BÁO CÁO KỸ THUẬT: KHUNG LÀM VIỆC TOÀN CỤC (GLOBAL FRAMEWORK) CHO DỰ BÁO PHỤ TẢI VÀ PHÁT HIỆN BẤT THƯỜNG TRÊN QUY MÔ LỚN

Tác giả: Trương Xuân Huy

Ngày thực hiện: 29/01/2026

Mã nguồn tham chiếu: `XGBoost_Global_Final_Fixed.ipynb , Demo_XGBoost_2.ipynb`

TÓM TẮT (ABSTRACT)

Nghiên cứu này đề xuất một giải pháp toàn diện để giải quyết bài toán dự báo tiêu thụ năng lượng và phát hiện bất thường cho mạng lưới hơn 1.600 tòa nhà (Building Data Genome Project 2). Đối mặt với thách thức về sự đa dạng quy mô tải và giới hạn tài nguyên tính toán, chúng tôi phát triển một **Mô hình Toàn cục (Global Model)** duy nhất dựa trên thuật toán **Extreme Gradient Boosting (XGBoost)** được tăng tốc bởi GPU. Hệ thống tích hợp quy trình xử lý dữ liệu theo lô (Chunk Processing), kỹ thuật nén bộ nhớ động, và tối ưu hóa siêu tham số tự động bằng Optuna. Đặc biệt, để phát hiện bất thường trong môi trường không nhãn, chúng tôi đề xuất cơ chế **Ngưỡng thích nghi ngữ cảnh (Contextual Adaptive Thresholding)** dựa trên thống kê bền vững (Robust Statistics/IQR). Kết quả thực nghiệm mô phỏng trên 200 tòa nhà cho thấy mô hình đạt độ chính xác dự báo cao (RMSE ~14.7) và hiệu suất phát hiện bất thường xuất sắc với điểm F1-Score đạt 0.957.

1. GIỚI THIỆU (INTRODUCTION)

1.1. Bối cảnh và Thách thức

Trong kỷ nguyên lưới điện thông minh, dữ liệu từ các đồng hồ đo tiên tiến (AMI) cung cấp cơ hội lớn để tối ưu hóa năng lượng. Tuy nhiên, việc xử lý dữ liệu từ hàng nghìn tòa nhà đặt ra hai thách thức lớn:

1. **Tính mở rộng (Scalability):** Các phương pháp truyền thống thường huấn luyện một mô hình riêng biệt (Local Model) cho mỗi tòa nhà. Với $N = 1600$ tòa nhà, việc quản lý N mô hình là gánh nặng lớn về tài nguyên và bảo trì [1].
2. **Tính** bất định (Heterogeneity): Các tòa nhà có quy mô tiêu thụ khác nhau (từ vài kW đến hàng MW) và hành vi tiêu thụ đa dạng, gây khó khăn cho việc huấn luyện chung [3].

1.2. Đóng góp của Nghiên cứu

Chúng tôi đề xuất chuyển dịch từ cách tiếp cận cục bộ sang cách tiếp cận toàn cục (Global Approach), trong đó một mô hình duy nhất học các đặc trưng chung của tất cả tòa nhà. Các

đóng góp chính bao gồm:

- Xây dựng quy trình xử lý dữ liệu (Data Pipeline) tối ưu bộ nhớ cho Big Data.
- Ứng dụng XGBoost với hàm mất mát `reg:squarederror` và tăng tốc phần cứng (GPU Hist).
- Phát triển thuật toán phát hiện bất thường dựa trên số dư (Residual-based) sử dụng ngưỡng động IQR, khắc phục nhược điểm của các phương pháp thống kê truyền thống (như Z-score) vốn nhạy cảm với nhiễu.

2. PHƯƠNG PHÁP LUẬN (METHODOLOGY)

2.1. Tối ưu hóa Dữ liệu và Quản lý Bộ nhớ (Data Pipeline Optimization)

Dữ liệu đầu vào là chuỗi thời gian của 1.636 tòa nhà với tần suất lấy mẫu 1 giờ. Để xử lý khối lượng dữ liệu này trên tài nguyên giới hạn (Google Colab), chúng tôi áp dụng hai kỹ thuật:

a. Xử lý theo lô (Chunk Processing): Thay vì tải toàn bộ dữ liệu (Wide Format) vào RAM, chúng tôi chia nhỏ dữ liệu thành các nhóm (chunks) gồm k tòa nhà (ví dụ: $k = 50$), thực hiện chuyển đổi sang dạng dọc (Long Format) và nối (concat) tuần tự.

b. Ép kiểu dữ liệu động (Dynamic Downcasting): Chúng tôi giảm mức tiêu thụ bộ nhớ khoảng 60-70% bằng cách kiểm tra phạm vi giá trị của từng cột số và ép kiểu về dạng nhỏ nhất có thể (ví dụ: `float64` $\rightarrow$ `float32`, `int64` $\rightarrow$ `int16`) mà không làm mất thông tin [Code: `reduce_mem_usage`].

Công thức xác định kiểu dữ liệu $T(x)$ cho đặc trưng x:

$$T(x) = \begin{cases} \text{int8} & \text{if } \min(x) \geq -128 \wedge \max(x) \leq 127 \\ \text{float32} & \text{if } x \in \mathbb{R} \end{cases}$$

2.2. Kỹ thuật Đặc trưng (Feature Engineering)

Mô hình XGBoost không tự động hiểu tính tuần tự của chuỗi thời gian như RNN/LSTM. Do đó, bước trích xuất đặc trưng là tối quan trọng [5].

a. Mã hóa **Thời gian Chu kỳ (Cyclical Time Encoding):** Thời gian (giờ trong ngày, ngày trong tuần) có tính chu kỳ. Để mô hình hiểu 23:00 gần với 00:00, chúng tôi sử dụng phép biến đổi lượng giác:

$$x_{sin} = \sin\left(\frac{2\pi t}{T}\right), \quad x_{cos} = \cos\left(\frac{2\pi t}{T}\right)$$

Trong đó $T = 24$ (giờ) hoặc $T = 7$ (ngày).

b. Đặc trưng Độ trễ và Tự tương quan (Lag Features): Để nắm bắt sự phụ thuộc vào quá khứ, chúng tôi tạo các biến trễ cho mỗi tòa nhà b:

$$L_k(t, b) = y_{t-k,b} \quad \text{với } k \in \{1, 24, 168\}$$

Các giá trị k tương ứng với giờ trước, ngày hôm trước, và tuần trước, giúp mô hình học được tính mùa vụ (seasonality).

c. Đặc trưng Vận tốc (Velocity/Derivative): Để phát hiện sự thay đổi tải đột ngột, chúng tôi tính đạo hàm bậc nhất rời rạc:

$$\Delta y_t = y_t - y_{t-1}$$

d. Chuẩn hóa Logarit (Log-Transformation): Do biên độ tải giữa các tòa nhà chênh lệch lớn, chúng tôi áp dụng biến đổi `log1p` cho biến mục tiêu để ổn định phương sai:

$$y' = \ln(1 + y)$$

Điều này biến bài toán về cùng một không gian sai số tương đối, giúp mô hình toàn cục hội tụ tốt hơn.

2.3. Mô hình Dự báo: XGBoost (Extreme Gradient Boosting)

Chúng tôi sử dụng XGBoost, một thuật toán dựa trên cây quyết định (Gradient Boosted Decision Trees), nổi tiếng với hiệu suất cao và khả năng xử lý dữ liệu bảng [2].

Hàm mục tiêu (Objective Function) tại bước lặp t:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

Trong đó:

- l: Hàm mất mát (Squared Error).
- $\Omega(f)$: Hàm điều chuẩn (Regularization) kiểm soát độ phức tạp của cây, bao gồm trọng số L1 (`reg_alpha`) và L2 (`reg_lambda`).
- **Cấu hình phần cứng:** Sử dụng tham số `tree_method='hist'` và `device='cuda'` để tận dụng GPU Tesla T4, giúp tăng tốc độ huấn luyện gấp 10-20 lần so với CPU.

2.4. Tối ưu hóa Siêu tham số (Hyperparameter Optimization)

Thay vì Grid Search tốn kém, chúng tôi sử dụng **Optuna** với thuật toán TPE (Tree-structured Parzen Estimator) để tìm kiếm bộ tham số tối ưu. Không gian tìm kiếm bao gồm:

- `learning_rate` : [0.01, 0.2] (Log scale)
- `max_depth` : [6, 12]
- `subsample` , `colsample_bytree` : [0.6, 0.9]
- `n_estimators` : [500, 1500]

Quy trình tối ưu hóa sử dụng Early Stopping (dừng sớm) để tránh Overfitting.

2.5. Khung Phát hiện Bất thường: Ngưỡng thích nghi (Adaptive Thresholding)

Phát hiện bất thường dựa trên số dư (Residual-based) yêu cầu xác định ngưỡng T. Thay vì dùng ngưỡng tĩnh, chúng tôi đề xuất phương pháp **Grouped Robust Statistics** [3].

Bước 1: Tính số dư (Residual Calculation)

$$e_{t,b} = |y_{t,b}^{real} - y_{t,b}^{pred}|$$

Bước 2: Thống kê Bền vững theo Nhóm (Grouped Robust Statistics) Đối với mỗi tòa nhà b, tính Trung vị (Median) và Khoảng tứ phân vị (IQR) của số dư. Chúng tôi sử dụng Median/IQR thay vì Mean/Std vì chúng ít bị ảnh hưởng bởi các giá trị ngoại lai cực đoan.

$$IQR_b = Q3_b(e) - Q1_b(e)$$

Bước 3: Ngưỡng Động (Contextual Threshold) Ngưỡng cảnh báo cho tòa nhà b được xác định là:

$$Threshold_b = Median_b(e) + K \times IQR_b$$

Trong đó K là hệ số độ nhạy. Thực nghiệm cho thấy $K = 3.0$ mang lại sự cân bằng tốt nhất.

Bước 4: Ngưỡng Sàn (Floor Threshold) Để tránh báo động giả khi mô hình dự báo quá tốt (sai số xấp xỉ 0), chúng tôi áp dụng ngưỡng tối thiểu là 10% tải trung bình của tòa nhà:

$$T_{final} = \max(Threshold_b, 0.1 \times \mu_b)$$

3. THIẾT LẬP THỰC NGHIỆM (EXPERIMENTAL SETUP)

3.1. Tập dữ liệu và Tiền xử lý

- **Nguồn:** Building Data Genome Project 2.

- **Quy mô thử nghiệm:** 200 tòa nhà đầu tiên (do giới hạn RAM Colab Free, tuy nhiên pipeline thiết kế sẵn sàng cho 1600+ tòa nhà).
- **Chia tập:** 80% đầu tiên cho huấn luyện (Train), 20% cuối cùng cho kiểm tra (Test). Dữ liệu được chia theo thời gian (Time-series split) để tránh rò rỉ dữ liệu (data leakage).

3.2. Giao thức Giả lập Sự cố (Fault Injection Protocol)

Do dữ liệu thực tế không có nhãn bất thường (unlabeled), chúng tôi sử dụng phương pháp tiêm lỗi nhân tạo để đánh giá định lượng [4]:

- **Loại lỗi:** Bất thường điểm (Point Anomaly) - Tăng đột biến tải.
- **Cường độ:** $y_{injected} = y_{real} \times 1.5$ (Tăng 50%).
- **Tỷ lệ:** 5% ngẫu nhiên trên tập Test.
- **Phạm vi:** Áp dụng ngẫu nhiên trên toàn bộ 200 tòa nhà.

4. KẾT QUẢ VÀ THẢO LUẬN (RESULTS & DISCUSSION)

4.1. Hiệu suất Dự báo (Forecasting Performance)

Mô hình XGBoost sau khi tối ưu hóa đạt kết quả ấn tượng trên tập kiểm tra (Global Test Set):

Chỉ số	Giá trị	Nhận xét
RMSE	**14.7247**	Sai số trung bình phương gốc thấp, cho thấy mô hình bám sát biến động tải.
MAE	**1.9217**	Sai số tuyệt đối trung bình rất nhỏ (trên thang đo thực tế).

Kết quả này khẳng định rằng việc sử dụng `log1p` và các đặc trưng `lag` đã giúp mô hình toàn cục xử lý hiệu quả sự khác biệt quy mô giữa các tòa nhà.

4.2. Hiệu suất Phát hiện Bất thường (Anomaly Detection Performance)

Đánh giá trên tập dữ liệu đã tiêm 34,701 điểm lỗi vào 200 tòa nhà:

Chỉ số	Kết quả	Ý nghĩa thực tiễn
Precision	**0.9242**	Hơn 92% các cảnh báo là chính xác. Giảm thiểu "báo động giả" gây phiền toái.
Recall	**0.9923**	Hệ thống phát hiện được hơn 99% các sự cố. Độ bỏ sót (False Negative) cực thấp.

F1-Score	**0.9570**	Hiệu suất tổng thể xuất sắc, vượt trội so với các phương pháp thống kê truyền thống.

4.3. Phân tích Trực quan (Visual Analysis)

Biểu đồ phân tích (Figure 1 trong Code) minh họa rõ ràng cơ chế hoạt động:

- Đường dự báo (xanh dương) bám sát tải nền thực tế.
- Vùng an toàn (Safe Zone - màu xanh lá cây) bao phủ các biến động ngẫu nhiên nhỏ.
- Các điểm bất thường (dấu X màu đỏ) nằm ngoài vùng an toàn đều được định vị chính xác.

5. KẾT LUẬN (CONCLUSION)

Nghiên cứu này đã trình bày một quy trình khép kín (end-to-end framework) để xây dựng hệ thống quản lý năng lượng thông minh quy mô lớn.

1. **Tính hiệu quả:** Mô hình toàn cục (Global Model) chứng minh khả năng thay thế hàng nghìn mô hình cục bộ, giúp tiết kiệm tài nguyên tính toán đáng kể.
2. **Tính chính xác:** Sự kết hợp giữa XGBoost và Optuna mang lại độ chính xác dự báo cao.
3. **Tính mạnh mẽ (Robustness):** Cơ chế ngưỡng thích nghi dựa trên IQR hoạt động hiệu quả trên nhiều loại hình tòa nhà khác nhau, đạt F1-Score ~0.96.

Hướng phát triển tiếp theo:

- Tích hợp thêm dữ liệu thời tiết (nhiệt độ, độ ẩm) để nâng cao độ chính xác dự báo.
- Thử nghiệm các loại lỗi phức tạp hơn (như lỗi trôi dạt - drift anomalies).
- Triển khai mô hình dưới dạng API thời gian thực.

TÀI LIỆU THAM KHẢO (REFERENCES)

[1] Miller, C., et al. "Building Data Genome Project 2." *Nature Scientific Data* (2020). [2] Chen, T., & Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System." *KDD '16*. [3] Stjelja, D., et al. "Building consumption anomaly detection: A comparative study of two probabilistic approaches." *Energy & Buildings* (2024). [4] Ambat, A., & Sahoo, J. "Anomaly detection and prediction of energy consumption for smart homes using machine learning." *ETRI Journal* (2024). [5] Alba, E.L., et al. "Electricity Consumption Forecasting: An Approach Using Cooperative Ensemble Learning." *Forecasting* (2024).

Báo cáo được trích xuất và tổng hợp tự động từ kết quả thực thi mã nguồn Python trên môi