DATA LABELING FOR POWER SYSTEM OPERATION MONITORING

GÁN NHÃN DỮ LIỆU PHỤC VỤ GIÁM SÁT VẬN HÀNH HỆ THỐNG ĐIỆN

Nguyễn Thị Thanh Tân, Vũ Thị Thu Nga*, Đặng Việt Hùng

Trường Đại học Điện lực

Ngày nhận bài: 01/10/2024, Ngày chấp nhận đăng: 18/10/2024, Phản biện: PGS.TS Nguyễn Quang Uy

Abstract:

During the process of collecting data for power system operation monitoring, the received data is often compiled in tabular form. This is the most common way to arrange data which makes easier to manipulate, analyze, and manage data. However, data tabular has characteristics such as missing or noisy data, many categories, many different values and difficult to apply Transfer learning that cause makes it difficult to monitor the operation of the power system with a huge amount of data. To monitor operations in real time, system data is often included in operational monitoring software with the new trend of using Artificial intelligence (AI) technology. In which, a complete database is structured and properly labeled, it is the most important and time-consuming step that helps the system provide accurate results, and the module identifies factors to train the model to achieve the best results. This study has built processes, labeling library set and labeled power system data for deployment purposes of power system operation monitoring models.

Keywords: data labeling, AI, power system operation monitoring.

Tóm tắt:

Trong quá trình thu thập dữ liệu phục vụ giám sát vận hành hệ thống điện, dữ liệu nhận được thường được thống kê ở dạng bảng, đây là cách phổ biến nhất để sắp xếp dữ liệu, giúp cho việc thao tác, phân tích và quản lý dữ liệu trở nên dễ dàng hơn. Tuy nhiên, dữ liệu ở dạng bảng có những đặc điểm như: dữ liệu bị thiếu hoặc nhiễu, có nhiều hạng mục, đặc trưng hạng mục thường có nhiều giá trị khác nhau và khó áp dụng Transfer Learning gây khó khăn trong việc theo dõi giám sát quá trình vận hành hệ thống điện với khối lượng dữ liệu khổng lồ. Để phục vụ giám sát vận hành theo thời gian thực, dữ liệu hệ thống thường được đưa vào các phần mềm theo dõi vận hành với xu hướng mới sử dụng công nghệ AI. Trong sử dụng công nghệ AI, bộ cơ sở dữ liệu hoàn chỉnh được cấu trúc và gán nhãn đúng là bước chiếm nhiều thời gian và quan trọng nhất giúp cho hệ thống cung cấp kết quả chính xác, mô đun xác định các yếu tố để huấn luyện mô hình đạt hiệu quả tốt nhất. Nghiên cứu này đã thực hiện xây dựng quy trình, bộ thư viện gán nhãn và gán nhãn dữ liệu hệ thống điện phục vụ cho mục đích triển khai các mô hình giám sát vận hành hệ thống điện.

Từ khóa: gán nhãn dữ liệu, trí tuệ nhân tạo, giám sát vận hành hệ thống điện.

1. GIỚI THIỆU

Giám sát vận hành hệ thống điện cần sử dụng dữ liệu đa nguồn, bao gồm dữ giám sát ngoài (dữ liệu hình ảnh), dữ liệu giám sát vận hành bên trong hệ thống điện thu nhận từ các hệ thống đo xa, SCADA (dữ liệu hệ thống điện), dữ liệu cơ sở hạ tầng lưới điện (PMIS, GIS, ...). Trong đó, dữ liệu hệ thống điện và dữ liệu cơ sở hạ tầng lưới điện thường được tổ chức và lưu trữ dưới dạng dự liệu dạng bảng. Trong các module phân tích và dự báo, dữ liệu hệ thống điện phải được tích hợp và đồng bộ hóa với dữ liệu hình ảnh để huấn luyện các thuật toán AI [1]-[7]. Việc huấn luyện các thuật toán AI thường đòi hỏi một lượng lớn dữ liệu hệ thống điện đã được gán nhãn [2]. Gán nhãn dữ liệu (Data Labeling) là quá trình gán nhãn cho các mẫu dữ liệu trong một tập hợp dữ liệu để xác định các thông tin cụ thể mà các mẫu đó đại diện. Đây là một bước quan trọng trong quá trình phát triển các mô hình học máy, đặc biệt là trong các bài toán giám sát. Thực tế luôn yêu cầu các các hệ thống AI có khả năng mang lại kết quả cuối cùng không chỉ chính xác mà còn phù hợp và kịp thời. Điều này đòi hỏi công đoạn gán nhãn phải chính xác, ít sai sót nhất có thể giúp mô đun xác định các yếu tố để huấn luyện mô hình đạt hiệu quả tốt nhất. Bất kỳ mô hình hoặc hệ thống nào có hệ thống ra quyết định do máy điều khiển ở điểm tựa, cần có gán nhãn dữ liệu để đảm bảo các quyết định là chính xác và phù hợp.

Bài báo này tập trung vào việc xây dựng quy trình và bộ thư viện để gán nhãn dữ liệu hệ thống điện đồng thời thực hiện gán nhãn trên các dữ liệu thu thập được từ đa nguồn phục vụ giám sát vận hành hệ thống điện. Để xây dựng thư viện gán nhãn dữ liệu hệ thống điện, trước tiên nhóm đã tiến hành khảo sát các đặc trưng của dữ liệu dạng bảng, các thuật toán gán nhãn dữ liệu dạng bảng, các yêu cầu về tính năng thực tế đối với việc gán nhãn dữ liệu. Trên cơ sở đó, nhóm đã phân tích, lựa chọn và đề xuất quy trình gán nhãn dữ liệu hệ thống điện cũng như các tính năng cần thiết của thư viện gán nhãn và thực hiện gán nhãn.

Các đóng góp chính của chúng tôi trong bài báo này bao gồm :

Thứ nhất, chúng tôi đề xuất quy trình và bộ công cụ gán nhãn dữ liệu hệ thống điện nhằm nâng cao hiệu quả của các thuật toán học máy trong dự báo bất thường trên lưới truyền tải điện cao thế.

Thứ hai, chúng tôi đề xuất phương pháp gán nhãn dữ liệu bán tự động dựa trên mô hình lựa chọn đặc trưng và phân lớp dữ liệu hệ thống điện.

Phần còn lại của bài báo được cấu trúc như sau: Phần 2 cung cấp tóm tắt về các hướng tiếp cận liên quan. Quy trình và phương pháp gán nhãn dữ liêu đề xuất được mô tả chi tiết trong phần 3. Phần 4 trình bày kết quả thực nghiệm. Một số kết luận và hướng đi trong tương lai được thảo luận trong phần 5.

2. HƯỚNG TIẾP CẬN LIÊN QUAN

Trong bối cảnh phát triển của học máy đặc biệt là học sâu. Nhiều kỹ thuật và công cụ gán nhãn dữ liệu đã đề xuất. Các kỹ thuật này về cơ

bản có thể chia thành ba hướng tiếp cận chính: Gán nhãn thủ công (manual annotation), gán nhãn bán tự động (semi-automatic annotation) và gán nhãn tự động (automatic annotation).

Trong hướng tiếp cận gán nhãn thủ công, người gán nhãn (chuyên gia) sẽ xác định và gán nhãn cho các đối tượng, dữ liệu hoặc thông tin trong một tập hợp dữ liệu cụ thể. Gán nhãn thủ công có thể đạt được độ chính xác cao vì con người có thể hiểu các mẫu phức tạp và những biến thể mà các hệ thống tự động có thể bỏ lỡ. Tuy nhiên, việc gán nhãn thủ công thường tốn nhiều thời gian và công sức, đặc biệt khi cần gán nhãn một khối lượng lớn dữ liệu. Ngoài ra, những người gán nhãn khác nhau có thể diễn giải các bất thường một cách khác nhau, dẫn đến sự không nhất quán trong các mẫu được gán nhãn. Các công cụ thường được sử dụng cho việc gán nhãn thủ công bao gồm:

Labelbox được thiết kế với giao diện trực quan, giúp người dùng dễ dàng tạo, quản lý và gán nhãn dữ liệu. Công cụ này hỗ trợ nhiều loại gán nhãn như phân loại, khoanh vùng, và gán nhãn theo đối tượng. Tuy nhiên, công cụ này bị hạn chế về tính năng tùy chỉnh sâu hơn cho quy trình gán nhãn. Ngoài ra, đây là một công cụ tính phí nên có thể tốn kém đối với khối lượng dữ liệu lớn.

Label Studio là một công cụ gán nhãn dữ liệu mã nguồn mở, cho phép người dùng gán nhãn cho nhiều loại dữ liệu khác nhau như dữ liệu dạng bảng, hình ảnh, video và âm thanh. Người dùng tùy chỉnh giao diện gán nhãn và quy trình làm việc theo nhu cầu cụ thể. Tuy nhiên, để sử dụng công cụ này cần một số kiến thức kỹ thuật để cài đặt và cấu hình, đặc biệt khi triển khai trên máy chủ riêng. Mặc dù là mã nguồn mở nhưng một số tính năng nâng cao có thể yêu cầu phí sử dụng hoặc cài đặt phức tạp.

Gán nhãn bán tự động kết hợp đầu vào của con người với các công cụ tự động để tăng tốc quá trình gán nhãn. Các phương pháp tự động có thể đề xuất các nhãn, sau đó người gán nhãn sẽ tinh chỉnh và xác thực. Các công cụ tự động có thể nhanh chóng tạo ra các chú thích ban đầu, giảm thiểu thời gian cần thiết cho người gán nhãn. Tuy nhiên, đối với các phương pháp gán nhãn bán tự động, việc thiết lập và hiệu chỉnh các công cụ tự động đòi hỏi thời gian và kiến thức chuyên môn. Hiệu quả tổng thể của gán nhãn bán tự động thường phụ thuộc vào độ chính xác và độ tin cậy của các công cụ tự động được sử dụng. Hầu hết các công cụ gán nhãn bán tự động đều là các phần mềm trả phí, điển hình gồm Prodigy, Dataloop, SuperAnnotate. Các công cụ này cung cấp các công cụ để gán nhãn, tổ chức và quản lý dữ liệu một cách hiệu quả. Tuy nhiên, việc sử dụng công cụ Thương mại thường rất tốn kém, đặc biệt khi lượng dữ liệu cần gán nhãn lớn. Mặc dù giao diện thân thiện, người dùng vẫn cần có kiến thức về học máy để tận dụng tối đa các tính năng. Noài ra, chất lượng nhãn gợi ý phụ thuộc vào mô hình học máy, có thể không chính xác trong một số trường hợp.

Gán nhãn tự động là quá trình sử dụng các thuật toán và mô hình học máy để tự động gán nhãn cho dữ liệu mà không cần sự can thiệp của con người. Quá trình này thường được áp

dụng trong các lĩnh vực như học máy, trí tuệ nhân tạo và xử lý dữ liệu lớn, nhằm tăng tốc độ và hiệu quả trong việc chuẩn bị dữ liệu cho các mô hình học máy. Phương pháp gán nhãn tự động giảm thiểu sự cần thiết phải có đội ngũ gán nhãn lớn, tiết kiệm chi phí cho dự án, có thể xử lý lượng lớn dữ liệu mà không cần tăng cường lực lượng lao động, giúp duy trì tính nhất quán trong gán nhãn, giảm thiểu sai sót do yếu tố con người. Tuy nhiên, chất lượng nhãn tự động phụ thuộc vào độ chính xác của mô hình học máy; nếu mô hình không đủ tốt, nhãn có thể không chính xác. Hướng tiếp cận này cần có một tập dữ liệu lớn và có nhãn để huấn luyện mô hình, điều này có thể tốn thời gian và công sức. Các mô hình có thể gặp khó khăn trong việc xử lý các tình huống không quen thuộc hoặc không được dự đoán trước. Gán nhãn tự động hiện vẫn là một hướng tiếp cận mới, chưa đạt được kết quả vượt trội để có thể thay thế các công cụ gán nhãn thủ công hoặc bán tự động. Các kỹ thuật gán nhãn tự động dựa trên các mô hình ngôn ngữ lớn LLMs (Large Language Models) đang là một xu thế mới trong lĩnh vực gán nhãn tự động [8]-[11].

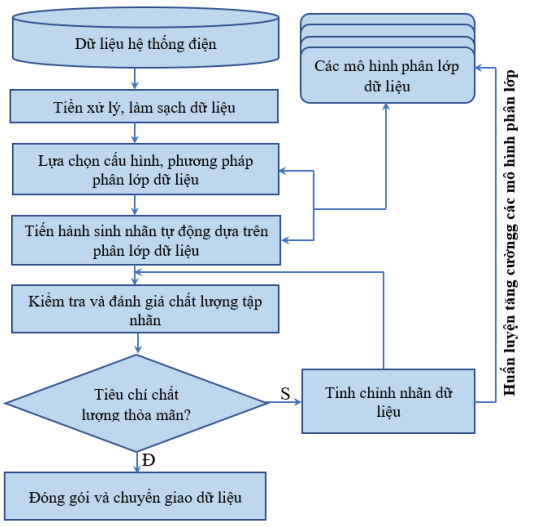
3. XÂY DỰNG QUY TRÌNH GÁN NHÃN DỮ LIỆU

Các khảo sát từ thực tế cho thấy trong quá trình gán nhãn dữ liệu thường gặp phải một số vấn đề như dữ liệu có chất lượng gán nhãn thấp, tốc độ gán nhãn dữ liệu chậm, nhãn dữ liệu không nhất quán, v.v. Chúng tôi đề xuất quy quy trình gán nhãn dữ liệu hệ thống điện theo hướng tiếp cận bán tự động (xem Hình 1) nhằm tận dụng ưu thế của các mô hình phân lớp dữ liệu trong sinh nhãn tự động, giảm thiểu thời gian và công sức của việc gán nhãn dữ liệu.

Đầu vào của quy trình gán nhãn là tập dữ liệu hệ thống điện được trích xuất từ hệ thống SCADA và được lưu trữ dưới dạng bảng tính excel (.csv, .xlsx). Quy trình gán nhãn dữ liệu bao gồm các bước cơ bản như sau:

B1. Tiền xử lý dữ liệu:

Dữ liệu ngoại lai (dữ liệu bị sai lệch do lỗi cảm biến hoặc lỗi truyền thông) và dữ liệu bị thiếu (một số giá trị có thể không được ghi lại do sự cố kỹ thuật hoặc mất kết nối) là hai vấn đề thường gặp đối với dữ liệu SCADA. Trong quy trình đề xuất, tiền xử lý dữ liệu là một bước quan trọng, được thực hiện nhằm chuẩn hóa dữ liệu, bỏ nhiễu và các dữ liệu ngoại lai, khắc phục các thuộc tính dữ liệu bị thiếu, hướng tới đảm bảo chất lượng đầu vào cho các mô hình sinh nhãn tự động.



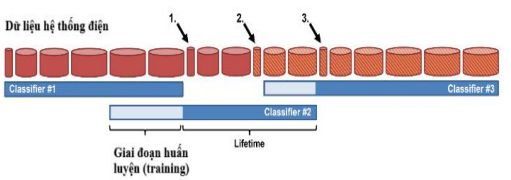
Hình 1. Quy trình gán nhãn dữ liệu

B2. Lựa chọn cấu hình, phương pháp phân lớp dữ liệu:

Trong quy trình đề xuất, các thuật toán phân lớp dữ liệu được sử dụng để xác định nhãn tự động cho từng mẫu dữ liệu. Độ chính xác của thuật toán phân lớp quyết định hiệu suất của việc gán nhãn. Cụ thể, độ chính xác của thuật toán càng cao thì chất lượng của tập nhãn tự động càng tốt có nghĩa là tỷ lệ mẫu cần phải gán nhãn hoặc tinh chỉnh càng ít. Bộ phân lớp dữ liệu ở đây được xây dựng dựa trên mô hình học kết hợp XGBoost thích nghi.

B3. Sinh nhãn dữ liệu tự động

XGBoost (Extreme Gradient Boosting) là một thuật toán học máy mạnh mẽ, được sử dụng rộng rãi trong các bài toán phân loại và hồi quy. XGBoost xây dựng mô hình bằng cách kết hợp nhiều cây quyết định. Mỗi cây mới sẽ học từ các sai số của cây trước đó. XGBoost Sử dụng gradient descent để tối ưu hóa hàm mất mát, giúp cải thiện độ chính xác của mô hình. Phương pháp sinh nhãn tự động dựa trên mô hình XGBoost được mô tả trên Hình 2:



Hình 2. Phương pháp sinh nhãn tự động

Thuật toán 1 trình bày mã giả của phương pháp sinh nhãn tự động trong Hình 2. Trong đó, các tham số chính của mô hình sinh nhãn bao gồm:

max_w: Kích thước cửa sổ lớn nhất;

W: cửa sổ trượt (sliding window);

lifetime: Thời gian của mỗi bộ phân lớp;

training_time: Thời gian huấn luyện mỗi bộ phân lớp

count: Là biến đếm để đếm xem có bao nhiêu của sổ trượt đã được sử dụng;

M: Là mô hình tốt nhất tại mỗi bước huấn luyện;

nextM: Là mô hình hiện tại được xét;

model_update: Giá trị cờ (flag), nhận giá trị đúng (True) hoặc sai (False) để xác định có cập nhật mô hình hay không;

active_reset: Giá trị cờ (True/False) để xác định có khởi tạo lại mô hình hay không;

update_M: Hàm cập nhật mô hình M;

update_nextM: Hàm cập nhật bộ phân lớp bộ phân lớp nextM;

train_new_tree_nextM: Hàm huấn luyện cây phân lớp mới nextM;

train_new_classification: Hàm huấn luyện bộ phân lớp mới.

Thuật toán bắt đầu bằng cách thêm vào cửa sổ trượt *W* các mẫu huấn luyện. Sau đó, tiến hành kiểm tra xem kích thước của cửa có thỏa mãn hay không. Nếu có, một bản sao của w trường hợp đầu tiên từ cửa sổ *W* được sao chép sang cửa sổ *W'*, và các mô hình hiện tại và tiếp theo

được tải.

Thuật toán bắt đầu bằng cách thêm vào cửa sổ trượt *W* các mẫu huấn luyện. Sau đó, tiến hành kiểm tra xem kích thước của cửa có thỏa mãn hay không. Nếu có, một bản sao của w trường hợp đầu tiên từ cửa sổ *W* được sao chép sang cửa sổ *W′*, và các mô hình hiện tại và tiếp theo được tải.

Thuật toán 1: Thuật toán huấn luyện mô hình sinh nhãn tự động P-XGBoost
Input: $X=\{(x,y)_i, i=1 \rightarrow n\}$
1. Thêm mẫu (x, y) vào cửa sổ (W);
2. count ←0;
3. if $|W| > w$ then
4. W' ← sao chép w dữ liệu đầu tiên từ W ;
5. Tải mô hình (M) trước đó;
6. Tải mô hình $nextM$ trước đó;
7. if $M \neq$ NULL then
8. if *training_time* ≥ (*lifetime* - *count*) then
9. if $nextM \neq$ NULL then
10. Nếu *model_update* = True thì
12. $nextM$ ← updatc_nextM(W');
13. $nextM$ ← train_new_trec_nextM(W');
14. save_nextM($nextM$);
15. else:
16. $nextM$ ← train_new_classification (W');
17. save_nextM($nextM$);
18. else if *count* ≥ *lifetime* then:
19. $M \leftarrow nextM$;
20. $nextM$ ← NULL
21. *count* ←0
22. if *active_reset* = *True* then:
23. $w \leftarrow 0$
24. if mode_update = True then:
25. M ← update_M(W')
26. M ←train_new_tree_M (W')
27. save_M(M);
28. else: M ←train_new_classifier (W'); save_M(M);
29. if detect-drift = True then:
30. incorrect_class ← not (M.predict(X) =y)
31. ADWIN.add (incorrect_class)
32. if ADWIN.drift_detection = True then:
33. $w \leftarrow 0$
34. $W = W - W'$
35. Count = count + 1
36. if $w <$ max _w then:
$w = w + 1$
37. return M

Sau đó, thuật toán kiểm tra xem có một bộ phân lớp đã được xây dựng chưa. Nếu không, một bộ phân lớp XGBoost mới được huấn luyện và lưu lại để cập nhật sau này. Nếu đã có một bộ phân lớp XGBoost đã được huấn luyện (trained), nếu giá trị cờ model_update là True thì các bộ phân lớp yếu (weak classifier) hiện có được cập nhật, một bộ phân lớp yếu mới được huấn luyện và bộ phân lớp XGBoost được lưu lại để cập nhật sau này.

Tùy thuộc vào số lượng cửa sổ được xử lý, cần phải bắt đầu huấn luyện một bộ phân lớp mới để thay thế bộ phân lớp hiện tại sau này. Do đó, khi thời gian huấn luyện bộ phân lớp XGBoost hiện tại vượt quá một khoảng thời gian đã định trước, một bộ phân lớp XGBoost mới cần được huấn luyện. Nếu đã có một bộ phân lớp XGBoost mới, nếu cờ *model_update* là True, các bộ phân lớp yếu hiện có sẽ được cập nhật, một bộ phân lớp yếu mới được huấn luyện và bộ phân lớp XGBoost mới được lưu lại để cập nhật để sau đó cập nhật.

Tuy nhiên, nếu bộ phân lớp đã đạt đến lifetime (tuổi thọ) của nó thì bộ phân lớp đó sẽ được thay thế bằng bộ phân lớp mới và bộ đếm cửa sổ trượt được khởi tạo lại (đặt lại = 0). Nếu cờ active-reset là True thì kích thước w của cửa sổ được đặt lại về giá trị tối thiểu của nó.

Ngoài ra, đối với mỗi cửa sổ trượt đã xử lý, nếu cờ *detect_drift* được bật thì cửa sổ trượt thích nghi ADWIN được sử dụng. Sau đó, kích thước w của cửa sổ được đặt lại về giá trị tối thiểu của nó. Cuối cùng, cửa sổ trượt được dịch chuyển, bộ đếm của cửa sổ trượt được tăng lên và kích thước w được tăng lên nếu chưa đạt đến kích thước cửa sổ tối đa.

Các siêu tham số được tinh chỉnh trong thuật toán bao gồm:

- Tốc độ học (eta): Một giá trị giữa 0 và 1 cho biết tốc độ mà mô hình sẽ học từ dữ liệu đã được

xử lý khi một bộ học yếu mới được đưa vào.

- Độ sâu tối đa (max_depth): Độ sâu tối đa mà cây huấn luyện yếu (weak learner tree) yếu có thể đạt được. Giá trị này thường nằm trong khoảng từ 2 đến 10.

- Kích thước cửa sổ tối đa (max_window_size): Một giá trị nguyên xác định kích thước tối đa của cửa sổ trượt lưu trữ dữ liệu đầu vào.

- Thời gian sống của bộ phân loại (lifetime): Một giá trị nguyên xác định thời gian sống của mô hình XGBoost hiện tại trước khi bị thay thế.

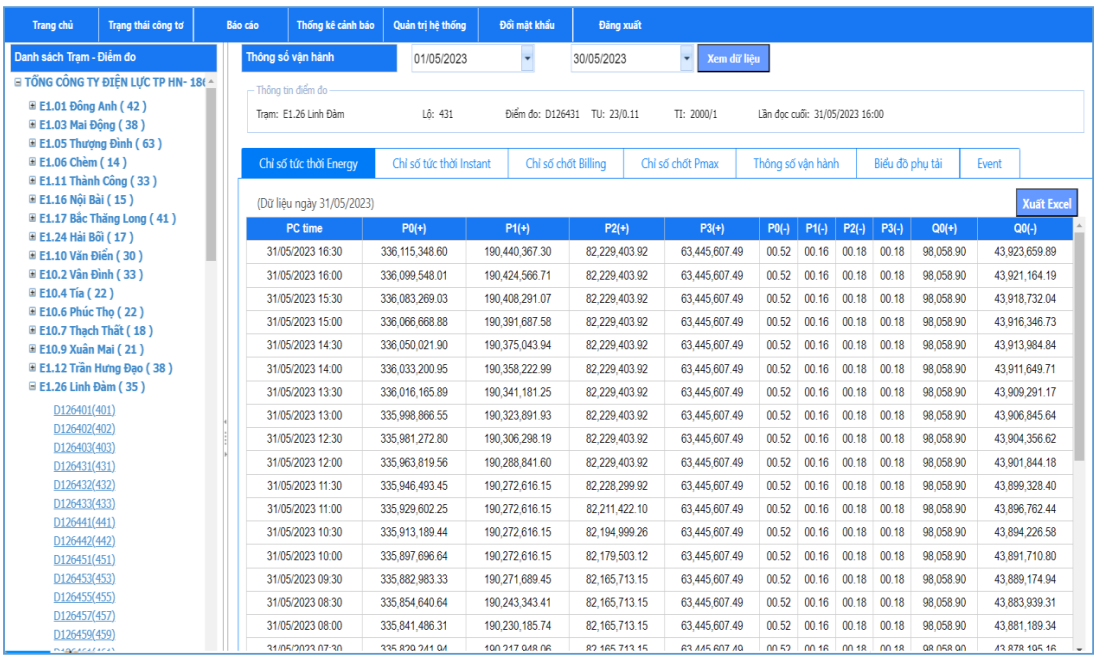
- Thời gian huấn luyện (training_time): Một giá trị nguyên xác định thời gian huấn luyện của mỗi bộ phân loại phụ trợ (auxiliary classifier). Giá trị này thường nằm trong khoảng từ 60% đến 80% thời gian sống của mô hình XGBoost hiện tại.

- Detect_drift: Một giá trị boolean xác định xem ADWIN có được sử dụng hay không.

- Số lượng lớp (num_classes): Một giá trị nguyên xác định có bao nhiêu lớp trong luồng dữ liệu.

B4. Kiểm tra và đánh giá chất lượng tập nhãn:

Như đã đề cập, hiệu quả của các phương pháp gán nhãn bán tự động nhìn chung phụ thuộc rất nhiều vào phương pháp sinh nhãn tự động. Mặc dù, bộ phân lớp XGBoost được đánh là một trong những mô hình phân lớp tốt nhất đối với dữ liệu dạng bảng. Tuy nhiên, trong một số trường hợp như dữ liệu còn nhiễu (noise data), dữ liệu ngoại lai (outlier) hoặc dữ liệu có sự khác biệt với đa số thì độ chính xác phân lớp của mô hình sẽ không đảm bảo, dẫn tới một số nhãn dữ liệu sai. Do đó, công tác kiểm tra, đánh giá chất lượng tập nhãn cần phải được



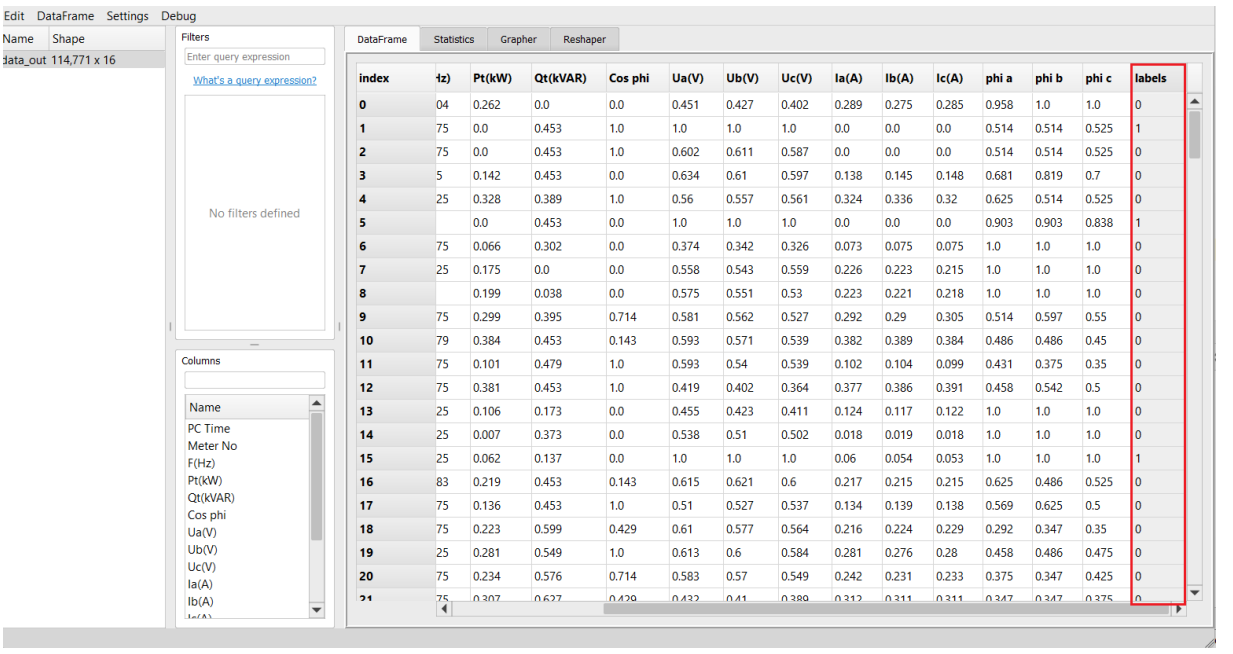
Hình 3. Ứng dụng kết nối, truy xuất thông tin giám sát vận hành

thực hiện nhằm phát hiện, loại bỏ hoặc tinh chỉnh lại các nhãn dữ liệu bị sai.

Các nhãn lớp sau khi được tinh chỉnh sẽ được cung cấp ngược lại để huấn luyện tăng cường các mô hình sinh nhãn tự động, hướng tới giảm thiểu sai số của mô hình này. Bước tinh chỉnh và huấn luyện tăng cường mô hình phân lớp XGBoost được tiến hành lặp đi lặp lại cho tới khi các tiêu chí chất lượng của tập mẫu được thỏa mãn. Đóng gói và chuyển giao dữ liệu là bước xử lý cuối cùng trong quy trình gán nhãn.

4. THỰC HIỆN GÁN NHÃN DỮ LIỆU HỆ THỐNG ĐIỆN

Theo phân cấp quản lý của Tập đoàn điện lực Việt Nam, dữ liệu từ các trạm biến áp được truyền về trung tâm giám sát vận hành (OCC) đặt tại Tổng công ty. Dữ liệu được quản lý theo cơ chế web sever, cho phép các công ty Điện lực kết nối, truy xuất và khai thác thông tin. Dữ liệu thu thập được phân cấp quản lý theo từng TBA, tại mỗi trạm dữ liệu được quản lý theo các điểm đo. Thông tin tại mỗi điểm đo được phân chia thành các mục như chỉ số tức thời Energy, chỉ số tức thời Instant, chỉ số chốt Billing, chỉ số chốt Pmax, thông số vận hành (Hình 3). Để chuẩn bị dữ liệu gán nhãn và huấn luyện mô hình, nghiên cứu đã xây dựng bộ công cụ kết nối và trích xuất dữ liệu từ web server của Tổng công ty điện lực quản lý hệ thống. Các bước trích xuất dữ liệu được thực hiện hoàn toàn tự động. Trước tiên chương trình sẽ tự động thiết lập các tham số môi trường để kết nối và đăng nhập hệ thống, sau khi đăng nhập thành công thuật toán trích xuất dữ liệu sẽ tiến hành duyệt lần lượt từng trạm biến áp trong danh sách trạm. Do dữ liệu của mỗi trạm được phân cấp quản lý theo các



index	tz)	Pt(kW)	Qt(kVAR)	Cos phi	Ua(V)	Ub(V)	Uc(V)	Ia(A)	Ib(A)	Ic(A)	phi a	phi b	phi c	labels
0	04	0.262	0.0	0.0	0.451	0.427	0.402	0.289	0.275	0.285	0.958	1.0	1.0	0
1	75	0.0	0.453	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.514	0.514	0.525	1
2	75	0.0	0.453	1.0	0.602	0.611	0.587	0.0	0.0	0.0	0.514	0.514	0.525	0
3	5	0.142	0.453	0.0	0.634	0.61	0.597	0.138	0.145	0.148	0.681	0.819	0.7	0
4	25	0.328	0.389	1.0	0.56	0.557	0.561	0.324	0.336	0.32	0.625	0.514	0.525	0
5		0.0	0.453	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.903	0.903	0.838	1
6	75	0.066	0.302	0.0	0.374	0.342	0.326	0.073	0.075	0.075	1.0	1.0	1.0	0
7	25	0.175	0.0	0.0	0.558	0.543	0.559	0.226	0.223	0.215	1.0	1.0	1.0	0
8		0.199	0.038	0.0	0.575	0.551	0.53	0.223	0.221	0.218	1.0	1.0	1.0	0
9	75	0.299	0.395	0.714	0.581	0.562	0.527	0.292	0.29	0.305	0.514	0.597	0.55	0
10	79	0.384	0.453	0.143	0.593	0.571	0.539	0.382	0.389	0.384	0.486	0.486	0.45	0
11	75	0.101	0.479	1.0	0.593	0.54	0.539	0.102	0.104	0.099	0.431	0.375	0.35	0
12	75	0.381	0.453	1.0	0.419	0.402	0.364	0.377	0.386	0.391	0.458	0.542	0.5	0
13	25	0.106	0.173	0.0	0.455	0.423	0.411	0.124	0.117	0.122	1.0	1.0	1.0	0
14	25	0.007	0.373	0.0	0.538	0.51	0.502	0.018	0.019	0.018	1.0	1.0	1.0	0
15	25	0.062	0.137	0.0	1.0	1.0	1.0	0.06	0.054	0.053	1.0	1.0	1.0	1
16	83	0.219	0.453	0.143	0.615	0.621	0.6	0.217	0.215	0.215	0.625	0.486	0.525	0
17	75	0.136	0.453	1.0	0.51	0.527	0.537	0.134	0.139	0.138	0.569	0.625	0.5	0
18	75	0.223	0.599	0.429	0.61	0.577	0.564	0.216	0.224	0.229	0.292	0.347	0.35	0
19	25	0.281	0.549	1.0	0.613	0.6	0.584	0.281	0.276	0.28	0.458	0.486	0.475	0
20	75	0.234	0.576	0.714	0.583	0.57	0.549	0.242	0.231	0.233	0.375	0.347	0.425	0
21	75	0.307	0.627	0.429	0.432	0.41	0.389	0.312	0.311	0.311	0.347	0.347	0.375	0

Hình 4. Kết quả của thuật toán gán nhãn dữ liệu

điểm đo nên bước tiếp theo thuật toán sẽ tiếp tục duyệt từng điểm đo trong trạm để trích xuất dữ liệu trong từng mục thông tin của mỗi điểm đo. Từ các kết quả khảo sát, phân tích trong quá trình thực hiện đề tài cho thấy các thông số vận hành có thể có trọng số cao (có khả năng tác động đến kết quả đầu ra của thuật toán) bao gồm: thời gian (thời điểm đo), công suất tác dụng (P), công suất phản kháng (Q), hệ số công suất (cosj), điện áp pha a, b, c (Ua, Ub, Uc), dòng điện pha a, b, c (Ia, Ib, Ic), góc lệch pha giữa dòng và áp pha a, b, c (ja, jb, jc), tần số dòng điện (f). Nguyên lý thực hiện gán nhãn dữ liệu hệ thống điện được mô tả cụ thể như sau: từ nguồn dữ liệu thô đầu vào đã được thu thập, trước tiên bằng kinh nghiệm chuyên gia, chúng tôi trích ra một phần dữ liệu để huấn luyện mô hình XGBoost sinh nhãn tự động. Các chuyên gia con người sau đó cần kiểm tra và thẩm định lại toàn bộ tập dữ liệu đã được gán nhãn tự động để đảm bảo chất lượng của tập mẫu huấn luyện, hạn chế sai sót và tránh làm sai lệch kết quả của thuật toán.

Theo cơ chế hoạt động của hệ thống SCADA, dữ liệu được truyền về trung tâm theo một chu kỳ thời gian (do người dùng thiết lập, chẳng hạn 10 phút, 20 phút hoặc 30 phút trên một chu kỳ). Như vậy, mỗi hàng trong bảng là dữ liệu được thu thập tại mỗi thời điểm t. Để phục vụ cho mục đích tích hợp dữ liệu đa nguồn trong một thao tác phân tích của hệ thống AI, mỗi mẫu dữ liệu ở đây được gán một trong hai giá trị nhãn là 0 hoặc 1 tương ứng với trạng thái bình thường hoặc bất thường. Một mẫu dữ liệu được coi là bình thường nếu số giá trị thuộc tính trong mẫu dữ liệu đó nhỏ hơn một giá trị ngưỡng a cho trước và ngược lại nếu số giá trị thuộc tính trong một mẫu dữ liệu lớn hơn hoặc bằng một giá trị ngưỡng a thì mẫu đó được coi là bất thường. Để thuận tiện cho việc xử lý và kiểm soát, dữ liệu được chia thành nhiều file theo một quy cách mã hóa cụ thể. Dữ liệu sau khi nạp vào hệ thống, thuật toán sẽ tiến hành xét lần lượt từng thuộc tính (từng cột) trong bảng và áp dụng thuật toán phát hiện bất thường cho cột dữ liệu đó. Trường hợp phát hiện có bất thường trên cột dữ liệu, thuật toán sẽ ghi nhận các bất thường được phát hiện.

Đối với một mẫu dữ liệu bất kỳ, nếu giá trị nhãn đúng và giá trị nhãn được sinh là giống nhau thì nhãn được sinh được coi là chính xác. Cuối cùng, độ chính xác của thuật toán gán nhãn được tính bằng số mẫu được gán đúng trên tổng số mẫu cần gán. Nghiên cứu đã tiến hành đánh giá chất lượng của thuật toán trên một tập dữ liệu thử nghiệm gồm 3000 mẫu. Kết quả thực nghiệm cho thấy thuật toán đạt độ chính xác khoảng trên 80% (nghĩa là có khoảng trên 80% số mẫu được sinh nhãn đúng với thực tế).

5. KẾT LUẬN

Gán nhãn dữ liệu tự động là một bước xử lý quan trọng, giúp tăng hiệu suất, giảm thời gian và công sức trong việc gán nhãn. Chất lượng của một thuật toán gán/sinh nhãn tự động thường được tính bằng độ chính xác của tập nhãn được sinh. Nghiên cứu tập trung vào việc gán nhãn dữ liệu vận hành HTĐ, phục vụ tích hợp xây dựng các thuật toán phân tích dự báo bất thường trong kiểm tra, giám sát lưới truyền

tải điện 110 kV. Cụ thể nghiên cứu đã xây dựng quy trình gán nhãn, thư viện gán nhãn và tập trung vào các vấn đề liên quan đến gán nhãn dữ liệu HTĐ bao gồm trích xuất đặc trưng dữ liệu, quản lý, lưu trữ và biểu diễn dữ liệu, v.v. Nghiên cứu đã căn cứ vào các thông số đặc trưng có khả năng tác động/ảnh hưởng đến các kết quả phân tích, dự báo của các thuật toán AI), các tiêu chuẩn qui phạm đã được quy định trong các thông tư quy định hệ thống điện phân phối, kinh nghiệm của chuyên gia con người để đề xuất thuật toán gán nhãn hệ thống điện. Các dữ liệu sau khi được gán sẽ được thẩm định, đánh giá trực tiếp bởi chuyên gia con người.

Lời cảm ơn

Nhóm tác giả trân trọng cảm ơn chương trình hỗ trợ nghiên cứu, phát triển và ứng dụng công nghệ của công nghiệp 4.0, mã số: KC-4.0.31/19-25, đã hỗ trợ trong quá trình nghiên cứu.

TÀI LIỆU THAM KHẢO

[1]. Matthias Klumpp, Benedikt Severin, Henrik Lechte, Jannes Heinrich Diedrich Menck, and Maria Keil, "Driving Big Data - Integration and Synchronization of Data Sources for Artificial Intelligence Applications with the Example of Truck Driver Work Stress and Strain Analysis," Int. Conf. Inf. Syst., vol. Proceedings. 3, 2022.

[2]. James Densmore, Data Pipelines Pocket Reference: Moving and Processing Data for Analytics. O'Reilly Media, 2021.

[3]. Fredriksson, T., Mattos, D.I., Bosch, J., Olsson, H.H., "Data Labeling: An Empirical Investigation into Industrial Challenges and Mitigation Strategies," Prod.-Focus. Softw. Process Improv. PROFES 2020 Lect. Notes Comput. Sci., vol. 12562, 2020, doi: https://doi.org/10.1007/978-3-030-64148-1_13.

[4]. N. Kshetri, "Data Labeling for the Artificial Intelligence Industry: Economic Impacts in Developing Countries," IT Prof., vol. 23, no. 2, pp. 96–99, 2021, doi: 10.1109/MITP.2020.2967905.

[5]. Wei Lee, Chien-Wei Chang, Po-An Yang, Chi-Hsuan Huang, Ming-Kuang Daniel Wu, Chu-Cheng Hsieh, Kun-Ta Chuang, "Effective Quality Assurance for Data Labels through Crowdsourcing and Domain Expert Collaboration," Int. Conf. Extending Database Technol., pp. 646–649, 2018.

[6]. H. Zhang et al, "Hierarchical Crowdsourcing for Data Labeling with Heterogeneous Crowd," IEEE 39th Int. Conf. Data Eng. ICDE, pp. 1234–1246, 2023, doi: 10.1109/ICDE55515.2023.00099.

[7]. J. Zhou, R. Cao, J. Kang, K. Guo and Y. Xu, "An Efficient High-Quality Medical Lesion Image Data Labeling Method Based on Active Learning," IEEE Access, vol. 8, pp. 144331–144342, 2020, doi: 0.1109/ACCESS.2020.3014355.

[8]. Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, Christos Faloutsos, "Large Language Models(LLMs) on Tabular Data: Prediction, Generation, and Understanding -- A Survey", https://doi.org/10.48550/arXiv.2402.17944.

[8]. Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, Huan Liu, "Large Language Models for Data Annotation and Synthesis: A Survey", https://doi.org/10.48550/arXiv.2402.13446.

[9]. Jing Zhang, Xindong Wu, Victor S. Sheng, "Learning from crowdsourced labeled data: a survey," Artif Intell Rev, vol. 46, pp. 543–576, 2016, doi: https://doi.org/10.1007/s10462-016-9491-9.

[10].Yucheng Ruan, Xiang Lan, Jingying Ma, Yizhi Dong, Kai He, Mengling Feng, "Language Modeling on Tabular Data: A Survey of Foundations, Techniques and Evolution", https://doi.org/10.48550/arXiv.2408.10548

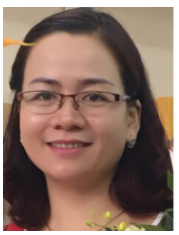
[11]. Junyuan Hong. Lingjuan Lyu, Jiayu Zhou, Michael Spranger, "Outsourcing Training without Uploading Data via Efficient Collaborative Open-Source Sampling," 36th Conf. Neural Inf. Process. Syst. NeurIPS 2022, pp. 1–14, 2022.

[12]. M. W., Python Data Cleaning Cookbook: Modern techniques and Python tools to detect and remove dirty data and extract key insights, Packt Publishing. 2020.

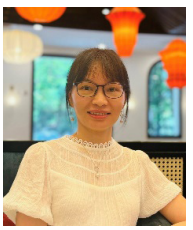
[13]. "Thông tư Thông tư số 39/2015/TT-BCT của Bộ Công thương: Quy định hệ thống điện phân phối ngày 18-11-2015."

[14]. "Thông tư số 39/2022/TT-BCT của Bộ Công thương: Sửa đổi, bổ sung một số điều của Thông tư số 25/2016/TT-BCT ngày 30 tháng 11 năm 2016 của Bộ trưởng Bộ Công Thương quy định hệ thống điện truyền tải."

Giới thiệu tác giả:



Tác giả Nguyễn Thị Thanh Tân tốt nghiệp đại học và nhận bằng Thạc sĩ lần lượt trong các năm 1999 và 2001 chuyên ngành Khoa học máy tính, ngành Công nghệ Thông tin tại Đại học Công nghệ - Đại học Quốc gia Hà Nội. Nhận bằng tiến sĩ chuyên ngành Khoa học máy tính tại Viện Hàn Lâm-Viện Khoa học và Công nghệ Việt Nam. Hiện nay tác giả công tác tại Trường Đại học Điện lực.



Lĩnh vực nghiên cứu: Xử lý ảnh, học máy, công nghệ tri thức, khai phá dữ liệu, trí tuệ nhân tạo (AI).

Tác giả Vũ Thị Thu Nga tốt nghiệp đại học ngành hệ thống điện năm 2004, nhận bằng Thạc sĩ ngành kỹ thuật điện năm 2007 tại Đại học Bách khoa Hà Nội; nhận bằng Tiến sĩ ngành kỹ thuật điện năm 2014 tại Đại học Toulouse III (Paul Sabatier) – Pháp. Năm 2023, tác giả được công nhận chức danh Phó giáo sư và hiện nay là giảng viên Trường Đại học Điện lực.

Lĩnh vực nghiên cứu: Tích điện không gian, HVDC, vật liệu cách điện, kỹ thuật điện cao áp, rơle và tự động hóa trong hệ thống điện.

Tác giả Đặng Việt Hùng tốt nghiệp đại học và nhận bằng Thạc sĩ lần lượt vào các năm 2002 và 2004 chuyên ngành Hệ thống điện tại Đại học Bách khoa Hà Nội; nhận bằng Tiến sĩ Kỹ thuật Điện tại École Centrale de Lyon, Pháp vào năm 2010. Hiện nay tác giả là giảng viên tại Đại học Điện lực Hà Nội, Việt Nam.

Lĩnh vực nghiên cứu: chất lượng điện năng, vật liệu kỹ thuật điện cao áp, tự động hóa hệ thống cung cấp điện.