lượng video mất nhãn: 7941 (90.4%)
- Số lượng lab: 21
- Số chuột trung bình trong mỗi video: 2.90
- Độ dài trung bình mỗi video: 134.6s

Test set:

- Số lượng video: 1
- Số lượng lab: 1

Phân phối hành vi:

- Tổng số hành vi: 37
- Hành vi xuất hiện phổ biến nhất trong một video: sniff
- Số lượng hành vi trung bình trong mỗi video: 5.8
- Số lượng hành vi nhiều nhất trong một video: 76



Hình 1. Thống Kê Top 15 Số Lần Xuất Hiện Nhiều Nhất Trong 1 Video Của Các Hành Vi

C. Kết quả

Điểm số trên leaderboard

- **Public leaderboard score**: 0.449
- **Xếp hạng**: 427 / 1396

D. Thời gian thực thi

- **Training time**: 6h 21m 20s
- **Inference time** (trên public sample test): 10s
- **Total pipeline**: 6h 21m 30s

Bảng I
Thống kê phân phối số lượng và thời lượng các hành vi trong tập dữ liệu

Hành vi	Số sự kiện	Số frames	Tỷ lệ (%)
sniff	37,837	2,150,590	45.25
sniffgenital	7,862	703,931	9.40
attack	7,462	518,870	8.92
rear	4,408	230,889	5.27
sniffbody	3,518	103,274	4.21
approach	3,270	85,968	3.91
sniffface	2,811	71,527	3.36
mount	2,747	284,931	3.29
escape	2,071	90,304	2.48
reciprocalsniff	1,492	40,682	1.78
defend	1,409	89,982	1.69
selfgroom	1,356	81,761	1.62
dig	1,127	78,770	1.35
climb	1,010	60,334	1.21
chase	826	26,704	0.99
intromit	691	347,964	0.83
avoid	530	25,674	0.63
dominancemount	410	17,556	0.49
dominance	329	38,154	0.39
huddle	299	24,148	0.36
disengage	279	12,021	0.33
follow	233	40,144	0.28
rest	233	87,573	0.28
attemptmount	223	5,938	0.27
shepherd	201	29,451	0.24
flinch	184	1,677	0.22
chaseattack	124	5,413	0.15
tussle	122	5,056	0.15
exploreobject	105	3,678	0.13
freeze	105	31,660	0.13
submit	86	8,478	0.10
run	76	1,732	0.09
dominancegroom	53	4,724	0.06
genitalgroom	50	6,270	0.06
allogroom	45	7,580	0.05
biteobject	33	2,326	0.04
ejaculate	3	1,611	0.00

V. Thảo luận

A. Phân tích kết quả

1) Hiệu quả của FPS-aware features: Cơ chế *FPS-aware feature scaling* là đóng góp quan trọng nhất của nghiên cứu. Khi không được chuẩn hóa theo tốc độ khung hình (fps), các đặc trưng thời gian như *rolling mean*, vận tốc, hay độ cong (curvature) sẽ mất tính nhất quán giữa các video. Ví dụ, một cửa sổ trượt 30 frame mang ý nghĩa hoàn toàn khác ở video 30 fps (tương đương 1 giây) so với video 60 fps (tương đương 0.5 giây). Việc scale tự động đảm bảo mọi đặc trưng thời gian đều đại diện cho cùng một khoảng thời gian vật lý, giúp mô hình tổng quát hóa tốt hơn.

Điều này mang tính quyết định trong cuộc thi MABe 2025 do dữ liệu được thu thập từ nhiều phòng thí nghiệm với thiết bị ghi hình không đồng nhất. Một số video đạt 30 fps, số khác 60 fps, thậm chí có video có fps không chuẩn. Nếu không xử lý, mô hình sẽ học các mẫu (patterns) phụ thuộc vào thông số kỹ thuật của camera thay vì hành vi thực tế của chuột.

2) Vai trò của Ensemble: Việc kết hợp (Ensemble) nhiều mô hình boosting giúp tăng tính ổn định và giảm phương sai (variance):

Bảng II
THỐNG KÊ HYPERPARAMETERS CHÍNH CỦA CÁC MÔ HÌNH ENSEMBLE

1. Mô hình Cơ bản (LGBM, XGBoost, CatBoost)	
Model	LGBM (lgbm_225)
Hyperparameters	$N_{\text{est}} = 225$, $LR = 0.07$, $min\_child\_samples = 40$
Sampling	$N_{\text{samples}} = 1.5M$
Optimization	Subsample / Colsample = 0.8
Model	LGBM (lgbm_150)
Hyperparameters	$N_{\text{est}} = 150$, $LR = 0.1$, $max\_depth = 8$, $reg_{\alpha/\lambda} = 0.1$
Sampling	$N_{\text{samples}} \approx 1.2M$
Optimization	Subsample / Colsample = $0.7/0.9$
Model	LGBM (lgbm_100)
Hyperparameters	$N_{\text{est}} = 100$, $LR = 0.05$, $max\_depth = 10$, $num\_leaves = 127$
Sampling	$N_{\text{samples}} \approx 0.9M$
Optimization	Subsample = 0.75
Model	XGBoost (xgb_180)
Hyperparameters	$N_{\text{est}} = 180$, $LR = 0.08$, $max\_depth = 6$
Sampling	$N_{\text{samples}} \approx 1.25M$
Optimization	$\gamma = 1.0$ (GPU) / 0.0 (CPU)
Model	CatBoost (cat_120)
Hyperparameters	$Iterations = 120$, $LR = 0.1$, $depth = 6$
Sampling	$N_{\text{samples}} = 1.5M$
Optimization	GPU / CPU
2. Mô hình Tối ưu GPU với Early Stopping	
Model	XGBoost (xgb1)
Hyperparameters	$N_{\text{est}} = 2000$, $LR = 0.05$, $max\_leaves = 255$
Sampling	$N_{\text{samples}} = 0.75M$, ES (PRAUC)
Optimization	Lossguide, Subsample = 0.9
Model	XGBoost (xgb2)
Hyperparameters	$N_{\text{est}} = 1400$, $LR = 0.06$, $max\_depth = 7$
Sampling	$N_{\text{samples}} = 1.0M$, ES (PRAUC)
Optimization	Subsample / Colsample = $0.7/0.8$
Model	CatBoost (cat_bay)
Hyperparameters	$Iterations = 4000$, $LR = 0.03$, $depth = 8$
Sampling	$N_{\text{samples}} = 0.75M$, ES (PRAUC)
Optimization	Balanced class weights, Bayesian bootstrap

Ghi chú: N_{est}: số cây; LR: learning rate; $N_{samples}$: kích thước tập con phân tầng; ES: Early Stopping.

- **LightGBM** (với `num_leaves` nhỏ): Học các pattern tổng quát.
- **LightGBM** (với `num_leaves` lớn): Học các pattern phức tạp.
- **XGBoost GPU**: Khai phá sâu các đặc trưng.
- **CatBoost**: Đa dạng hóa dữ liệu thông qua Bayesian bootstrap.

Chiến lược này giúp cân bằng giữa *underfitting* và *overfitting*, đặc biệt hiệu quả đối với các lớp hành vi hiếm gặp có ít dữ liệu huấn luyện.

3) Stratified Sampling: Kỹ thuật *Stratified subset sampling* giúp giảm đáng kể thời gian huấn luyện mà không làm suy giảm hiệu suất mô hình. Phương pháp này khả thi vì:

- Tập con vẫn bảo toàn phân phối lớp của tập gốc.
- Số lượng 1–1.5 triệu frame vẫn đủ lớn để mô hình học được các pattern chính.
- Việc Ensemble nhiều mô hình giúp bù đắp lượng phương sai gia tăng do sử dụng tập con.

Kỹ thuật này đặc biệt quan trọng khi cần thực hiện nhiều thử nghiệm cấu hình trong giới hạn thời gian của cuộc thi.

B. Ưu điểm của phương pháp

Phương pháp đề xuất thể hiện các ưu điểm chính sau:

1) **Tính thích ứng:** FPS-aware features đảm bảo hoạt động tốt trên dữ liệu đa nguồn, đa thiết bị.
2) **Hiệu suất:** Stratified sampling giảm thiểu chi phí tính toán huấn luyện.
3) **Xử lý mất cân bằng:** Kết hợp hiệu quả class weights và metric tùy chỉnh (AUPRC).
4) **Tự động hóa:** Các tham số như metric, patience được điều chỉnh tự động dựa trên dữ liệu.
5) **Tính ổn định:** Quy trình Robustify và quản lý SEED chặt chẽ đảm bảo kết quả hợp lệ và có thể tái lập.

C. Hạn chế và Cải thiện

1) Hạn chế: Mặc dù hệ thống đặc trưng được thiết kế chi tiết, phương pháp vẫn dựa chủ yếu vào *feature engineering* thủ công, đòi hỏi kiến thức miền và có thể bỏ sót các pattern tiềm ẩn. Ngoài ra, các mô hình boosting về bản chất xử lý từng frame độc lập (sau khi đã trích xuất đặc trưng cửa sổ), do đó khả năng nắm bắt các phụ thuộc thời gian dài hạn (long-range temporal patterns) còn hạn chế so với LSTM hay Transformer.

Về mặt triển khai, việc tạo ra hàng trăm đặc trưng thủ công tiêu tốn lượng lớn bộ nhớ (RAM). Quy trình xử lý nhiễu và ngoại lai hiện tại còn đơn giản, chưa áp dụng các bộ lọc phức tạp để xử lý triệt để nhiễu tracking.

2) Hướng cải thiện: Các hướng nghiên cứu tiềm năng để nâng cao hiệu quả bao gồm:

- **Mô hình lai (Hybrid):** Sử dụng đầu ra của boosting làm đầu vào cho 1D-CNN hoặc LSTM để học sâu hơn các phụ thuộc thời gian.
- **Meta-learning:** Thay vì cố định ngưỡng, có thể huấn luyện một meta-model để dự đoán ngưỡng tối ưu dựa trên đặc tính video (fps, phân bố điểm số).
- **Tối ưu hóa đặc trưng:** Áp dụng phân tích tầm quan trọng (Feature Importance) hoặc Neural Architecture Search để loại bỏ các đặc trưng thừa.
- **Xử lý tín hiệu:** Tích hợp Kalman filter hoặc Savitzky-Golay filter để làm mượt tọa độ đầu vào.
- **Augmentation:** Tăng cường dữ liệu bằng các phép biến đổi hình học (xoay, lật) để tăng khả năng tổng quát hóa.

VI. Kết luận

A. Tóm tắt đóng góp

Bài tập lớn này trình bày một hệ thống hoàn chỉnh để phát hiện hành vi chuột tự động từ dữ liệu tracking, với các đóng góp kỹ thuật chính:

1) FPS-aware feature engineering đảm bảo tính nhất quán của đặc trưng thời gian trên dữ liệu đa dạng fps.
2) Stratified subset sampling giúp huấn luyện hiệu quả trên tập dữ liệu lớn.
3) Ensemble boosting models với cấu hình đa dạng và early stopping tự động.
4) Robustness mechanisms đảm bảo submission luôn hợp lệ.

Phương pháp đạt điểm 0.449 trên public leaderboard của cuộc thi MABe 2025, cho thấy hiệu quả của các kỹ thuật đề xuất.

B. Bài học kinh nghiệm

Kiến thức miền rất quan trọng trong việc hiểu rõ sinh học hành vi chuột giúp thiết kế đặc trưng phù hợp. Ví dụ, biết rằng grooming thường có head-body decoupling cao, hoặc chase có correlation giữa approach và leading direction. Lọc bỏ dữ liệu không phù hợp (chuột ngủ) cải thiện hiệu suất đáng kể. Đầu tư thời gian hiểu dữ liệu luôn đáng giá hơn là mù quáng thử nhiều mô hình phức tạp. Quản lý SEED chặt chẽ từ đầu giúp debug dễ dàng và so sánh các thử nghiệm công bằng. Mỗi thay đổi nhỏ đều cho kết quả ổn định giữa các lần chạy. Càng nhiều thành phần tự động (metric selection, patience, class weights) càng giảm manual tuning và human error. Các mô hình boosting truyền thống với feature engineering tốt vẫn cạnh tranh được với deep learning, đặc biệt khi dữ liệu không quá lớn và thời gian hạn chế.

C. Hướng phát triển

Những hướng phát triển tiềm năng cho tương lai: Tối ưu pipeline để chạy real-time trên video stream, phục vụ ứng dụng giám sát hành vi liên tục trong phòng thí nghiệm. Mở rộng phương pháp sang các loài động vật khác với minimal retraining. Phát triển visualization tools để giải thích các dự đoán của mô hình, giúp các nhà nghiên cứu sinh học hiểu và tin tưởng hệ thống hơn. Xây dựng cơ chế để mô hình chủ động đề xuất những frame cần được label thêm, giảm chi phí annotation. Kết hợp tracking data với audio (ultrasonic vocalization của chuột) hoặc các sensor khác (temperature, light) để cải thiện độ chính xác.

Acknowledgment

Cảm ơn Kaggle và MABe Challenge organizers đã cung cấp dataset và platform. Cảm ơn cộng đồng Kaggle đã chia sẻ insights và discussions.

Tài liệu

[1] Kaggle Competition, "MABe: Mouse Behavior Detection Challenge 2025,"2025. [Online]. Available: https://www.kaggle.com/competitions/MABe-mouse-behavior-detection

[2] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, "DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning,"*Nat. Neurosci.*, vol. 21, no. 9, pp. 1281–1289, Sep. 2018.

[3] J. J. Sun, A. Kennedy, E. Zhan, D. J. Anderson, Y. Yue, and P. Perona, "Task programming: Learning data efficient behavior representations,"in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2876–2885.

[4] K. Luxem, F. Fuhrmann, J. Kürsch, S. Remy, and P. Bauer, "Identifying behavioral structure from deep variational embeddings of animal motion,"*Commun. Biol.*, vol. 5, no. 1, art. 1267, Nov. 2022.

[5] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree,"in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 3146–3154.

[6] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system,"in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Aug. 2016, pp. 785–794.

[7] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features,"in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 31, 2018, pp. 6638–6648.

[8] A. I. Hsu and E. A. Yttri, "B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors,"*Nat. Commun.*, vol. 12, no. 1, art. 5188, Aug. 2021.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique,"*J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[10] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,"*PLOS ONE*, vol. 10, no. 3, art. e0118432, Mar. 2015.

[11] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python,"*J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

Phụ lục A
Thuật ngữ chuyên ngành

Phụ lục này cung cấp định nghĩa ngắn gọn và chuẩn hóa các thuật ngữ kỹ thuật được sử dụng xuyên suốt tài liệu, nhằm đảm bảo tính rõ ràng và nhất quán.

A. FPS (Frames Per Second)

Số lượng khung hình được ghi nhận trong một giây của video. FPS ảnh hưởng trực tiếp đến ý nghĩa thời gian của các đặc trưng chuỗi và cần được chuẩn hóa để đảm bảo tính nhất quán giữa các video.

B. FPS-aware Feature Engineering

Kỹ thuật trích xuất đặc trưng trong đó mọi tham số liên quan đến thời gian (rolling window, lag, span) đều được scale theo FPS thực tế của từng video, giúp các đặc trưng duy trì cùng ý nghĩa vật lý.

C. Frame-level Prediction

Dự đoán hành vi tại từng khung hình riêng lẻ thay vì dự đoán trên toàn bộ đoạn video. Cách tiếp cận này cho phép phát hiện chính xác thời điểm bắt đầu và kết thúc của hành vi.

D. Event-level Representation

Biểu diễn hành vi dưới dạng các đoạn liên tục (start frame, end frame) được tạo ra từ các dự đoán frame-level sau khi áp dụng hậu xử lý.

E. Stratified Subset Sampling

Chiến lược lấy mẫu trong đó một tập con dữ liệu được chọn sao cho vẫn bảo toàn phân phối lớp của tập dữ liệu gốc, giúp giảm chi phí huấn luyện mà không làm sai lệch mô hình.

F. Class Imbalance

Hiện tượng phân phối không đồng đều giữa các lớp, phổ biến trong bài toán nhận dạng hành vi khi một số hành vi hiếm xảy ra hơn đáng kể so với các hành vi khác.

G. Temporal Smoothing

Quá trình làm mượt chuỗi dự đoán theo thời gian (ví dụ: rolling mean) nhằm giảm nhiễu và hiện tượng nhấp nháy (jitter) trong dự đoán frame-level.

H. Ensemble Learning

Phương pháp kết hợp nhiều mô hình học máy độc lập để tạo ra dự đoán cuối cùng, giúp tăng độ ổn định và khả năng tổng quát hóa của hệ thống.

I. Gradient Boosting Decision Trees (GBDT)

Họ mô hình học máy dựa trên việc kết hợp nhiều cây quyết định yếu theo hướng giảm gradient của hàm mất mát, bao gồm LightGBM, XGBoost và CatBoost.

J. Early Stopping

Cơ chế dừng huấn luyện khi hiệu suất trên tập validation không còn cải thiện sau một số vòng lặp nhất định, nhằm tránh overfitting.

K. ROC và AUPRC

ROC (Receiver Operating Characteristic) và AUPRC (Area Under Precision-Recall Curve) là các thước đo đánh giá mô hình phân loại. Trong bối cảnh dữ liệu mất cân bằng, AUPRC thường mang lại thông tin đánh giá hữu ích hơn.

L. Reproducibility

Khả năng tái lập kết quả thực nghiệm khi chạy lại pipeline với cùng cấu hình và SEED ngẫu nhiên, là yêu cầu quan trọng trong nghiên cứu khoa học.

M. Lag

Lag trong chuỗi thời gian là độ trễ giữa một sự kiện và hiệu ứng của nó lên các quan sát, thường được đo bằng số khung hình hoặc đơn vị thời gian. Các giá trị trễ được sử dụng để tạo đặc trưng vận tốc (khác biệt vị trí giữa thời điểm hiện tại và quá khứ) và nắm bắt mối quan hệ phụ thuộc theo thời gian trong mô hình hóa.

N. Exponential Weighted Mean

Trung bình trọng số lũy thừa, hay Exponential Weighted Moving Average (EWMA), là kỹ thuật làm mượt chuỗi trong đó các quan sát quá khứ được gán trọng số giảm dần theo hàm mũ. Khác với trung bình trượt đơn giản (các quan sát có trọng số bằng nhau), phương pháp này nhấn mạnh dữ liệu gần hiện tại hơn, giúp phản ánh nhanh các thay đổi trong hành vi.

O. Pattern

Trong học máy, *pattern* (mẫu) là một quy luật hoặc cấu trúc lặp lại trong dữ liệu giúp làm nổi bật ý nghĩa tiềm ẩn. Các mẫu cho phép phân loại dữ liệu ngẫu nhiên để đưa ra quyết định có cơ sở.

P. Manual Hyperparameter Tuning

Manual hyperparameter tuning (điều chỉnh siêu tham số thủ công) là quá trình thử nghiệm nhiều bộ siêu tham số bằng tay; mỗi phép thử được thực hiện bởi người dùng và cần công cụ theo dõi thí nghiệm để ghi lại các kết quả. Phương pháp này mang lại quyền kiểm soát cao hơn nhưng đòi hỏi nhiều thời gian và công sức.

Q. Minimal Retraining

Trong các phương pháp như meta learning hoặc few-shot learning, mô hình được thiết kế để thích ứng nhanh với nhiệm vụ mới chỉ với một lượng nhỏ dữ liệu huấn luyện bổ sung. Meta learning giúp mô hình học cách học, cho phép nó cải thiện khả năng thích ứng với các nhiệm vụ mới với rất ít bước huấn luyện bổ sung. Khả năng này được gọi là “minimal retraining” và rất hữu ích khi mở rộng mô hình sang loài khác hoặc điều kiện khác mà không cần đào tạo lại toàn bộ.