Dự báo Trạng thái Hệ thống

Dự đoán Trạng thái Người dùng qua sự biến động Tài Nguyên

31 tháng 1, 2026

Nhóm Dự báo Hệ thống

Contents

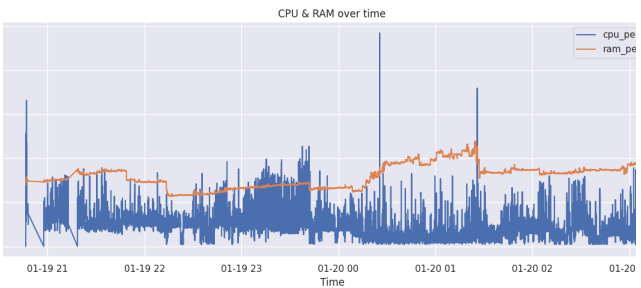
1 Tổng quan Dự án

Tầm nhìn & Sự thay đổi (Pivot)

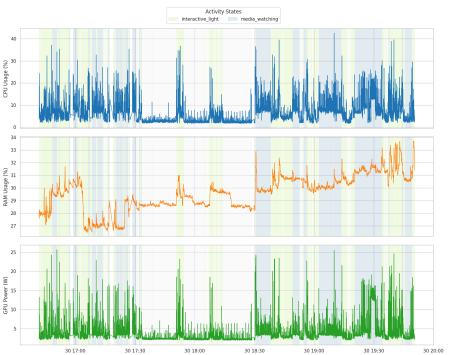
- **Mục tiêu:** Phân loại trạng thái người dùng (**Nghỉ**, **Tương tác nhẹ**, **Giải trí/Media**) chỉ sử dụng các chỉ số phần cứng không xâm lấn.
- **Sự thay đổi:** Ban đầu thử dự đoán mức sử dụng CPU thô.
  - **Thất bại vì:** 99.9% mức sử dụng là 5-15% (Bẫy "đoán mò 10%").
  - **Kết quả:** Chúng tôi không thể dự đoán **khi nào** người dùng bắt đầu một tác vụ, nhưng chúng tôi có thể xác định **họ đang làm gì** thông qua phản ứng của phần cứng.

Tầm nhìn & Sự thay đổi (ii)

- **Bộ dữ liệu:**
  - Người dùng A (Linux): 42,000 dòng | Người dùng B: 6,000 dòng | Người dùng C (Windows): 20,000 dòng.
  - **Đã loại bỏ:** Tất cả 68,000 dòng vì thiếu chỉ số GPU, thứ đã được chứng minh là thiết yếu.



Bộ dữ liệu cũ (CPU/RAM)



Bộ dữ liệu mới (bao gồm GPU)

Tầm nhìn & Sự thay đổi (ii) (ii)

Bộ dữ liệu mới (Mạng và Đĩa)

Kiến trúc Thu thập Dữ liệu

- **Môi trường:** Arch Linux + Niri Window Manager (có khả năng script sâu).
- **Nguồn:**
  - `psutil`: CPU, RAM, Đĩa, Mạng.
  - `intel_gpu_top`: "Bằng chứng thép" để phát hiện trạng thái.
  - `libinput`: Được sử dụng để tính toán WPM/Trạng thái nghỉ thực tế (ground-truth) cho việc huấn luyện.
- **Rào cản Kỹ thuật:** Để lấy dữ liệu GPU mà không cần `sudo`, chúng tôi đã áp dụng:

```
sudo setcap cap_perfmon+ep /usr/bin/intel_gpu_top
```

Kiến trúc Thu thập Dữ liệu (ii)

- **Xác thực:** Thông báo cảnh báo thời gian thực (`watch_metrics.nu`).

```
# watch_metrics.nu
def main [...params: string] {
  loop {
    open comprehensive_activity_log.csv | last | select ...$params
    | notify-send -t 1000 $"($in)"
    sleep 1sec
  }
}
```

2 Kỹ thuật & Gán nhãn

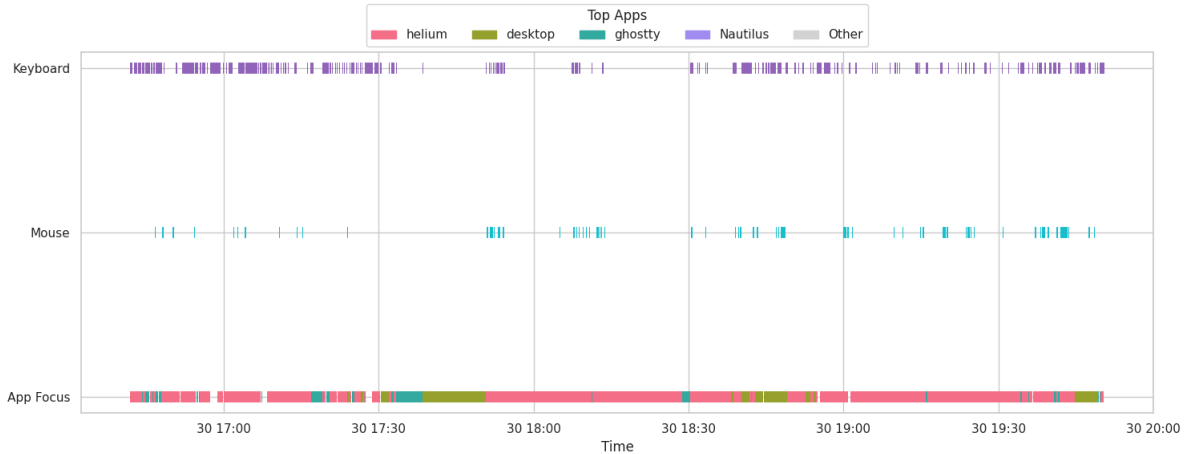
Chiến lược Gán nhãn: Thủ công-nhưng-Chính xác

1. **Gán thẻ thủ công:** Phím tắt Niri được gán cho `current_state.txt`.
2. **Heuristic cho trạng thái nghỉ:** Nhãn sau xử lý.

```
# live_inference.py (Heuristic Override)
if prediction == 'interactive_light':
    # If GPU is in deep sleep (RC6) and CPU is very low,
    # or if the GPU engines are essentially dead (< 1%):
    if (rc6_mean_5s > 99.0 and cpu_mean_5s < 5.0) or (max_gpu_raw < 1.0):
        prediction = 'Idle'
```

Chiến lược Gán nhãn: Thủ công-nhưng-Chính xác (ii)

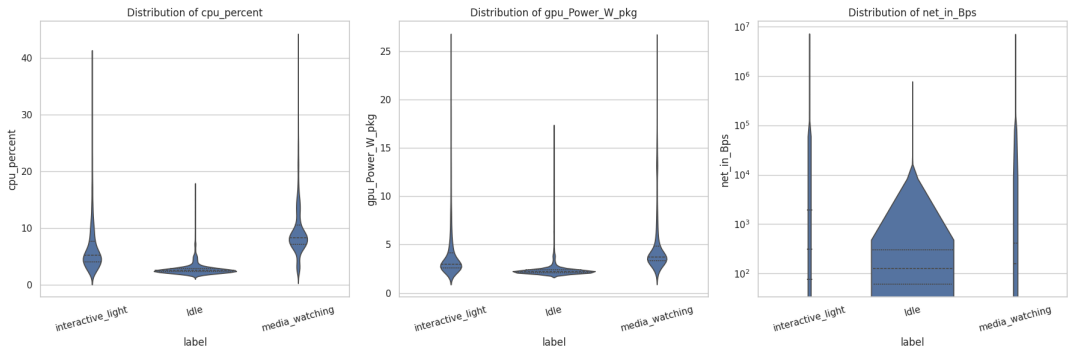
3. **Tập huấn luyện cuối cùng:** 11,000 dòng dữ liệu được gán nhãn chất lượng cao.



Kỹ thuật Đặc trưng: Cửa sổ trượt (Rolling Windows)

Các chỉ số thô quá nhiễu. Chúng tôi đã thiết kế các đặc trưng qua hai cửa sổ:

- **Ngắn (5 giây):** Nắm bắt các đợt tăng đột ngột (ví dụ: tải trang).
- **Dài (30 giây):** Nắm bắt các mẫu duy trì (ví dụ: luồng video 24fps).
- **Thông tin từ "Max GPU":** Phát hiện ra rằng `max_gpu` (mức tải cao nhất trên bất kỳ engine đơn lẻ nào) là yếu tố phân biệt số 1 giữa "Nghỉ" và "Đọc tĩnh".



3 Mô hình Học máy

Sự phát triển của Mô hình & Bẫy rò rỉ

- **Thuật toán:** Random Forest Classifier (xử lý các gai phi tuyến tính).
- **Mô hình "Rò rỉ" (v1):** Đạt độ chính xác 87%, nhưng chúng tôi nhận ra nó đang đọc `idle_time_sec`. Vì chúng tôi không thể sử dụng log đầu vào trong môi trường thực tế (production), chúng tôi đã huấn luyện lại.
- **Mô hình Chỉ dùng Hệ thống (v2):** Độ chính xác giảm xuống 82%, nhưng hoạt động tốt hơn đáng kể trong các kịch bản suy luận "trực tiếp" (live) nhờ dựa hoàn toàn vào các mẫu tài nguyên.

Top 5 Đặc trưng (v2):

1. `gpu_RC6_pct_mean_5s`
2. `max_gpu_mean_5s`
3. `cpu_percent_mean_30s`
4. `cpu_percent_mean_5s`
5. `gpu_RC6_pct_std_5s`

Hiệu suất v1 (Rò rỉ):

Lớp	Độ chính xác (Precision)	Độ nhạy (Recall)	Điểm F1
Nghỉ	0.70	1.00	0.82
Tương tác (Nhẹ)	0.83	0.97	0.89
Xem Media	0.97	0.77	0.86

Công cụ Suy luận (Logic Lai)

Để giải quyết sự chồng chéo giữa "Nghỉ vs. Đọc nhẹ" (nơi các chỉ số gần như giống hệt nhau), chúng tôi đã triển khai một **Cơ chế Ghi đè Nhận thức Phần cứng**:

```
if prediction == 'interactive_light':
    # If GPU is in deep sleep (RC6) and CPU is very low,
    # or if the GPU engines are essentially dead (< 1%):
    if (rc6_mean_5s > 99.0 and cpu_mean_5s < 5.0) or (max_gpu_raw < 1.0):
        prediction = 'Idle'
```

Kết quả: Phát hiện trạng thái Nghỉ với độ chính xác cao mà không cần hook chuột/bàn phím.

4 Kết quả & Kết luận

Hiệu suất Mô hình (v2 Chỉ dùng Hệ thống)

Lớp	Độ chính xác	Độ nhạy	Điểm F1
Nghỉ	0.53	0.59	0.56
Tương tác (Nhẹ)	0.76	0.94	0.84
Xem Media	0.97	0.76	0.85

Lưu ý: Độ chính xác khi Xem Media là đặc biệt cao (97%) do tín hiệu độc nhất của Video Command Streamer (VCS) engine.

Kết quả & Tương lai

Quyền riêng tư: Việc phát hiện trạng thái đạt được mà không bao giờ ghi lại phím bấm hay tiêu đề cửa sổ nhạy cảm trong môi trường thực tế.

Khả năng phản hồi: Tốc độ lấy mẫu 1 giây với chi phí tài nguyên tối thiểu.

Công việc Tương lai:

Độ trễ (Hysteresis): Triển khai độ trễ chuyển đổi trạng thái để ngăn chặn hiện tượng “nhấp nháy”.

Bao gồm các hoạt động tương tác nặng (ví dụ: biên dịch mã, chạy các tác vụ ML nặng, chơi game) và tải xuống nền (ví dụ: torrenting). Vì tôi thường tải xuống hàng chục gigabyte dữ liệu media, tôi tin rằng danh mục này đáng được đưa vào.