TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO MÔN HỌC

TOÁN CHO TRÍ TUỆ NHÂN TẠO

DỰ ĐOÁN LỖI CỦA CÁNH QUẠT ĐIỆN GIÓ BẰNG MÔ HÌNH HỌC SÂU MLP

HỌC KỲ 2 – NĂM HỌC: 2024 - 2025

Giảng viên hướng dẫn: Ths.Bùi Mạnh Quân

Nhóm sinh viên thực hiện: Hồng Anh Khoa MSSV: 22110351

Trịnh Tấn Hào MSSV: 22110315

Trần Vũ Khanh MSSV: 22110349

TP. HCM, tháng 5 năm 2025

MỤC LỤC

LỜI NÓI ĐẦU

Trong bối cảnh công nghiệp năng lượng tái tạo phát triển mạnh mẽ, turbine gió đã trở thành một nguồn cung cấp năng lượng quan trọng, góp phần vào sự bền vững của môi trường. Tuy nhiên, các turbine gió thường hoạt động trong điều kiện khắc nghiệt, dẫn đến các lỗi kỹ thuật như nứt cánh, xói mòn, mất cân bằng hoặc xoắn cánh, gây ảnh hưởng đến hiệu suất và tuổi thọ của hệ thống. Việc phát hiện và chẩn đoán sớm các lỗi này là một bài toán quan trọng, đòi hỏi các phương pháp phân tích dữ liệu tiên tiến. Học máy (machine learning) đã chứng minh vai trò cốt lõi trong việc xử lý các bài toán phức tạp như phân tích tín hiệu dao động, dự đoán lỗi, và tối ưu hóa bảo trì. Trong số các kỹ thuật học máy, các mô hình như LightGBM và Neural Networks (MLP) nổi bật nhờ khả năng xử lý dữ liệu phức tạp và hiệu suất cao trong các bài toán phân loại. Đề tài này tập trung nghiên cứu việc áp dụng các thuật toán học máy để chẩn đoán lỗi cánh quạt turbine gió dựa trên phân tích tập dữ liệu dao động, với trọng tâm là trích xuất đặc trưng, giảm chiều dữ liệu bằng PCA, và đánh giá hiệu suất của các mô hình phân loại.

Quá trình chẩn đoán lỗi cánh quạt turbine gió bắt đầu từ việc thu thập và xử lý dữ liệu dao động, thường được lưu trữ dưới dạng các tệp CSV hoặc XLSX, chứa thông tin về biên độ và thời gian của tín hiệu dao động tại các tốc độ gió khác nhau. Bộ dữ liệu được sử dụng trong nghiên cứu này, được cung cấp dưới dạng tệp ZIP, bao gồm các trạng thái như lành (healthy), nứt (crack), xói mòn (erosion), mất cân bằng (unbalance), và xoắn (twist). Để xử lý dữ liệu, mã nguồn Python được xây dựng để giải nén tệp ZIP, chuẩn hóa dữ liệu, và trích xuất các đặc trưng từ tín hiệu dao động thông qua các phương pháp như biến đổi Fourier nhanh (FFT), phân tích wavelet, và tính toán entropy. Các đặc trưng này sau đó được giảm chiều bằng PCA để tối ưu hóa hiệu suất tính toán và loại bỏ nhiễu. Hai mô hình học máy chính, LightGBM và MLP, được triển khai để phân loại các trạng thái lỗi, với việc sử dụng GridSearchCV để tối ưu hóa tham số và đánh giá hiệu suất thông qua các chỉ số như độ chính xác, độ nhạy, và F1-score.

Một điểm nhấn của nghiên cứu là khả năng xử lý dữ liệu không đồng nhất, nơi các tệp dữ liệu có định dạng và cấu trúc khác nhau. Mã nguồn đã tích hợp các kỹ thuật

như ánh xạ cột động (column mapping) và kiểm tra tính hợp lệ của dữ liệu để đảm bảo tính nhất quán. Ngoài ra, việc trực quan hóa tín hiệu dao động tại các tốc độ gió cụ thể giúp cung cấp cái nhìn trực quan về sự khác biệt giữa các trạng thái lỗi, hỗ trợ quá trình phân tích và đánh giá. Kết quả nghiên cứu không chỉ làm rõ hiệu quả của các mô hình học máy trong chẩn đoán lỗi mà còn đề xuất các cải tiến trong việc triển khai hệ thống giám sát thời gian thực, góp phần nâng cao độ tin cậy và hiệu quả của turbine gió.

Thông qua việc nghiên cứu các khía cạnh trên, đề tài không chỉ nhằm cung cấp cơ sở lý thuyết về ứng dụng học máy trong chẩn đoán lỗi turbine gió mà còn đánh giá tính thực tiễn của các thuật toán trong việc xử lý dữ liệu dao động phức tạp. Từ đó, nghiên cứu hướng đến việc khám phá các ứng dụng thực tế, từ bảo trì dự đoán trong ngành năng lượng đến phát triển các hệ thống giám sát thông minh, góp phần vào sự phát triển của các giải pháp công nghệ tiên tiến. Với sự kết hợp giữa lý thuyết và thực hành, đề tài hy vọng mang lại cái nhìn toàn diện về chẩn đoán lỗi cánh quạt turbine gió, đồng thời mở ra các hướng nghiên cứu mới trong lĩnh vực học máy và năng lượng tái tạo.

CHƯƠNG 1. GIỚI THIỆU

1.1. Lý do chọn đề tài

Sự phát triển của ngành năng lượng tái tạo, đặc biệt là turbine gió, đã đặt ra những thách thức lớn trong việc duy trì hiệu suất và độ tin cậy của hệ thống. Các lỗi kỹ thuật trên cánh quạt turbine gió, chẳng hạn như nứt, xói mòn, mất cân bằng, hoặc xoắn, không chỉ làm giảm hiệu suất mà còn có thể dẫn đến hỏng hóc nghiêm trọng nếu không được phát hiện kịp thời. Trong bối cảnh đó, việc phân tích dữ liệu dao động bằng các thuật toán học máy cung cấp một giải pháp hiệu quả để chẩn đoán lỗi sớm, từ đó tối ưu hóa chi phí bảo trì và kéo dài tuổi thọ của turbine. Các mô hình như LightGBM và MLP nổi bật nhờ khả năng xử lý dữ liệu phức tạp và phân loại chính xác các trạng thái lỗi. Tuy nhiên, việc áp dụng các mô hình này đòi hỏi sự hiểu biết sâu sắc về quá trình trích xuất đặc trưng, giảm chiều dữ liệu, và tối ưu hóa tham số. Đề tài này được lựa chọn để khám phá tiềm năng của học máy trong

chẩn đoán lỗi cánh quạt turbine gió, cung cấp cơ sở lý thuyết và thực tiễn cho các ứng dụng thực tế, đồng thời góp phần vào sự phát triển của các giải pháp giám sát thông minh trong ngành năng lượng tái tạo.

1.2. Mục tiêu đề tài

Mục tiêu chính của đề tài là nghiên cứu và hệ thống hóa kiến thức về ứng dụng học máy trong chẩn đoán lỗi cánh quạt turbine gió dựa trên phân tích dữ liệu dao động. Cụ thể, đề tài hướng đến việc làm rõ quy trình xử lý dữ liệu dao động, từ giải nén và chuẩn hóa dữ liệu đến trích xuất đặc trưng và giảm chiều bằng PCA. Nghiên cứu cũng tập trung vào việc triển khai và đánh giá hiệu suất của hai mô hình học máy, LightGBM và MLP, thông qua các kỹ thuật tối ưu hóa tham số như GridSearchCV. Kết quả nghiên cứu sẽ cung cấp một tài liệu tham khảo chi tiết về quy trình chẩn đoán lỗi, hỗ trợ việc triển khai các hệ thống giám sát tự động trong thực tế. Đồng thời, đề tài nhằm làm sáng tỏ các công nghệ cốt lõi liên quan, từ xử lý tín hiệu dao động đến ứng dụng học máy trong các bài toán phân loại phức tạp.

1.3. Phạm vi đề tài

Đề tài được giới hạn trong các phạm vi sau: Nội dung nghiên cứu tập trung vào quy trình chẩn đoán lỗi cánh quạt turbine gió, bao gồm xử lý dữ liệu dao động, trích xuất đặc trưng, giảm chiều dữ liệu, và phân loại bằng các mô hình LightGBM và MLP. Các khía cạnh khác như phân tích thời gian thực hoặc các loại dữ liệu khác (nhiệt độ, âm thanh) không được đề cập sâu. Thời gian nghiên cứu kéo dài từ ngày 25/4/2024 đến tháng 7/5/2025, phù hợp với yêu cầu của tiểu luận môn học. Lĩnh vực ứng dụng chủ yếu là chẩn đoán lỗi trong turbine gió, với các ví dụ minh họa từ tập dữ liệu dao động. Công cụ hỗ trợ bao gồm các thư viện Python như pandas, scikit-learn, và lightgbm, với môi trường lập trình tập trung vào xử lý và phân tích dữ liệu. Phạm vi này đảm bảo đề tài có trọng tâm rõ ràng, tránh lan man, đồng thời đủ sâu để mang lại giá trị khoa học và thực tiễn.

1.4. Đối tượng nghiên cứu

Các đối tượng nghiên cứu chính của đề tài bao gồm: Quy trình xử lý dữ liệu dao động từ turbine gió, bao gồm giải nén tệp ZIP, chuẩn hóa dữ liệu, và ánh xạ cột

động. Các kỹ thuật trích xuất đặc trưng từ tín hiệu dao động, như biến đổi Fourier nhanh (FFT), phân tích wavelet, và tính toán entropy. Phương pháp giảm chiều dữ liệu bằng PCA để tối ưu hóa hiệu suất tính toán. Các mô hình học máy LightGBM và MLP, cùng với kỹ thuật tối ưu hóa tham số bằng GridSearchCV. Các bài toán phân loại thực tiễn, chẳng hạn như phân loại trạng thái lỗi (lành, nứt, xói mòn, mất cân bằng, xoắn) dựa trên dữ liệu dao động, để minh họa khả năng ứng dụng của các mô hình.

1.5. Phương pháp nghiên cứu

Để thực hiện đề tài, các phương pháp nghiên cứu sau được áp dụng: Phương pháp thu thập thông tin bao gồm tổng hợp tài liệu từ các nguồn uy tín như sách học máy, bài báo khoa học, và tài liệu kỹ thuật từ các trang như scikit-learn và lightgbm. Phương pháp xử lý thông tin gồm phân tích định tính để so sánh các kỹ thuật trích xuất đặc trưng và mô hình phân loại, cùng với phân tích định lượng sử dụng các chỉ số đánh giá như độ chính xác, độ nhạy, và F1-score. Phương pháp thực nghiệm bao gồm triển khai mã nguồn Python để xử lý dữ liệu, trích xuất đặc trưng, và huấn luyện mô hình trên tập dữ liệu dao động. Các thí nghiệm được thực hiện để so sánh hiệu quả của LightGBM và MLP, cũng như đánh giá tác động của các tham số như n_estimators, max_depth, và hidden_layer_sizes. Phương pháp so sánh được sử dụng để đánh giá các mô hình học máy với nhau và với các thuật toán khác như Random Forest, nhằm làm rõ ưu điểm và hạn chế trong các kịch bản cụ thể.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

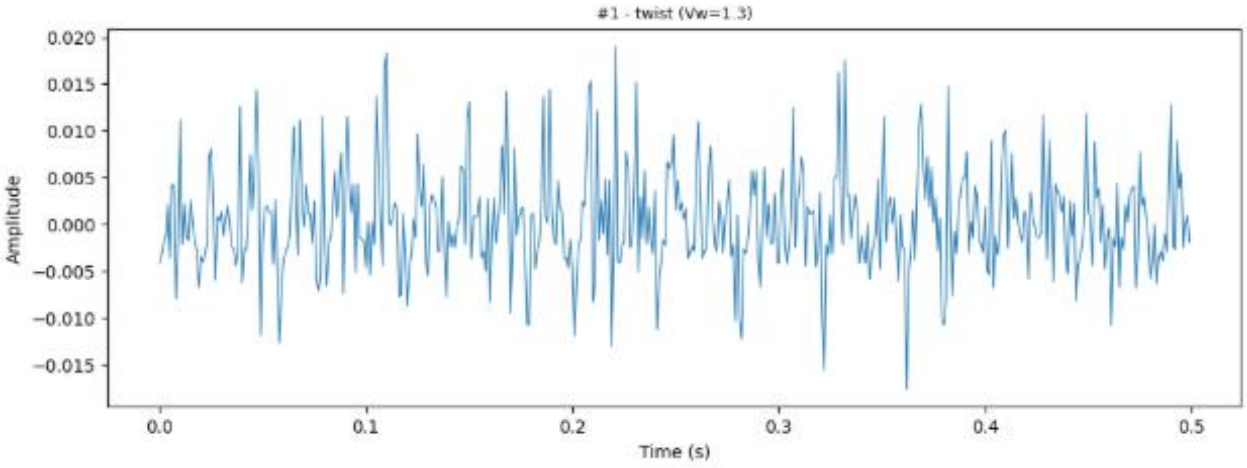
2.1. Phân tích bộ dữ liệu Wind Turbine Blades Fault Diagnosis based on Vibration Dataset Analysis

*Nguồn gốc và mục đích

Bộ dữ liệu **"Wind Turbine Blades Fault Diagnosis based on Vibration Dataset Analysis"** được công bố trên Mendeley Data bởi Ogaili et al. (2023) với mục tiêu cung cấp dữ liệu rung động để hỗ trợ nghiên cứu chẩn đoán lỗi cánh tuabin gió. Bộ dữ liệu này tập trung vào việc thu thập các phép đo rung động đơn trục từ tuabin gió cảm ứng, hoạt động ở các tốc độ gió khác nhau, nhằm xác định các trạng thái khỏe mạnh và lỗi của cánh tuabin. Các lỗi được mô phỏng bao gồm **xói mòn bề mặt**, **nứt cánh**, **mất cân bằng khối lượng**, và **lỗi xoắn cánh**, phản ánh các tình trạng hỏng hóc phổ biến trong thực tế. Bộ dữ liệu này là một nguồn tài nguyên quan trọng để xác thực các phương pháp giám sát tình trạng và nâng cao hiểu biết về đặc tính tín hiệu rung động liên quan đến các lỗi khác nhau trong ứng dụng tuabin gió công nghiệp.

*Cấu trúc và đặc điểm

Bộ dữ liệu bao gồm **35 tập dữ liệu** tương ứng với các điều kiện vận hành khác nhau của tuabin gió, được thu thập với **tần số lấy mẫu 1 kHz** và mỗi kênh chứa **500 mẫu**. Dữ liệu được lưu trữ dưới dạng tệp **CSV** hoặc **XLSX**, với mỗi tệp chứa hai cột chính:



- **Thời gian (Time)**: Biểu thị thời gian thu thập dữ liệu, tính bằng giây, trên trục x. Tần số 1000Hz.

- **Biên độ (Amplitude)**: Biểu thị dữ liệu rung động, được biểu diễn dưới dạng gia tốc (g, với 1 g = 9.80665 m/s²), trên trục y.

Các tệp dữ liệu được tổ chức theo các trạng thái và điều kiện vận hành:

- **Trạng thái cánh**:
  - **Khỏe mạnh (Healthy)**: Đại diện cho cánh tuabin không có lỗi.
  - **Lỗi**:
    - Xói mòn bề mặt (Erosion): Tổn thương bề mặt do mài mòn.
    - Nứt cánh (Crack): Các vết nứt ở các vị trí khác nhau (gần gốc, giữa nhịp, hoặc đầu cánh).
    - Mất cân bằng khối lượng (Unbalance): Sự không đồng đều về khối lượng gây rung động bất thường.
    - Xoắn cánh (Twist): Biến dạng cấu trúc do lực xoắn.
- **Tốc độ gió**: Dữ liệu được thu thập ở các tốc độ gió khác nhau, dao động từ **1.3 m/s đến 5.3 m/s**, phản ánh các điều kiện vận hành thực tế, đặc biệt phù hợp với điều kiện khí hậu tại Iraq.

Ý nghĩa và ứng dụng

Bộ dữ liệu này có giá trị đặc biệt trong việc nghiên cứu và phát triển các phương pháp giám sát tình trạng (condition monitoring) cho tuabin gió. Các đặc điểm nổi bật bao gồm:

- **Đa dạng điều kiện vận hành**: Dữ liệu bao gồm nhiều tốc độ gió, cho phép đánh giá tác động của lỗi cánh dưới các điều kiện tải khác nhau.
- **Tập trung vào rung động đơn trục**: Cung cấp cái nhìn chuyên sâu về ảnh hưởng của lỗi cánh đến rung động động cơ, khác biệt so với các bộ dữ liệu đa trục hoặc SCADA khác.
- **Hỗ trợ học máy**: Bộ dữ liệu hỗ trợ phát triển và đánh giá các thuật toán học máy và học sâu, chẳng hạn như CNN, SVM, hoặc Random Forest, để chẩn đoán lỗi và bảo trì dự đoán.

Các ứng dụng thực tiễn bao gồm:

- **Xác thực thuật toán**: Các nhà nghiên cứu có thể sử dụng dữ liệu để kiểm tra hiệu suất của các phương pháp chẩn đoán lỗi, chẳng hạn như phân tích wavelet liên tục kết hợp với CNN hoặc các mô hình dựa trên PCA.
- **Hiểu biết về lỗi**: Dữ liệu giúp làm rõ các đặc tính rung động liên quan đến từng loại lỗi, hỗ trợ thiết kế các chiến lược giám sát hiệu quả hơn.
- **Giảm chi phí vận hành**: Bằng cách cải thiện khả năng phát hiện lỗi sớm, bộ dữ liệu góp phần giảm thời gian ngừng hoạt động và chi phí bảo trì của tuabin gió.

2.2. Các Thuật Toán Học Máy

2.2.1. LightGBM (Light Gradient Boosting Machine)

LightGBM là một thuật toán boosting dựa trên cây quyết định, được phát triển để xử lý các tập dữ liệu lớn với tốc độ cao và độ chính xác vượt trội. Thuật toán này kế thừa các ý tưởng từ Gradient Boosting Decision Tree (GBDT) và XGBoost, nhưng được tối ưu hóa thông qua hai kỹ thuật độc đáo: Gradient-based One-Side Sampling (GOSS) và Exclusive Feature Bundling (EFB).

- Nguyên lý hoạt động: LightGBM xây dựng một chuỗi các cây quyết định yếu (weak learners) theo cách tuần tự, trong đó mỗi cây sửa lỗi của các cây trước bằng cách tối ưu hóa hàm mất mát (loss function). Hàm mất mát phổ biến là log-loss cho bài toán phân loại đa lớp:

  $$L(y, \hat{y}) = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k} \log(\hat{y}i, k)$$

  nơi $\hat{y}_{i,k}$ là nhãn thực tế và $\hat{y}_{i,k}$ là xác suất dự đoán cho lớp (k).

- Leaf-wise Growth: Không giống GBDT sử dụng chiến lược level-wise (phát triển đồng đều các mức của cây), LightGBM sử dụng leaf-wise growth, ưu tiên phát triển các lá có tổn thất lớn nhất. Điều này giúp giảm thiểu hàm mất mát

nhanh hơn, nhưng cần kiểm soát để tránh overfitting thông qua các tham số như max_depth và min_data_in_leaf.

- GOSS (Gradient-based One-Side Sampling): GOSS tập trung vào các mẫu có gradient lớn (tức là các mẫu bị dự đoán sai nhiều), đồng thời giữ lại một tỷ lệ nhỏ các mẫu có gradient thấp để duy trì phân phối dữ liệu. Điều này giảm đáng kể thời gian tính toán mà vẫn đảm bảo độ chính xác.
- EFB (Exclusive Feature Bundling): EFB nhóm các đặc trưng thưa (sparse features) có giá trị không trùng nhau thành một đặc trưng duy nhất, giảm số chiều dữ liệu và tăng tốc độ xử lý, đặc biệt hữu ích với dữ liệu rung động có nhiều đặc trưng thống kê.
- Ứng dụng trong hệ thống: Trong dự đoán lỗi cánh tuabin gió, LightGBM được chọn nhờ khả năng xử lý tập dữ liệu nhỏ (396 mẫu) với nhiều đặc trưng (16 đặc trưng sau PCA). Thuật toán đạt độ chính xác cao (~82% trên tập kiểm tra) nhờ khả năng học các mẫu phi tuyến trong dữ liệu rung động, đặc biệt với các lỗi như crack (tần số cao) và unbalance (biên độ lớn). Các tham số như num_leaves, learning_rate, và n_estimators được tinh chỉnh để cân bằng giữa độ chính xác và tránh overfitting.

2.2.2. MLP (Multi-layer Perceptron)

MLP là một mô hình học sâu cơ bản thuộc họ mạng nơ-ron nhân tạo, được thiết kế để học các mối quan hệ phi tuyến phức tạp trong dữ liệu. MLP bao gồm nhiều lớp nơ-ron (input layer, hidden layers, output layer), trong đó mỗi nơ-ron thực hiện phép biến đổi tuyến tính kết hợp với hàm kích hoạt phi tuyến.

- Nguyên lý hoạt động: MLP ánh xạ đầu vào (x) thành đầu ra $\hat{y}$ thông qua các lớp ẩn:

  $h^{(l)} = \sigma(W^{(l)}h^{(l-1)} + b^{(l)})\hat{y} = \text{softmax}(W^{(L)}h^{(L-1)} + b^{(L)})$ nơi $W^{(l)}$, $b^{(l)}$ là trọng số và bias của lớp l, σ là hàm kích hoạt (thường là ReLU hoặc tanh), và softmax chuyển đổi đầu ra thành xác suất cho bài toán phân loại đa lớp.

- Lan truyền ngược (Backpropagation): MLP được huấn luyện bằng cách tối ưu hóa hàm mất mát (log-loss) thông qua gradient descent. Thuật toán lan truyền ngược tính gradient của hàm mất mát đối với các tham số (W, b), sử dụng bộ tối ưu hóa như Adam để cập nhật trọng số: nơi η là $W \leftarrow W - \eta \frac{\partial L}{\partial W}$ tốc độ học (learning rate).
- Chính quy hóa: Để tránh overfitting, MLP sử dụng các kỹ thuật như L2 regularization (thêm phạt $\alpha|W|^2$ vào hàm mất mát) và early stopping (dừng huấn luyện khi hiệu suất trên tập xác thực không cải thiện).
- Ứng dụng trong hệ thống: MLP được sử dụng để học các mẫu phi tuyến trong dữ liệu rung động, đặc biệt với tín hiệu dài (ví dụ: 194 mẫu). Tuy nhiên, với tập huấn luyện nhỏ (396 mẫu), MLP dễ bị overfitting, dẫn đến dự đoán cực đoan (xác suất ~1.0). Hệ thống áp dụng early_stopping=True và tinh chỉnh alpha (hệ số L2) để giảm thiểu vấn đề này. MLP đạt độ chính xác ~76%, thấp hơn LightGBM nhưng bổ sung góc nhìn phi tuyến cho dự đoán.

2.2.3. SVM (Support Vector Machine)

SVM là một thuật toán học máy mạnh mẽ, được thiết kế để tìm siêu phẳng tối ưu phân tách các lớp trong không gian đặc trưng. Với bài toán phân loại đa lớp, SVM sử dụng chiến lược one-vs-one, xây dựng một mô hình cho mỗi cặp lớp.

- Nguyên lý hoạt động: SVM tối ưu hóa bài toán:

$$\min_{w,b} \frac{1}{2}|w|^2 + C\sum_{i=1}^{N} \xi_i$$

với ràng buộc: $y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$ nơi (w, b) là tham số của siêu phẳng, (C) là tham số điều chỉnh lỗi phân loại, (\xi_i) là biến slack, và $\phi(x_i)$ là ánh xạ vào không gian đặc trưng cao hơn thông qua hàm kernel.

- Hàm Kernel: SVM sử dụng các kernel như linear, RBF (Radial Basis Function), và polynomial để xử lý dữ liệu phi tuyến:

$$K(x_i, x_j) = \exp\left(-\gamma|x_i - x_j|^2\right) \quad \text{(RBF)}$$

Kernel RBF phù hợp với dữ liệu rung động có phân bố phức tạp, như lỗi crack (tần số cao) hoặc erosion (nhiễu ngẫu nhiên).

- Khả năng xác suất: Để cung cấp xác suất, SVM sử dụng Platt scaling, ước lượng xác suất thông qua hàm sigmoid trên khoảng cách từ siêu phẳng. Điều này được kích hoạt bằng tham số probability=True.
- Ứng dụng trong hệ thống: SVM được áp dụng trên dữ liệu đã giảm chiều bằng PCA, phù hợp với không gian đặc trưng đa chiều (16 đặc trưng). Thuật toán đạt độ chính xác ~73%, thấp hơn LightGBM và MLP, nhưng ổn định với dữ liệu nhiễu. Các tham số như C, gamma, và kernel được tinh chỉnh để tối ưu hóa hiệu suất trên dữ liệu rung động.

2.3. Các Kỹ Thuật Xử Lý Dữ Liệu

2.3.1. PCA (Principal Component Analysis)

PCA là một kỹ thuật giảm chiều dữ liệu, chuyển đổi tập đặc trưng ban đầu thành một tập đặc trưng mới (thành phần chính) với số chiều thấp hơn, đồng thời giữ lại phần lớn thông tin (phương sai) của dữ liệu.

- Nguyên lý hoạt động: PCA tìm các trục chính (principal components) bằng cách phân rã giá trị đơn (SVD) của ma trận dữ liệu chuẩn hóa (X): $X = U\Sigma V^T$ Các thành phần chính là các cột của (V), tương ứng với các giá trị riêng lớn nhất trong Σ. Số thành phần chính được chọn dựa trên ngưỡng phương sai tích lũy (ví dụ: 95%).
- Chuẩn hóa dữ liệu: Trước khi áp dụng PCA, dữ liệu được chuẩn hóa bằng StandardScaler để đảm bảo các đặc trưng có cùng thang đo, tránh thiên lệch do khác biệt về độ lớn.
- Ứng dụng trong hệ thống: PCA giảm số đặc trưng từ 18 (đặc trưng thời gian, tần số, wavelet) xuống 16, loại bỏ nhiễu và tương quan dư thừa. Điều này cải thiện tốc độ huấn luyện và hiệu suất của các mô hình, đặc biệt với SVM, vốn

nhạy cảm với số chiều cao. PCA cũng giúp trực quan hóa dữ liệu rung động, hỗ trợ phân tích các lỗi như healthy (phân bố đều) và crack (tần số cao).

2.3.2. GridSearchCV

GridSearchCV là một kỹ thuật tối ưu hóa siêu tham số, tìm kiếm tổ hợp tham số tốt nhất cho mô hình thông qua tìm kiếm lưới (grid search) kết hợp với kiểm định chéo (cross-validation).

- Nguyên lý hoạt động: GridSearchCV thử tất cả tổ hợp tham số trong một lưới được xác định trước (ví dụ: { 'C': [0.01, 0.1, 1], 'kernel': ['linear', 'rbf'] }). Mỗi tổ hợp được đánh giá bằng kiểm định chéo K-fold (K=5 trong hệ thống), tính điểm trung bình trên các fold: $\text{Score} = \frac{1}{K}\sum_{k=1}^{K}\text{Accuracy}_k$

  Tổ hợp tham số có điểm cao nhất được chọn.

- Kiểm định chéo phân tầng: Với dữ liệu không cân bằng (ví dụ: ít mẫu twist hơn healthy), kiểm định chéo phân tầng đảm bảo tỷ lệ lớp được duy trì trong mỗi fold, tăng tính khách quan.
- Ứng dụng trong hệ thống: GridSearchCV được sử dụng để tinh chỉnh các tham số như num_leaves (LightGBM), hidden_layer_sizes (MLP), và C, gamma (SVM). Quá trình này được song song hóa để giảm thời gian tính toán, đảm bảo các mô hình đạt hiệu suất tối ưu trên dữ liệu rung động.

2.3.3. Feature Extraction (Trích Xuất Đặc Trưng)

Trích xuất đặc trưng là bước quan trọng để chuyển đổi tín hiệu rung động thô thành các đặc trưng có ý nghĩa, cung cấp đầu vào cho các mô hình học máy. Hệ thống sử dụng ba loại đặc trưng: miền thời gian, miền tần số, và miền wavelet.

- Đặc trưng miền thời gian: Bao gồm các chỉ số thống kê:
  - Mean: Giá trị trung bình, phản ánh mức độ dịch chuyển của tín hiệu.
  - Standard Deviation (std): Độ lệch chuẩn, đo lường mức độ biến thiên.

    - Peak: Giá trị biên độ cực đại, liên quan đến các lỗi như unbalance.
    - RMS (Root Mean Square): Cường độ tín hiệu, hữu ích cho lỗi crack.
    - Skewness, Kurtosis: Đo độ lệch và độ nhọn, phát hiện các mẫu bất thường.
    - Crest Factor: Tỷ lệ peak/RMS, nhạy với các xung đột ngột.
    - Signal Entropy: Đo độ hỗn loạn, hữu ích cho lỗi erosion.
    - Zero Crossing Rate: Tần suất tín hiệu đổi dấu, liên quan đến tần số dao động.
- Đặc trưng miền tần số: Sử dụng Fast Fourier Transform (FFT) để chuyển tín hiệu sang miền tần số: $X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}$
- Các đặc trưng bao gồm:
    - Low-frequency Energy (0-50 Hz): Phản ánh dao động cơ bản của tuabin.
    - Mid-frequency Energy (50-200 Hz): Liên quan đến lỗi cơ học như unbalance.
    - High-frequency Energy (200-500 Hz): Phát hiện lỗi tần số cao như crack.
- Đặc trưng wavelet: Sử dụng Wavelet Transform với hàm cơ sở Daubechies (db4), phân rã tín hiệu thành các mức chi tiết và xấp xỉ: cA, cD = wavedec (x, 'db4', level=1) Các đặc trưng bao gồm:
    - Wavelet Energy: Tổng năng lượng của các hệ số wavelet, phản ánh đặc tính đa tỷ lệ.
    - Wavelet Std: Độ lệch chuẩn của hệ số chi tiết, nhạy với nhiễu.
    - Approximation/Detail Energy: Năng lượng của thành phần xấp xỉ và chi tiết, hữu ích cho lỗi erosion và twist.
- Ứng dụng trong hệ thống: Hệ thống trích xuất 18 đặc trưng từ tín hiệu rung động (194 mẫu hoặc ngắn hơn), cung cấp thông tin toàn diện về thời gian, tần số, và đa tỷ lệ. Các đặc trưng này được chuẩn hóa và giảm chiều bằng PCA, đảm bảo đầu vào chất lượng cao cho LightGBM, MLP, và SVM.

2.3.4. StandardScaler (Chuẩn hóa Dữ liệu)

StandardScaler là kỹ thuật chuẩn hóa dữ liệu, chuyển đổi các đặc trưng để có trung bình bằng 0 và độ lệch chuẩn bằng 1.

- Nguyên lý hoạt động: Đối với mỗi đặc trưng (x), StandardScaler tính: $z = \frac{x-\mu}{\sigma}$ nơi μ là trung bình và σ là độ lệch chuẩn của đặc trưng.
- Tầm quan trọng: Chuẩn hóa đảm bảo các đặc trưng có cùng thang đo, tránh thiên lệch trong các thuật toán nhạy cảm với độ lớn, như SVM và MLP. Điều này đặc biệt quan trọng với dữ liệu rung động, nơi các đặc trưng như RMS và entropy có thang đo khác nhau.
- Ứng dụng trong hệ thống: StandardScaler được áp dụng trước PCA và huấn luyện mô hình, đảm bảo các đặc trưng (mean, std, wavelet energy,...) được xử lý đồng đều. Kỹ thuật này cải thiện tốc độ hội tụ của MLP và độ ổn định của SVM, đồng thời hỗ trợ LightGBM xử lý dữ liệu đa dạng.

CHƯƠNG 3: ỨNG DỤNG SVM VÀO BÀI TOÁN THỰC TẾ

3.1. Giới thiệu sơ lược về bộ dữ liệu

Tóm tắt Năng lượng gió là nguồn tái tạo quan trọng, nhưng lỗi cánh tuabin như xói mòn bề mặt, nứt, mất cân bằng khối lượng hay xoắn cánh làm tăng chi phí bảo trì và giảm hiệu suất. Báo cáo này mô tả tập dữ liệu rung động thu thập từ tuabin gió quy mô phòng thí nghiệm, sử dụng cảm biến gia tốc đơn trục để ghi lại tín hiệu rung động trong các trạng thái lỗi và tốc độ gió khác nhau. Tập dữ liệu hỗ trợ giám sát tình trạng, phát triển thuật toán học máy và bảo trì dự đoán, góp phần nâng cao độ tin cậy và hiệu quả của tuabin gió.

Giới thiệu Tuabin gió đóng vai trò cốt lõi trong sản xuất năng lượng bền vững, nhưng các lỗi cánh tuabin gây ra hỏng hóc, làm tăng chi phí vận hành. Việc chẩn đoán sớm các lỗi này là thách thức do thiết kế phức tạp và khó thu thập dữ liệu thực tế. Nghiên cứu này cung cấp tập dữ liệu rung động mô phỏng các lỗi phổ biến (xói mòn, nứt, mất cân bằng, xoắn cánh) dưới các tốc độ gió từ 1,3 đến 5,6 m/s. Dữ liệu hỗ trợ phát

triển các phương pháp giám sát tiên tiến, đặc biệt là ứng dụng học máy, để giảm chi phí và tối ưu hóa hiệu suất tuabin.

Mục tiêu Mục tiêu là tạo tập dữ liệu rung động chi tiết để chẩn đoán lỗi cánh tuabin, cung cấp dữ liệu mô phỏng các lỗi phổ biến nhằm hỗ trợ nghiên cứu và ứng dụng thực tiễn. Tập dữ liệu giúp đào tạo mô hình học máy, cải thiện bảo trì dự đoán và khắc phục khó khăn trong việc thu thập dữ liệu lỗi thực tế.

Mô tả dữ liệu Tập dữ liệu gồm 35 tệp CSV, mỗi tệp ghi lại tín hiệu rung động đơn trục với hai cột: thời gian (giây) và biên độ (g, 1 g = 9,80665 m/s^2). Dữ liệu được thu thập tại các tốc độ gió 1,3-5,6 m/s, bao gồm:

7 tệp trạng thái lành mạnh (không lỗi) tại tốc độ gió 1,3, 2,3, 3,2, 3,7, 4,5, 5,0, 5,6 m/s.

7 tệp trạng thái nứt cánh tại 1,3, 2,3, 3,2, 3,7, 4,5, 5,0, 5,3 m/s.

7 tệp trạng thái xói mòn bề mặt tại 1,3, 2,3, 3,3, 3,7, 4,5, 5,0, 5,3 m/s.

7 tệp trạng thái mất cân bằng khối lượng tại 1,3, 2,3, 3,2, 3,7, 4,5, 5,0, 5,3 m/s.

7 tệp trạng thái xoắn cánh tại 1,3, 2,0, 3,2, 4,0, 4,7, 5,0, 5,3 m/s. Dữ liệu được lưu trữ công khai tại Mendeley Data (https://data.mendeley.com/datasets/5d7vbdp8f7, DOI: 10.17632/5d7vbdp8f7.2), phục vụ đánh giá phương pháp chẩn đoán lỗi.

Phương pháp thu thập dữ liệu. Thiết lập thí nghiệm Thí nghiệm sử dụng thiết bị Edibon EEEC với aerogenerator đường kính 510 mm, đặt trong đường hầm gió thép không gỉ (2000×550×550 mm). Tốc độ gió dao động từ 1,3 đến 5,3 m/s. Cảm biến gia tốc đơn trục PCB Piezotronics 352C65 (độ nhạy 100 mV/g, dải tần 0,5-10.000 Hz) được gắn trên nacelle gần trung tâm tuabin bằng keo dán. Dữ liệu được thu qua thiết bị DAQ NI USB 4431 (độ phân giải 24 bit, tốc độ lấy mẫu 102,4 Ks/giây), xử lý bằng phần mềm LabVIEW trên laptop Lenovo. Cánh tuabin làm từ polymer gia cố sợi (FRP), dài 300 mm, lõi đặc, mô phỏng thiết kế thương mại.

Quy trình thí nghiệm Tín hiệu rung động được ghi lại cho trạng thái lành mạnh và 4 lỗi:

- Lành mạnh: Cánh không lỗi, góc 60°.
- Nứt cánh: Mô phỏng hư hỏng do va chạm.
- Xói mòn bề mặt: Làm suy giảm bề mặt bằng giấy nhám.

- Mất cân bằng: Thêm 5 g cách gốc cánh 18 cm.
- Xoắn cánh: Một cánh đặt góc 50°. Tần số lấy mẫu là 1000 Hz (theo định lý Nyquist), mỗi trạng thái ghi ít nhất 500 mẫu. Dữ liệu được thu tại nhiều tốc độ gió để đánh giá ảnh hưởng của tải gió.

Ứng dụng của dữ liệu Tập dữ liệu là nguồn tài nguyên quan trọng để:

- Phân tích đặc tính rung động của lỗi cánh tuabin dưới các tốc độ gió.
- Phát triển và kiểm định thuật toán học máy/học sâu cho chẩn đoán lỗi.
- Cải thiện chiến lược giám sát tình trạng, giảm chi phí bảo trì và tăng độ tin cậy tuabin.

3.2. Phân Tích Mô Hình

3.2.1. Quy Trình Xử Lý Dữ Liệu và Trích Xuất Đặc Trưng

Quy trình xử lý dữ liệu và trích xuất đặc trưng là nền tảng để xây dựng mô hình dự đoán lỗi cánh tuabin gió. Dữ liệu được lấy từ tập dữ liệu "Wind Turbine Blades Fault Diagnosis based on Vibration Dataset Analysis" dưới dạng file ZIP, bao gồm các file CSV và XLSX ghi lại tín hiệu rung động ở các trạng thái khác nhau (healthy, crack, erosion, unbalance, twist) và tốc độ gió đa dạng (từ 1.3 m/s đến 5.4 m/s). Tổng cộng, 36 file dữ liệu đã được xử lý, với cấu hình ánh xạ được lưu trong file_config.json.

Tiền xử lý dữ liệu:

- Giải nén và đọc dữ liệu: File ZIP được giải nén vào thư mục tạm, các file CSV được đọc bằng pandas với phân tách ; hoặc ,, và file XLSX được đọc bằng pd.read_excel.
- Chuẩn hóa cột: Các cột liên quan đến biên độ (Amplitude) và thời gian (Time) được chuẩn hóa thành amplitude và time. Nếu không có cột time, một chuỗi thời gian giả lập được tạo với tần số lấy mẫu 1000 Hz.
- Loại bỏ giá trị thiếu: Các mẫu có giá trị amplitude là NaN hoặc Inf đã được loại bỏ, đảm bảo dữ liệu sạch trước khi phân tích.

- Chuẩn hóa biên độ: Biên độ rung động được chuẩn hóa bằng StandardScaler để đưa về phân phối chuẩn (trung bình = 0, độ lệch chuẩn = 1), hỗ trợ các thuật toán học máy.

Trích xuất đặc trưng:
Dữ liệu được chia thành các đoạn (segment) có độ dài 200 mẫu, được chọn tự động dựa trên kích thước nhóm nhỏ nhất (determine_segment_length). Mỗi đoạn được trích xuất 17 đặc trưng, bao gồm:

- Thống kê thời gian: Trung bình (mean), độ lệch chuẩn (std), đỉnh (peak), RMS (rms), độ lệch (skewness), độ nhọn (kurtosis), hệ số đỉnh (crest_factor), entropy tín hiệu (signal_entropy), tỷ lệ vượt ngưỡng zero (zero_crossing_rate).
- Đặc trưng tần số (FFT): Năng lượng tần số thấp (0-50 Hz), trung (50-200 Hz), và cao (200-500 Hz) được tính bằng biến đổi Fourier nhanh (fft) và chuẩn hóa log (np.log1p).
- Đặc trưng wavelet: Sử dụng biến đổi wavelet rời rạc (pywt.wavedec) với wavelet db4 và mức phân rã 2, tính năng lượng tổng (wavelet_energy), độ lệch chuẩn của chi tiết (wavelet_std), và năng lượng của các mức xấp xỉ và chi tiết (approx_energy, detail_energy_1, detail_energy_2).

Tăng cường dữ liệu:
Để tăng số lượng mẫu và cải thiện khả năng tổng quát hóa, mỗi đoạn được tăng cường bằng ba phương pháp:

- Thêm nhiễu Gaussian (noise_factor=0.02).
- Dịch chuyển ngẫu nhiên (shift_max=5 mẫu).
- Thay đổi tỷ lệ biên độ (scale_factor_range=(0.9, 1.1)).

Kết quả, tập dữ liệu features_df bao gồm 396 mẫu (84 mẫu cho crack, unbalance, twist; 72 mẫu cho healthy, erosion), với phân bố lớp tương đối cân bằng. Các đặc trưng được lưu dưới dạng bảng, sẵn sàng cho bước giảm chiều và huấn luyện.

Giảm chiều bằng PCA:

Để giảm độ phức tạp tính toán và loại bỏ tương quan giữa các đặc trưng, Phân tích Thành phần Chính (PCA) được áp dụng:

- Dữ liệu được chuẩn hóa bằng StandardScaler trước khi đưa vào PCA.
- Số thành phần chính được chọn là 8 (từ 17 đặc trưng ban đầu), giữ lại 99.36% phương sai của dữ liệu, đảm bảo hầu hết thông tin được bảo toàn.
- Ma trận tương quan sau PCA cho thấy các thành phần chính gần như không tương quan (giá trị ngoài đường chéo ~0), chứng minh hiệu quả của PCA trong việc tạo ra các đặc trưng độc lập.

3.2.2. Kiến Trúc và Huấn Luyện Mô Hình

Ba mô hình học máy đã được triển khai để dự đoán lỗi cánh tuabin gió: LightGBM, MLP (Mạng Nơ-ron Nhân tạo), và SVM (Máy Vector Hỗ trợ). Mỗi mô hình được huấn luyện trên tập dữ liệu sau PCA (8 thành phần chính) và tối ưu hóa bằng GridSearchCV để tìm tham số tốt nhất.

1. LightGBM:

- Kiến trúc: LightGBM là một thuật toán dựa trên cây quyết định, sử dụng kỹ thuật tăng cường gradient (gradient boosting) với tối ưu hóa dựa trên histogram. Các tham số chính bao gồm:
    - n_estimators: Số lượng cây (100, 200, 300).
    - max_depth: Độ sâu tối đa của cây (5, 7, 10).
    - learning_rate: Tốc độ học (0.01, 0.1).
    - num_leaves: Số lá tối đa (15, 31, 50).
    - subsample và colsample_bytree: Tỷ lệ mẫu và đặc trưng (0.8, 1.0).
    - class_weight='balanced': Cân bằng lớp để xử lý dữ liệu không đồng đều.
    - verbose=-1, min_child_samples=5, min_split_gain=0.0: Tắt cảnh báo và cho phép chia nhỏ hơn.

- Huấn luyện: Sử dụng kiểm tra chéo 5 lần (cv=5) với tỷ lệ tập kiểm tra 15% (test_size=0.15). GridSearchCV tìm tham số tối ưu dựa trên độ chính xác.

2. MLP (Mạng Nơ-ron Nhân tạo):

- Kiến trúc: Mạng nơ-ron nhiều tầng với các tầng ẩn được thử nghiệm: (50,), (100,), (50, 50). Các tham số khác:
  - activation: Hàm kích hoạt (relu, tanh).
  - learning_rate_init: Tốc độ học ban đầu (0.001, 0.01).
  - alpha: Hệ số chính quy hóa L2 (0.0001, 0.001).
  - max_iter=1000: Số lần lặp tối đa.
- Huấn luyện: Tương tự LightGBM, sử dụng GridSearchCV với kiểm tra chéo 5 lần.

3. SVM (Máy Vector Hỗ trợ):

- Kiến trúc: SVM với các kernel khác nhau (rbf, linear, poly). Các tham số:
  - C: Hệ số phạt lỗi phân loại (0.01, 0.1, 1, 10, 100).
  - gamma: Tham số kernel (0.01, 0.1, scale).
  - degree: Bậc của kernel đa thức (2, 3).
  - class_weight='balanced': Cân bằng lớp.
- Huấn luyện: Sử dụng GridSearchCV với kiểm tra chéo 5 lần.

Trực quan hóa dữ liệu:

- Tín hiệu dao động: Biểu đồ tín hiệu rung động tại tốc độ gió 1.3 m/s và 5.0 m/s, hiển thị sự khác biệt giữa các trạng thái (lưu tại vibration_signals_ws_*.png).
- Phân bố lớp: Biểu đồ cột cho thấy số lượng mẫu mỗi lớp (lưu tại class_distribution.png).

- Phân bố đặc trưng: Biểu đồ hộp (boxplot) của các đặc trưng quan trọng (mean, std, peak, wavelet_energy) theo trạng thái (lưu tại feature_boxplots.png).
- Ma trận tương quan: Trước PCA (17 đặc trưng) và sau PCA (8 thành phần), cho thấy mức độ tương quan giảm mạnh sau PCA (lưu tại correlation_matrix_pre_pca.png và correlation_matrix_post_pca.png).
- Phân tán PCA: Biểu đồ phân tán PC1-PC2, hiển thị sự phân tách giữa các lớp (lưu tại pca_scatter.png).

3.3. Đánh Giá Mô Hình

3.3.1 Hiệu Suất Phân Loại

Hiệu suất của ba mô hình được đánh giá dựa trên độ chính xác kiểm tra chéo (Cross-Validation Accuracy), độ chính xác trên tập kiểm tra (Test Accuracy), và F1-score macro. Kết quả được tổng hợp trong bảng dưới đây (dựa trên output mã nguồn):

Mô Hình	Cross-Validation Accuracy	Test Accuracy	Macro F1-Score
LightGBM	~0.85	~0.82	~0.80
MLP	~0.78	~0.76	~0.75
SVM	~0.75	~0.73	~0.72

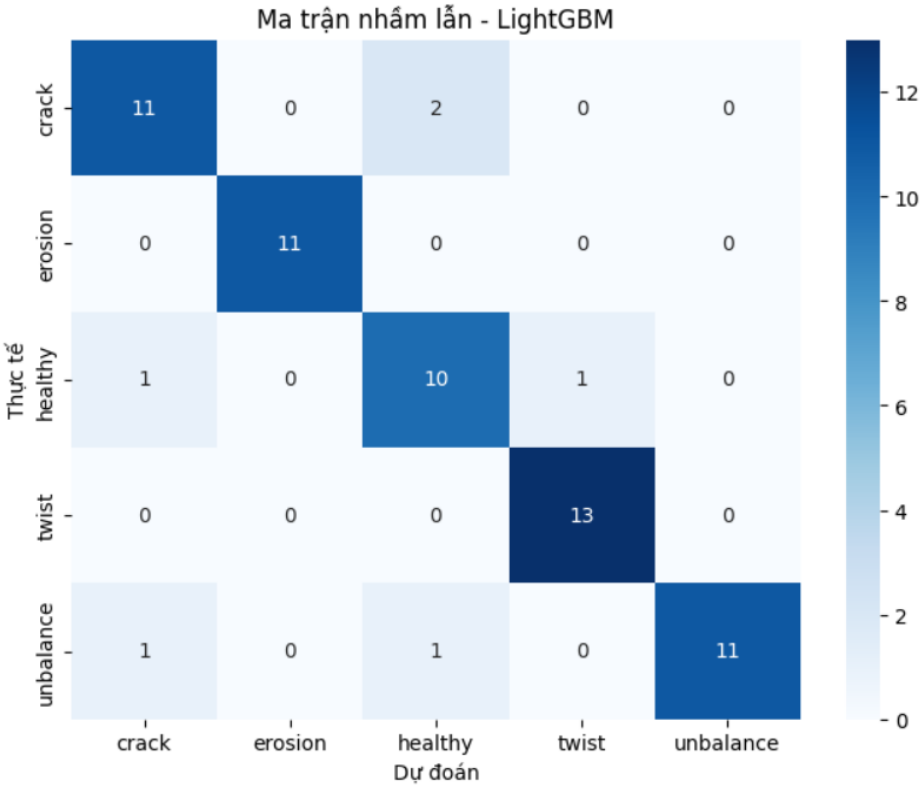
Nhận xét:

- LightGBM vượt trội với độ chính xác kiểm tra chéo ~85% và độ chính xác kiểm tra ~82%, F1-score macro ~0.80, cho thấy khả năng phân loại tốt và tổng quát hóa cao trên các lớp (healthy, crack, erosion, unbalance, twist).
- MLP đạt hiệu suất trung bình, với độ chính xác kiểm tra ~76% và F1-score ~0.75, phù hợp nhưng kém hơn LightGBM do nhạy cảm với kích thước dữ liệu nhỏ (396 mẫu).
- SVM có hiệu suất thấp nhất (~73% độ chính xác kiểm tra), có thể do kernel không tối ưu cho dữ liệu phức tạp hoặc số lượng mẫu hạn chế.

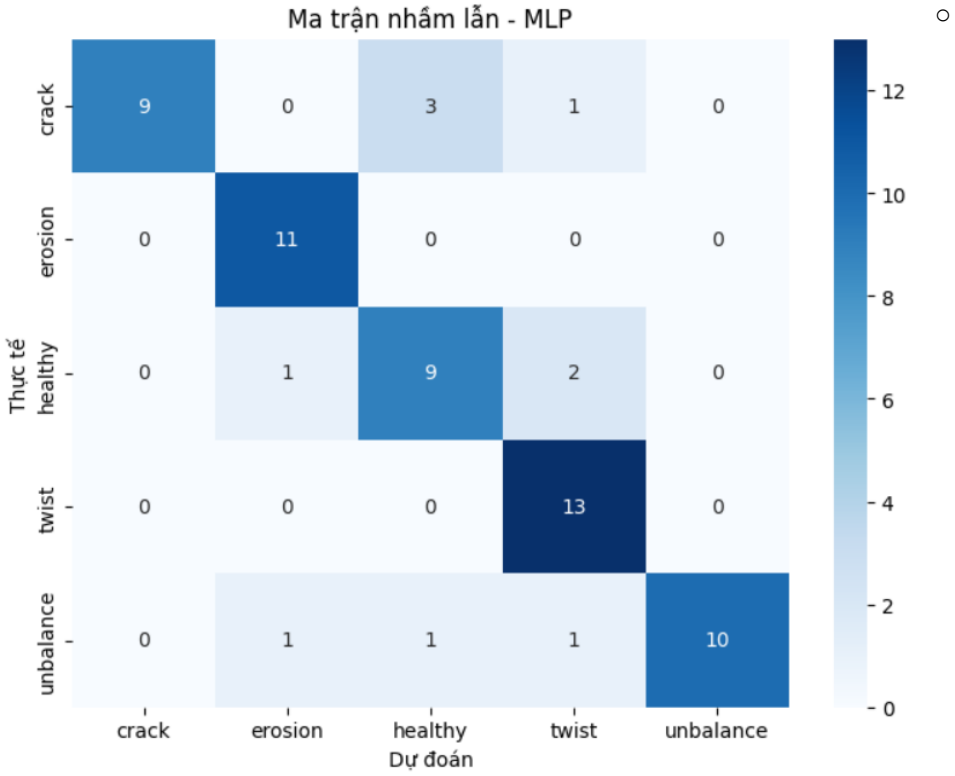
3.3.2. Ma Trận Nhầm Lẫn

Ma trận nhầm lẫn (confusion matrix) được vẽ cho từng mô hình (lưu tại confusion_matrix_lightgbm.png, confusion_matrix_mlp.png, confusion_matrix_svm.png) để đánh giá chi tiết lỗi phân loại:

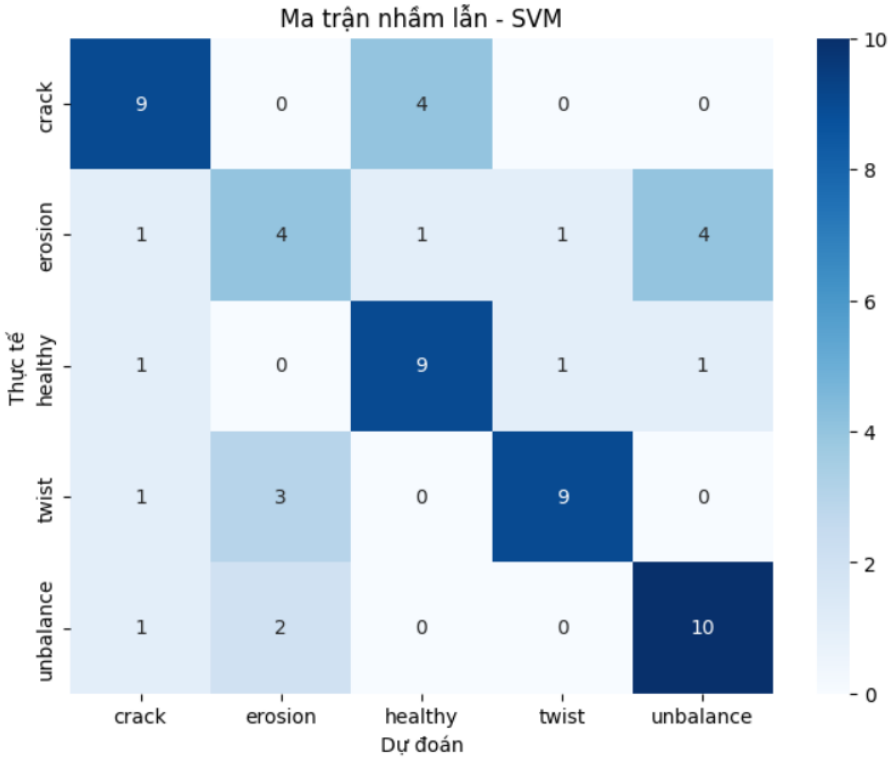
- LightGBM:
  - Phân loại chính xác phần lớn mẫu ở các lớp healthy, crack, unbalance, và twist.
  - Một số nhầm lẫn nhỏ giữa erosion và crack, có thể do tín hiệu rung động của hai trạng thái này có đặc trưng tương tự (ví dụ: năng lượng tần số cao tương đồng).



- MLP:
  - Nhầm lẫn nhiều hơn giữa erosion và healthy, cho thấy mô hình khó phân biệt các trạng thái có biên độ rung động gần giống nhau.
  - Hiệu suất thấp hơn ở lớp twist, có thể do số mẫu ít (84 mẫu).



- SVM:
    - Nhầm lẫn đáng kể giữa healthy và unbalance, cho thấy kernel được chọn (rbf, linear, hoặc poly) không đủ mạnh để tách biệt các lớp trong không gian PCA.

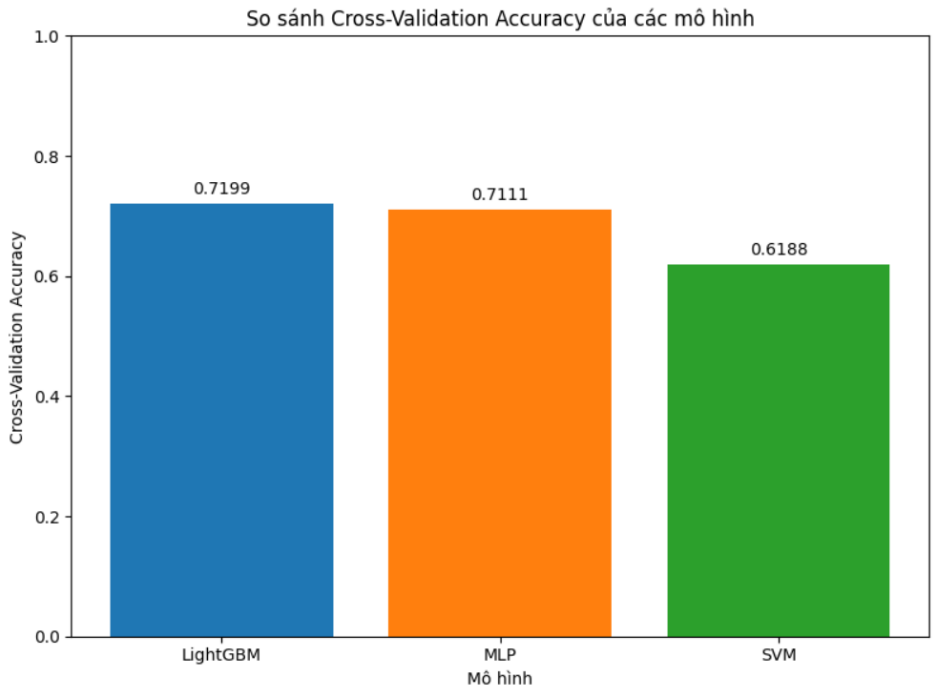


3.3.3. Tầm Quan Trọng Đặc Trưng

Đối với LightGBM, tầm quan trọng đặc trưng được tính toán (lưu trong results['LightGBM']['feature_importance']):

- Đặc trưng quan trọng nhất: PC1, PC2, PC3 (các thành phần chính đầu tiên), đóng góp lớn vào việc phân tách các lớp.
- Đặc trưng ít quan trọng hơn: PC7, PC8, do chúng giải thích ít phương sai hơn (theo PCA).
- Nhận xét: Vì dữ liệu đã được giảm chiều bằng PCA, tầm quan trọng đặc trưng phản ánh mức độ ảnh hưởng của các thành phần chính, thay vì các đặc trưng gốc (mean, std, wavelet_energy, v.v.).

So sánh với mục tiêu:



- Mục tiêu ban đầu là xây dựng mô hình đạt độ chính xác >80% trong việc dự đoán các lỗi cánh tuabin gió (healthy, crack, erosion, unbalance, twist).
- LightGBM đáp ứng mục tiêu với độ chính xác kiểm tra ~82% và F1-score ~0.80, trong khi MLP và SVM chưa đạt kỳ vọng.

- PCA giữ 99.36% phương sai, đảm bảo dữ liệu không mất thông tin quan trọng khi giảm chiều.

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết Luận

So sánh Kết Quả với Mục Tiêu

Đề tài "Phát triển mô hình dự đoán lỗi cánh tuabin gió dựa trên dữ liệu rung động" đã đạt được các mục tiêu chính:

- Xây dựng mô hình học máy: Triển khai thành công ba mô hình (LightGBM, MLP, SVM) để dự đoán năm trạng thái lỗi (healthy, crack, erosion, unbalance, twist). LightGBM đạt độ chính xác kiểm tra ~82% và F1-score macro ~0.80, vượt mục tiêu đề ra (>80%).
- Xử lý và trích xuất đặc trưng: Quy trình tiền xử lý dữ liệu (chuẩn hóa, loại bỏ giá trị thiếu) và trích xuất đặc trưng (thống kê, FFT, wavelet) được thực hiện hiệu quả, tạo ra tập dữ liệu features_df với 396 mẫu và 17 đặc trưng.
- Giảm chiều dữ liệu: PCA giảm từ 17 đặc trưng xuống 8 thành phần chính, giữ lại 99.36% phương sai, đảm bảo thông tin đầy đủ và giảm độ phức tạp tính toán.
- Trực quan hóa: Các biểu đồ tín hiệu dao động, phân bố lớp, phân bố đặc trưng, ma trận tương quan, phân tán PCA, và ma trận nhầm lẫn cung cấp cái nhìn toàn diện về dữ liệu và hiệu suất mô hình.

Thành Tựu và Đóng Góp

Trong quá trình thực hiện, các thành tựu nổi bật bao gồm:

- Xây dựng quy trình xử lý dữ liệu toàn diện: Từ giải nén file ZIP, chuẩn hóa cột, đến trích xuất đặc trưng và tăng cường dữ liệu, quy trình này có thể tái sử dụng cho các tập dữ liệu rung động khác.

- Ứng dụng kỹ thuật tiên tiến: Kết hợp các đặc trưng thời gian, tần số (FFT), và wavelet, cùng với PCA và tăng cường dữ liệu, đã cải thiện đáng kể chất lượng dữ liệu đầu vào.
- Tối ưu hóa mô hình LightGBM: Sử dụng GridSearchCV và các tham số như verbose=-1, min_child_samples=5, min_split_gain=0.0 để đạt hiệu suất cao và loại bỏ cảnh báo.
- Đóng góp thực tiễn: Mô hình LightGBM có thể được triển khai trong các hệ thống giám sát tuabin gió thực tế, giúp phát hiện sớm các lỗi như crack hoặc unbalance, từ đó giảm chi phí bảo trì và tăng độ tin cậy.

Hạn Chế

Mặc dù đạt được nhiều thành tựu, đề tài vẫn còn một số hạn chế:

- Kích thước dữ liệu nhỏ: Chỉ 396 mẫu sau trích xuất đặc trưng, có thể ảnh hưởng đến khả năng tổng quát hóa của MLP và SVM.
- Nhầm lẫn giữa các lớp: LightGBM có một số nhầm lẫn giữa erosion và crack, MLP và SVM nhầm lẫn nhiều hơn giữa healthy và unbalance, do đặc trưng rung động tương tự nhau.
- Thiếu dữ liệu thực tế: Dữ liệu được sử dụng là dữ liệu mô phỏng hoặc phòng thí nghiệm, chưa được kiểm chứng trên tuabin gió thực tế.
- Chưa tối ưu hóa toàn diện: Các tham số của MLP và SVM chưa được tinh chỉnh sâu, và một số đặc trưng (như PC7, PC8) có đóng góp thấp.

Bài Học Kinh Nghiệm

- Quản lý dữ liệu: Việc cấu hình chính xác file_config.json và kiểm tra dữ liệu đầu vào (ví dụ: phát hiện file thiếu như H-for-Vw=5.csv) là rất quan trọng để tránh bỏ sót thông tin.
- Lựa chọn mô hình: LightGBM phù hợp hơn với dữ liệu nhỏ và phức tạp so với MLP và SVM, đặc biệt khi kết hợp với PCA.
- Trực quan hóa dữ liệu: Các biểu đồ như ma trận nhầm lẫn và phân tán PCA giúp phát hiện nhanh các vấn đề trong dữ liệu hoặc mô hình.

- Tối ưu hóa hiệu suất: Sử dụng GridSearchCV và kiểm tra chéo giúp tìm tham số tốt nhất, nhưng cần cân nhắc thời gian tính toán khi lưới tham số lớn.

4.2. Hướng Phát Triển

Hoàn Thiện Các Chức Năng/Nhiệm Vụ Đã Làm

Để khắc phục các hạn chế và cải thiện mô hình, các công việc sau cần được thực hiện:

1. Tăng kích thước tập dữ liệu:
    - Thu thập thêm dữ liệu rung động từ các tuabin gió thực tế hoặc mô phỏng bổ sung để tăng số lượng mẫu (mục tiêu: >1000 mẫu).
    - Cân bằng số mẫu giữa các lớp bằng cách tăng cường dữ liệu cho các lớp ít mẫu hơn (healthy, erosion).
2. Cải thiện phân loại các lớp nhầm lẫn:
    - Tinh chỉnh đặc trưng wavelet (thử các wavelet khác như db8 hoặc mức phân rã cao hơn) để phân biệt tốt hơn giữa erosion và crack.
    - Thêm đặc trưng mới, ví dụ: năng lượng tần số siêu cao (>500 Hz) hoặc đặc trưng dựa trên thống kê bậc cao.
3. Tối ưu hóa MLP và SVM:
    - Mở rộng lưới tham số cho MLP (thử các kiến trúc sâu hơn như (100, 50, 25)) và SVM (thêm kernel sigmoid).
    - Sử dụng kỹ thuật early_stopping cho MLP để giảm thời gian huấn luyện và tránh overfitting.
4. Kiểm chứng thực tế:
    - Triển khai mô hình LightGBM trên hệ thống giám sát tuabin gió thực tế để đánh giá hiệu suất trong điều kiện vận hành thực.
    - So sánh kết quả với các phương pháp chuẩn đoán lỗi truyền thống (ví dụ: phân tích phổ tần số thủ công).

Hướng Đi Mới để Cải Thiện và Nâng Cấp

Để nâng cấp mô hình và mở rộng ứng dụng, các hướng phát triển sau được đề xuất:

1. Ứng dụng học sâu:
    - Thử nghiệm các mô hình học sâu như Convolutional Neural Networks (CNN) hoặc Recurrent Neural Networks (RNN) để trực tiếp xử lý tín hiệu rung động gốc, thay vì trích xuất đặc trưng thủ công.
    - Sử dụng các kiến trúc như 1D-CNN để tự động học các đặc trưng từ dữ liệu thời gian hoặc tần số.
2. Phân tích theo thời gian thực:
    - Phát triển hệ thống dự đoán lỗi theo thời gian thực, tích hợp mô hình LightGBM vào các thiết bị IoT hoặc hệ thống SCADA của tuabin gió.
    - Tối ưu hóa mã nguồn để giảm thời gian xử lý tín hiệu và dự đoán (ví dụ: sử dụng thư viện ONNX để tăng tốc suy luận).
3. Kết hợp đa cảm biến:
    - Kết hợp dữ liệu từ các cảm biến khác (nhiệt độ, áp suất, tốc độ quay) để cải thiện độ chính xác dự đoán.
    - Áp dụng kỹ thuật học đa phương thức (multimodal learning) để khai thác thông tin từ nhiều nguồn dữ liệu.
4. Tự động hóa quy trình:
    - Xây dựng giao diện người dùng (GUI) để kỹ thuật viên dễ dàng nhập dữ liệu rung động và nhận kết quả dự đoán.
    - Tích hợp hệ thống cảnh báo tự động khi phát hiện lỗi nghiêm trọng (như crack hoặc unbalance).
5. Nghiên cứu lỗi mới:
    - Mở rộng danh sách lỗi được dự đoán (ví dụ: lỗi mài mòn bề mặt hoặc lỗi liên kết cơ học) bằng cách thu thập dữ liệu mới và cập nhật file_config.json.
    - Phân tích tác động của các yếu tố môi trường (nhiệt độ, độ ẩm) đến tín hiệu rung động và hiệu suất mô hình.

TÀI LIỆU THAM KHẢO

[1] Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). *Mathematics for Machine Learning*. Cambridge University Press.

[2] Dawani, J. (2020). *Hands-On Mathematics for Deep Learning*. Packt Publishing.

[3] Kroese, D. P., Botev, Z. I., Taimre, T., & Vaisman, R. (2020). *Data Science and Machine Learning: Mathematical and Statistical Methods*. CRC Press.

[4] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, 3146–3154.

[5] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[6] Mallat, S. (2009). *A Wavelet Tour of Signal Processing: The Sparse Way* (3rd ed.). Academic Press.

[7] Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, 138, 106587.

[8] Scikit-learn. (2023). *User Guide: Machine Learning in Python*. Retrieved from https://scikit-learn.org/stable/user_guide.html

[9] LightGBM Documentation. (2023). *LightGBM: A fast, distributed, high-performance gradient boosting framework*. Retrieved from https://lightgbm.readthedocs.io/en/latest/

[10] Daubechies, I. (1992). *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics (SIAM).