HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

---




**Vũ Thị Hợp**

# NGHIÊN CỨU ỨNG DỤNG HỌC MÁY TRONG PHÁT HIỆN VÀ DỰ ĐOÁN LỖI HỆ THỐNG ĐIỀU KHIỂN UAV THÔNG QUA LOG BAY

**ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT**

*( Theo định hướng ứng dụng)*

HÀ NỘI – 2025

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

---



**Vũ Thị Hợp**

# NGHIÊN CỨU ỨNG DỤNG HỌC MÁY TRONG PHÁT HIỆN VÀ DỰ ĐOÁN LỖI HỆ THỐNG ĐIỀU KHIỂN UAV THÔNG QUA LOG BAY

**Chuyên ngành** : Hệ thống thông tin
**Mã số**: 8.48.01.04 (Hệ thống thông tin)

**ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT**

***(Theo định hướng ứng dụng)***

NGƯỜI HƯỚNG DẪN KHOA HỌC
TS. VŨ VĂN THỎA

HÀ NỘI - 2025

# LỜI CAM ĐOAN

Tôi cam đoan rằng công trình nghiên cứu này là thành quả của công sức cá nhân của tôi và không sao chép từ bất kỳ nguồn nào khác. Tất cả thông tin được trình bày trong đề án này đều là sản phẩm của công việc cá nhân hoặc được tổng hợp từ nhiều nguồn tài liệu khác nhau. Mọi tài liệu tham khảo đều được trích dẫn một cách hợp pháp và có nguồn gốc rõ ràng. Các dữ liệu và kết quả được trình bày trong đề án đều là trung thực và chưa từng được công bố trong bất kỳ công trình nghiên cứu nào khác.

**Tác giả đề án**



**Vũ Thị Hợp**

# LỜI CẢM ƠN

Lời đầu tiên, học viên xin gửi lời cảm ơn sâu sắc tới các thầy, cô giáo giảng viên khoa Công nghệ thông tin 1, khoa Đào tạo Sau đại học nói riêng và các thầy, cô giáo giảng viên Học viện Công nghệ Bưu chính Viễn thông nói chung. Trong suốt quá trình học tập tại Học viện, các thầy cô đã chỉ bảo, giảng dạy cho em biết bao kiến thức, kinh nghiệm quý báu để em có hành trang vững bước trong tương lai.

Em cũng xin được gửi lời cảm ơn tới thầy hướng dẫn TS. Vũ Văn Thỏa, cảm ơn thầy đã luôn hướng dẫn chỉ bảo tận tình em trong suốt quá trình học tập, nghiên cứu và thực hiện đề án này. Những lời khuyên, sự chỉ bảo của các thầy đã giúp em hoàn thành đề án tốt nghiệp này cũng như có thêm rất nhiều kiến thức, kinh nghiệm trong việc học tập và nghiên cứu.

Dù đã nỗ lực hoàn thành đề án, em hiểu rằng có thể không tránh khỏi những sai sót. Kính mong được thầy cô và các bạn thông cảm và đóng góp ý kiến.

Em xin trân trọng cảm ơn.

# MỤC LỤC

LỜI CAM ĐOAN ..... i
LỜI CẢM ƠN ..... ii
MỤC LỤC ..... iii
DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT ..... v
DANH MỤC CÁC BẢNG ..... vi
DANH MỤC CÁC HÌNH ..... vii
MỞ ĐẦU ..... 1
CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN VÀ CÁC VẤN ĐỀ LIÊN QUAN 3
1.1 Tổng quan về hệ thống điều khiển UAV ..... 3
1.1.1 Giới thiệu chung ..... 3
1.1.2 Các lỗi hệ thống điều khiển UAV thường gặp ..... 5
1.1.3 Phương pháp phát hiện và khắc phục lỗi hệ thống điều khiển UAV 6
1.2 Tổng quan về log bay UAV ..... 10
1.2.1 Khái niệm về log bay UAV ..... 10
1.2.2 Định dạng log bay UAV ..... 11
1.2.3 Thách thức trong xử lý log bay ..... 13
1.3 Bài toán phát hiện lỗi hệ thống điều khiển UAV thông qua log bay 16
1.3.1 Giới thiệu bài toán ..... 16
1.3.2 Ứng dụng phương pháp học máy giải quyết bài toán ..... 18
1.3.3 Tập dữ liệu log bay UAV cho bài toán ..... 20
1.4 Các nghiên cứu liên quan ..... 21
1.5 Kết luận chương 1 ..... 22
CHƯƠNG 2: NGHIÊN CỨU MỘT SỐ PHƯƠNG PHÁP HỌC MÁY ..... 24
2.1 Phương pháp SVM phân loại đa lớp (Multi-class SVM) ..... 24
2.1.1 Nguyên lý hoạt động ..... 24

2.1.2 Kernel ....26
2.1.3 Các siêu tham số ....28
2.1.4 Ưu nhược điểm ....28
2.2 Phương pháp mạng nơ ron nhân tạo ....30
2.2.1 Nguyên lý hoạt động ....30
2.2.2 Cấu trúc mạng đề xuất ....31
2.2.3 Hàm kích hoạt ....32
2.2.4 Siêu tham số cần tối ưu ....33
2.2.5 Ưu điểm và hạn chế ....35
2.3 Kết luận chương 2 ....36
CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ ....38
3.1 Thiết lập môi trường thử nghiệm trên Google Colab ....38
3.2 Thu thập dữ liệu: ....39
3.3 Triển khai quy trình thực nghiệm ....39
3.3.1 Tiền xử lý log bay ....39
3.3.2 Huấn luyện mô hình ....42
3.3.3 Thử nghiệm ....43
3.4 Đánh giá hiệu quả mô hình phát hiện lỗi UAV ....44
3.5 Kết luận chương 3 ....55
KẾT LUẬN ....57
DANH MỤC CÁC TÀI LIỆU THAM KHẢO ....59

# DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

| Ký hiệu, chữ viết tắt | Tiếng anh đầy đủ | Nghĩa tiếng việt |
|---|---|---|
| UAV | Unmanned Aerial Vehicles | Máy bay không người lái |
| SVM | Support Vector Machine | Máy hỗ trợ vec tơ |
| ANN | Artificial Neural Network | Mạng nơ ron nhân tạo |
| GPU | Graphics Processing Unit | Đơn vị xử lý đồ họa |
| RNN | Recurrent Neural Networks | Mạng neural hồi tiếp |
| LSTM | Long Short-Term Memory Network | Mạng bộ nhớ dài ngắn hạn |
| CNN | Convolutional Neural Network | Mạng neural tích chập |
| FC | Flight control | Bộ điều khiển bay |
| FTC | Fault-Tolerant Control | Điều khiển chịu lỗi |

# DANH MỤC CÁC BẢNG

Bảng 3-1 Phân bố nhãn ..... 43

Bảng 3-2 Kết quả phân tích từng lớp mô hình SVM ..... 50

Bảng 3-3 Kết quả từng lớp sau khi chạy mô hình ANN ..... 54

# DANH MỤC CÁC HÌNH

Hình 1-1 Sơ đồ biểu diễn hệ thống điều khiển UAV ..... 4

Hình 1-2 Minh họa log messages cơ bản ..... 11

Hình 1-3 Log định dạng Tlog ..... 12

Hình 1-4 Log định dạng CSV ..... 13

Hình 2-1 Minh họa nguyên lý hoạt động của SVM ..... 25

Hình 2-2 SVM tuyến tính ..... 26

Hình 2-3 SVM đa lớp sử dụng kernel RBF ..... 27

Hình 2-4 Mô hình mạng ANN đề xuất ..... 31

Hình 3-1 Lệnh cài thư viện trên Google Colab ..... 38

Hình 3-2 Ma trận nhầm lẫn model SVM ..... 50

Hình 3-3 Ma trận nhầm lẫn mô hình ANN ..... 53

Hình 3-4 Biểu đồ quá trình huấn luyện model Accuracy và Loss ..... 54

# MỞ ĐẦU

Hiện nay lĩnh vực hàng không có những bước đột phá một cách mạnh mẽ với sự phát triển nhanh chóng của các tổ hợp máy bay không người lái( UAV) được áp dụng cho các lĩnh vực như: khảo sát địa hình - bản đồ, quân sự, lực lượng vũ trang, giám sát và logistic. Sự phát triển nhanh chóng này kéo theo những yêu cầu cao hơn về đảm bảo độ tin cậy, an toàn và dễ dàng vận hành - một thách thức kỹ thuật quan trọng khi các hệ thống UAV ngày càng trở nên phức tạp.

Trong bối cảnh đó, nhật ký bay (flight log) được xem như một nguồn dữ liệu quý giá, cung cấp cái nhìn toàn diện về trạng thái và tình hình hoạt động của UAV. Tuy nhiên, phân tích bằng phương pháp truyền thống gặp nhiều hạn chế, bất cập trước khối lượng dữ liệu (data) khổng lồ và tính đa chiều của hệ thống hiện đại. Công nghệ học máy thể hiện ưu thế vượt trội trong việc xử lý khối lượng dữ liệu log bay lớn, với khả năng phát hiện các thông tin bất thường và dự đoán lỗi tiềm ẩn. So với các phương pháp dựa trên mô hình lý thuyết, học máy mang lại tỉ lệ chính xác cao hơn trong việc nhận diện các lỗi phức tạp, đồng thời cải thiện đáng kể hiệu quả bảo trì thông qua khả năng dự báo sớm. Việc ứng dụng học máy trong phân tích log bay giúp phát hiện nhanh chóng và kịp thời các sự cố như mất kết nối, lỗi cảm biến hay bất thường trong hành vi bay. Hơn nữa, công nghệ này còn cho phép phân tích đa chiều các tín hiệu hệ thống, giảm thiểu sai sót trong đánh giá tình trạng vận hành.

Nhiều nghiên cứu quốc tế đã chứng minh mức độ hiệu quả của học máy trong việc nâng cao mức độ tin cậy và an toàn của hệ thống UAV, mở ra triển vọng trong các ứng dụng rộng rãi tại Việt Nam nhằm nâng cao năng lực quản lý, giám sát và xử lý của đội bay. Xuất phát từ những cơ sở đó, nghiên cứu "**Nghiên cứu ứng dụng học máy trong phát hiện và dự đoán lỗi hệ thống điều khiển UAV thông qua log bay**" được xây dựng với mong muốn đóng góp và xây dựng những hướng xử lý công nghệ hiện đại, mang lại độ tin cậy cao hơn nữa cho lĩnh vực hàng không tại Việt Nam.

Mục tiêu chính của đề án là khai thác các kỹ thuật học máy để nâng cao khả năng phát hiện sớm và dự báo chính xác lỗi trong hệ thống điều khiển UAV, đáp ứng

tiêu chuẩn về bảo đảm độ tin cậy và tính kịp thời trong vận hành thực tế. Đối tượng nghiên cứu tập trung vào các mô hình học máy tiên tiến như SVM và mạng neuron nhân tạo, được tối ưu cho bài toán nghiên cứu phân tích và nhận diện bất thường dựa trên log bay. Phạm vi nghiên cứu được xác định trên cơ sở dữ liệu log bay từ các UAV cỡ nhỏ, tập trung vào các lỗi của phần mềm hay chương trình điều khiển có thể phát hiện thông qua phân tích tín hiệu hệ thống và hành vi bay.

Nội dung đề án được trình bày thành ba chương theo cấu trúc sau:

**Chương 1: Giới thiệu bài toán và các vấn đề liên quan**

Trình bày tổng quan về hệ thống điều khiển bay, log bay, bài toán dự đoán lỗi hệ thống điều khiển thông qua log bay và các nghiên cứu liên quan.

**Chương 2: Nghiên cứu một số phương pháp học máy**

Trình bày toàn diện nền tảng lý thuyết của hai phương pháp học máy tiêu biểu được ứng dụng trong bài toán phân tích log bay UAV và phân loại lỗi hệ thống, đó là máy vector hỗ trợ (SVM) và mạng neuron nhân tạo (ANN).

**Chương 3: Thử nghiệm và đánh giá kết quả**

Mục này sẽ trình bày một cách đầy đủ quá trình thiết lập môi trường thử nghiệm, thu thập và tinh chỉnh dữ liệu, triển khai hai mô hình học máy SVM và ANN để đánh giá khả năng phát hiện và phân loại lỗi hệ thống.

# CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN VÀ CÁC VẤN ĐỀ LIÊN QUAN

*Chương này tập trung trình bày tổng quan về hệ thống điều khiển bay, cấu trúc log bay UAV, bao gồm các lớp dữ liệu chính và định dạng lưu trữ, làm rõ những thách thức trong việc xử lý và phân tích lượng dữ liệu lớn, phức tạp này. Đồng thời làm rõ bài toán đặt ra và tập dữ liệu mà bài toán cần xử lý.*

## 1.1 Tổng quan về hệ thống điều khiển UAV

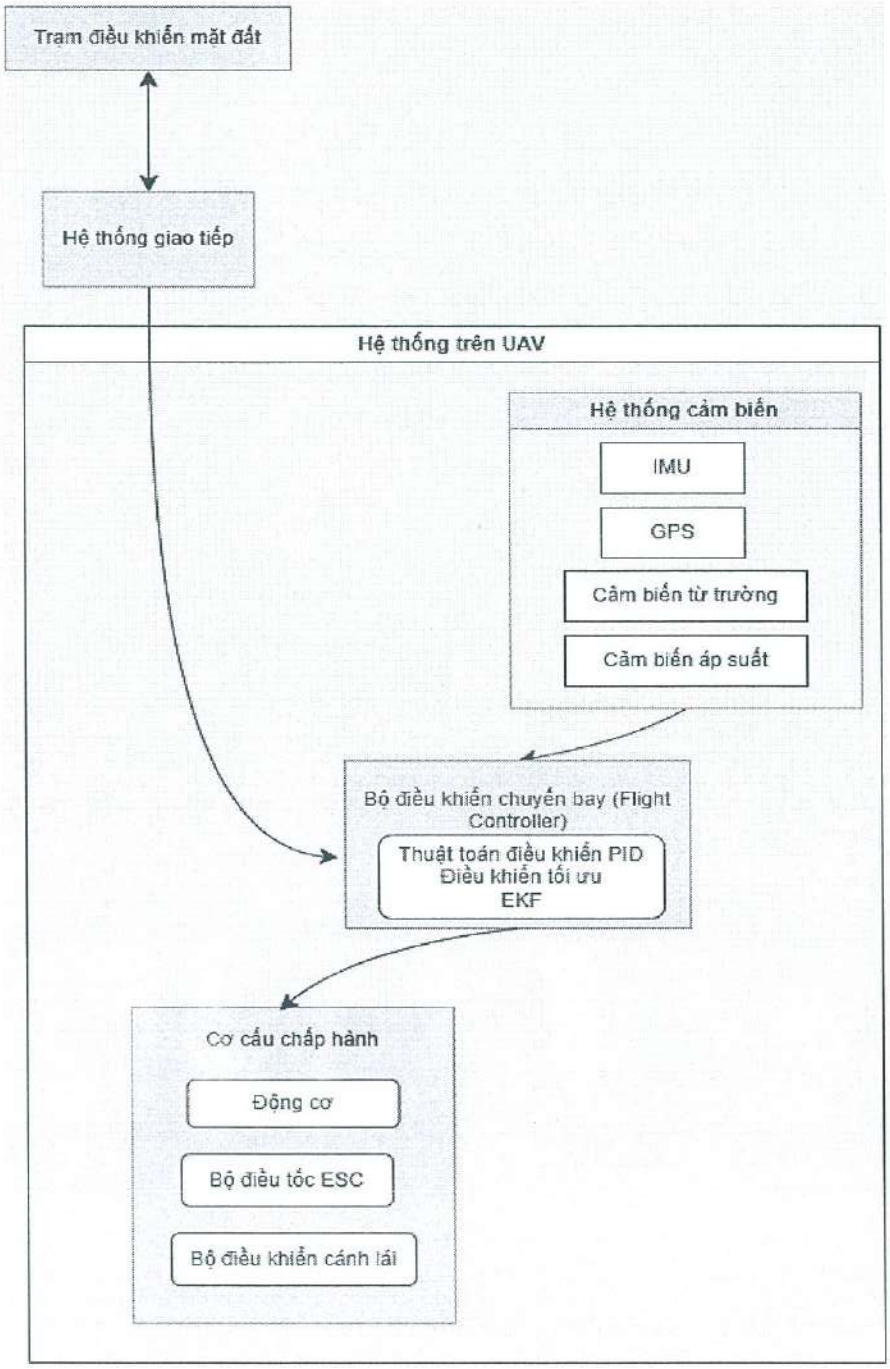
### 1.1.1 Giới thiệu chung

Hệ thống điều khiển UAV (Unmanned Aerial Vehicle) là một hệ thống tích hợp đa thành phần, hoạt động theo nguyên lý điều khiển vòng kín để duy trì ổn định và thực hiện các nhiệm vụ bay tự động [20].

Hệ thống này được bao gồm từ 4 thành phần chính có mối quan hệ chặt chẽ với nhau được mô tả trên hình 1.1. Bộ điều khiển chuyến bay (Flight Controller) đóng vai trò trọng tâm, thực hiện ghi nhớ log bay, xử lý thông tin và tính toán các lệnh điều khiển theo thời gian thực. Flight Controller (FC) đảm nhiệm việc triển khai các thuật toán điều khiển như PID, điều khiển tối ưu, lọc Kalman mở rộng (EKF) để tính toán trạng thái và giữ ổn định quỹ đạo bay [18]. Dữ liệu phục vụ cho các thuật toán này được cung cấp bởi các hệ thống cảm biến (Sensors), bao gồm IMU, GPS, cảm biến từ trường và cảm biến áp suất, nhằm xác định đầy đủ trạng thái không gian của UAV như vị trí, vận tốc, gia tốc, góc quay và độ cao [34]. Dựa trên kết quả phân tích, FC sẽ gửi tín hiệu đến các bộ truyền động (Actuators) như động cơ, bộ điều tốc và các bề mặt điều khiển, giúp UAV thực hiện các động tác bay như tăng giảm độ cao, thay đổi hướng hoặc giữ vị trí. Hệ thống thông tin liên lạc (Communication System) đảm bảo kết nối liên tục giữa UAV và trạm điều khiển tại mặt đất, cho phép gửi/nhận dữ liệu (data) và lệnh điều khiển (telemetry) trong thời gian thực [27].

Để đáp ứng các yêu cầu cần thiết trong quá trình vận hành ở các môi trường phức tạp, hệ thống điều khiển UAV hiện đại thường được tích hợp các cơ chế chịu

lỗi (fault-tolerant) [15]. Các thuật toán như bộ lọc Kalman, điều khiển PID và hoạch định quỹ đạo được ứng dụng để nâng cao khả năng tự vận hành và bảo đảm an toàn cho hệ thống [18].




**Hình 1-1 Sơ đồ biểu diễn hệ thống điều khiển UAV**

### 1.1.2 Các lỗi hệ thống điều khiển UAV thường gặp

Hệ thống tổ hợp điều khiển UAV phải đối mặt với đa dạng các lỗi kĩ thuật, có thể được chia thành ba nhóm cơ bản nhất dựa trên nguyên nhân và vị trí phát sinh:

- Lỗi cảm biến chiếm tỷ lệ đa số trong các sự cố UAV, với các dạng điển hình bao gồm mất tín hiệu GPS đột ngột do che khuất vệ tinh hoặc nhiễu điện từ, trôi giá trị cảm biến IMU theo thời gian( sensor drift), và nhiễu cao ở barometer, magnetometer. Các lỗi cảm biến có thể dẫn tới sai lệch trạng thái ước lượng và suy giảm hiệu năng điều khiển. Finn và cộng sự (2024) [10] đã mô tả chi tiết cơ chế tấn công giả mạo GPS (spoofing attacks) tinh vi, trong đó kẻ tấn công phát ra tín hiệu GPS giả mạo khiến UAV xác định sai vị trí, dẫn đến mất phương hướng nghiêm trọng. Các lỗi này đặc biệt nguy hiểm trong khoảng thời gian cất hạ cánh, yêu cầu về độ chính xác vị trí rất cao. Tại sự kiện 30/4/2025, buổi trình diễn drone ánh sáng sử dụng hơn 10.500 drone đã phải hủy bỏ do nhiễu sóng diện rộng, một số lượng lớn drone bất ngờ mất kiểm soát và rơi xuống.
- Lỗi cơ cấu chấp hành thường xuất phát từ các nguyên nhân: hư hỏng cơ khí, lỗi điện tử (cháy MOSFET trong ESC), lỗi động cơ, giảm lực đẩy động cơ, servo kẹt hoặc hoạt động không đồng bộ giữa các phần tử chấp hành. Những lỗi này tác động trực tiếp đến lực momen sinh ra và có thể gây mất ổn định bay nếu không phát hiện kịp thời [9], [14].

  Một tai nạn điển hình cho lỗi này xảy ra vào tháng 6 năm 2018 khi UAV chiến lược RQ-4 Global Hawk rơi xuống biển ngoài khơi căn cứ Hải quân Rota (Tây Ban Nha). Kết quả điều tra được công bố nguyên nhân tai nạn xuất phát từ kế hoạch nhiệm vụ không đầy đủ kết hợp với rò rỉ đường ống nhiên liệu, khiến động cơ mất khả năng duy trì lực đẩy trong chuyến bay dài. Sự cố này minh họa rõ ràng rủi ro của các lỗi kỹ thuật kết hợp sai sót vận hành trong hệ thống UAV.

- Lỗi phần mềm điều khiển thường bao gồm lỗi triển khai thuật toán, lỗi giao tiếp giữa các module, lỗi xử lý tín hiệu, lỗi tích hợp hệ thống, lỗi cấu hình có thể dẫn tới phản ứng điều khiển sai lệch. Những lỗi này đôi khi khó phát hiện bằng cách kiểm tra đơn giản, đặc biệt khi biểu hiện giống với nhiễu hay trạng thái biên [15]. Như trường hợp UAV bị mất khả năng giữ vị trí vì module GPS không truyền dữ liệu đầy đủ cho EKF do lỗi giao thức MAV Link bị xung đột khi có thêm một module camera truyền telemetry cùng lúc. Sự cố khiến bộ điều khiển không nhận được các gói vị trí mới, buộc UAV chuyển sang chế độ failsafe và hạ độ cao đột ngột. Vấn đề này chỉ xuất hiện khi tải truyền thông cao, nên khó phát hiện bằng kiểm thử thông thường.

### 1.1.3 Phương pháp phát hiện và khắc phục lỗi hệ thống điều khiển UAV

Để bảo đảm độ tin cậy và an toàn trong hoạt động, hệ thống điều khiển UAV đòi hỏi các cơ chế phát hiện và khắc phục lỗi (Fault Detection, Isolation and Recovery - FDI/R) hiệu quả. Lĩnh vực này đã trải qua sự chuyển dịch từ các phương pháp truyền thống dựa trên mô hình vật lý sang các tiếp cận hiện đại dựa trên dữ liệu, dẫn đến sự hình thành của một kiến trúc phòng thủ nhiều lớp.

#### 1.1.3.1 Phương pháp phát hiện và cô lập lỗi

a) Phương pháp dựa trên mô hình vật lý

Phương pháp này áp dụng các mô hình toán học chính xác để tái tạo động lực học của UAV trong môi trường mô phỏng, giúp kiểm tra hiệu suất và độ ổn định của bộ điều khiển. Khi có sự không đồng nhất giữa đầu ra tính toán của mô hình và đầu ra thực tế của hệ thống, một tín hiệu cảnh báo lỗi (residual) sẽ được tạo ra.

Các kĩ thuật phổ biến:

- Kỹ thuật ước lượng trạng thái và tham số: Đây là kỹ thuật được ứng dụng phổ biến nhất, sử dụng các bộ lọc (như Kalman, các biến thể phi tuyến như EKF, UKF) để tính toán và ước lượng các trạng thái hoặc tham số của hệ thống mà không thể

đo lường trực tiếp. Lỗi được phát hiện thông qua việc so sánh giữa các giá trị tính toán ước lượng và giá trị đo lường thực tế từ cảm biến. Khi sự sai khác vượt quá một ngưỡng cho phép, lỗi được xác định.

- Phân bổ điều khiển có thể cấu hình lại (Reconfigurable Control Allocation): Nghiên cứu của Zhou và cộng sự (2010) [34] là điển hình cho kĩ thuật này. Khi một cơ cấu truyền động (actuator) bị lỗi, thuật toán sử dụng phương pháp bình phương tối thiểu có trọng số (Weighted Least Squares) để phân phối lại các tín hiệu điều khiển đến các actuator còn hoạt động, đảm bảo UAV vẫn có thể duy trì được khả năng hoạt động. Phương pháp này giúp bảo đảm khả năng điều khiển và ổn định ngay cả khi một hoặc nhiều actuator bị lỗi, đặc biệt hữu ích với UAV đa rotor có khả năng dự phòng cơ cấu chấp hành.

Phương pháp này mang lại hiệu quả tốt đối với các dạng lỗi đã được mô hình hóa trước dễ triển khai trong môi trường kiểm soát, cung cấp khả năng giải thích và phân tích chi tiết. Phương pháp này cho phép nắm bắt chính xác nguyên nhân gốc rễ của lỗi, cung cấp thông tin chi tiết về sự cố và ước lượng mức độ nghiêm trọng.Tuy nhiên, nó lại đòi hỏi một mô hình toán học chính xác của đối tượng. Hiệu quả suy giảm đáng kể khi đối mặt với các lỗi chưa được mô hình hóa, sự bất định của mô hình hoặc nhiễu môi trường phức tạp [15]. Việc xây dựng mô hình đảm bảo chính xác cao cho toàn bộ hệ thống UAV là rất phức tạp và tốn kém.

b) Phương pháp dựa trên dữ liệu (data-driven)

Song song với sự nâng cấp của hệ thống cảm biến và sự mở rộng về năng lực lưu trữ dữ liệu, phương pháp dựa trên dữ liệu đang trở thành xu hướng chủ đạo nhờ khả năng học trực tiếp từ các data vận hành mà không cần một mô hình vật lý chính xác.

Các phương pháp nổi bật:

- Mô hình dự báo chuỗi thời gian: Wang và cộng sự (2019) [28] giới thiệu một mô hình LSTM dùng để dự đoán các giá trị cảm biến trong tương lai. Độ sai lệch vượt ngưỡng giữa giá trị dự báo và giá trị thực nghiệm được xem là dấu hiệu bất thường

phản ánh khả năng hệ thống đang phát sinh lỗi. Mô hình này chứng tỏ hiệu quả cao đối với các lỗi phát triển chậm (incipient faults), một nhóm lỗi mà các phương pháp giám sát truyền thống khó phát hiện do đặc tính thay đổi nhỏ và tích lũy theo thời gian.

- Trích xuất đặc trưng đa phân giải: Al-Haddad và Jaber (2023) [2] phát triển một kiến trúc mạng neuron sâu kết hợp biến đổi wavelet, khai thác đặc trưng tín hiệu rung động theo nhiều độ phân giải, từ đó tăng cường năng lực phát hiện những dạng lỗi tinh vi mà các phương pháp phân tích truyền thống dễ bỏ sót. Phương pháp này cho phép phân tích tín hiệu ở cả 2 miền: thời gian và tần số, đưa ra bức tranh tổng thể về tình trạng hệ thống.
- Phát hiện bất thường không giám sát: Các kỹ thuật như phân tích thành phần chính và Autoencoder được ứng dụng để học biểu diễn của trạng thái vận hành bình thường. Mọi sự sai lệch so với không gian biểu diễn này đều được xác định là bất thường, phù hợp cho các tình huống thiếu dữ liệu được gán nhãn lỗi. Các kỹ thuật phân nhóm dữ liệu như K-means và DBSCAN cũng được ứng dụng để chia nhóm các trạng thái vận hành và phát hiện các trạng thái bất thường.
- Mô hình học sâu: Các kiến trúc học sâu cho phép tối ưu hoá xử lý luồng dữ liệu đa chiều từ cảm biến.
    + Debele và cộng sự (2023) [9] đã giới thiệu một framework kết hợp Mạng neuron tích chập (CNN) với "cơ chế chú ý" (attention) để phát hiện và cô lập lỗi cơ cấu chấp hành, trong đó cơ chế Attention giúp tập trung vào các tín hiệu cảm biến mang tính chỉ báo cao. Kiến trúc này cho phép khai thác dữ liệu đa chiều thu thập từ nhiều cảm biến đồng thời, giúp xác định chính xác các bộ phận hoặc cơ cấu gặp sự cố.
    + Yousefi và cộng sự (2018) [31] phát triển một hệ thống giám sát dựa trên nền tảng điện toán đám mây sử dụng SVM đa lớp để phân loại lỗi theo thời gian thực, minh chứng hiệu quả thực tiễn của mô hình học máy cổ điển trong bài toán này. Hệ thống này cho phép giám sát nhiều UAV đồng thời và phát hiện lỗi theo thời gian thực.

### 1.1.3.2 Phương pháp khắc phục lỗi

Sau khi lỗi được phát hiện và cô lập, các chiến lược điều khiển chịu lỗi (Fault-Tolerant Control- FTC) được kích hoạt để duy trì vận hành ổn định và bảo đảm hiệu suất của hệ thống.

Điều khiển dự phòng (Passive FTC): Hệ thống được trang bị một bộ điều khiển có độ dự trữ ổn định cao, đảm bảo khả năng chịu đựng một tập hợp lỗi được xác định trước mà không cần thay đổi cấu trúc. Phương pháp này vận dụng các kỹ thuật điều khiển bền vững (robust control) để thiết kế FC có khả năng chống chịu với các bất định và lỗi hệ thống. Chẳng hạn, khi thiết kế bộ điều khiển cho quadcopter, kĩ sư sẽ thiết kế FC dự phòng trong trường hợp động cơ bị suy giảm 20% hiệu suất vẫn giữ được trạng thái cân bằng và bay ổn định. Phương pháp này nổi bật với các đặc tính: đơn giản, đáng tin cậy và phản ứng ngay lập tức. Tuy nhiên, nhược điểm là hiệu suất thấp trong điều kiện bình thường do thiết kế bảo thủ, và phạm vi bảo vệ hạn chế, chỉ hiệu quả với các lỗi nằm trong phạm vi giả định; các lỗi ngoài dự kiến có thể làm UAV mất ổn định hoặc sụp đổ.

Điều khiển thích nghi chủ động (Active FTC): Hệ thống tích hợp một module giám sát lỗi. Khi lỗi xảy ra, hệ thống chủ động điều chỉnh tham số hoặc thay thế cấu trúc bộ điều khiển dựa trên thông tin lỗi nhận được. Các kỹ thuật điều khiển thích nghi (adaptive control) được dùng để tự động chỉnh sửa tham số bộ điều khiển nhằm bù đắp cho ảnh hưởng của lỗi. Ví dụ, khi một cảm biến GPS bị mất, bộ điều khiển thích nghi có thể tự động tăng cường dựa trên các nguồn dữ liệu khác như hệ thống định vị vệ tinh DGPS (Differential Global Positioning System), hệ thống dẫn đường quán tính INS (Inertial Navigation System), hệ thống định hướng từ FVU (Flux Valve Unit) hay cảm biến rada để duy trì khả năng định vị và điều khiển UAV.

Điều khiển cấu trúc lại (Reconfigurable Control): Đây là dạng mạnh mẽ nhất của Active FTC. Khi xảy ra lỗi nghiêm trọng (như mất một động cơ), toàn bộ cấu trúc điều khiển có thể được thay đổi. Kỹ thuật then chốt của phương pháp này là phân bổ điều khiển (Control Allocation - CA). Bộ điều khiển cao cấp vẫn tính toán lực và

mô-men mong muốn. Khi một actuator bị lỗi, bộ phân bổ điều khiển sẽ tính toán lại cách phân phối các lệnh này đến các actuator còn hoạt động. Zhou và cộng sự (2010) [34] đã minh họa trên một hexacopter (6 động cơ), khi một động cơ hỏng, bộ điều khiển phân bổ sử dụng thuật toán bình phương tối thiểu có trọng số (Weighted Least Squares) nhằm lựa chọn phương án tốt nhất của 5 động cơ còn lại để tạo ra lực và mô-men gần giống với yêu cầu nhất, cho phép UAV tiếp tục bay mặc dù có thể bị nghiêng lệch.

Nghiên cứu của Kobayashi và Takahashi (2011) [15] cung cấp một khung thiết kế tích hợp cả ba phương pháp, hướng tới xây dựng một hệ thống điều khiển bay thông minh và có cơ chế phục hồi. Phương pháp này bao gồm việc thay đổi luật điều khiển, tái phân bổ tín hiệu điều khiển, hoặc thậm chí thay đổi cấu trúc hệ thống để duy trì khả năng điều khiển.

Việc kết hợp hài hòa giữa phương pháp dựa trên mô hình vật lý và dựa trên dữ liệu tạo ra hệ thống phát hiện và khắc phục lỗi toàn diện, đáp ứng tiêu chí ngày càng khắt khe về độ tin cậy và mức độ an toàn trong vận hành UAV. Các hướng nghiên cứu mới nhấn mạnh vào phát triển hệ thống tự học (self-learning) có khả năng cập nhật và thích nghi với các dạng lỗi chưa từng biết trước, tạo tiền đề cho kỷ nguyên mới của lĩnh vực hàng không không người lái.

## 1.2 Tổng quan về log bay UAV

### 1.2.1 Khái niệm về log bay UAV

Log bay UAV là một bản ghi toàn diện, lưu trữ toàn bộ tham số, dữ liệu quan trọng phát sinh trong suốt quá trình hoạt động của thiết bị bay không người lái. Các dữ liệu này bao gồm các tham số dữ liệu quỹ đạo di chuyển, trạng thái các hệ thống con( như điều hướng, động cơ, liên lạc), và các chỉ số hiệu suất vận hành [28] như trên hình 1.2. Về bản chất, log bay có thể được ví như một "hộp đen" hoặc " nhật ký chuyến bay" kỹ thuật số của UAV, không chỉ ghi lại lộ trình mà còn phản ánh một cách chi tiết cách thức hệ thống phản hồi và tương tác trong các yếu tố môi trường và thời tiết khác nhau [33]. Nhờ khả năng ghi nhớ và lưu giữ dữ liệu có tần suất cao và

đa dạng, log bay đóng vai trò cốt lõi trong việc phân tích nguyên nhân sự cố, nâng cao khả năng cảnh báo sớm các hiện tượng bất thường và xu hướng hỏng hóc [6], [34]. Từ đó, dữ liệu log trở thành nội dụng quan trọng cho việc thực hiện bảo trì, bảo dưỡng và dự đoán lỗi, giúp tối ưu hoá, nâng cao tuổi thọ trang thiết bị, cải thiện hiệu quả và bảo đảm mức độ an toàn trong các nhiệm vụ bay tiếp theo [28], [33].

Một log bay điển hình thường bao gồm bốn nhóm dữ liệu chính:

- Dữ liệu từ các cảm biến: góc hướng, gia tốc, vận tốc góc (gyroscope), áp suất khí quyển, độ cao, tọa độ GPS.
- Dữ liệu điều khiển: tín hiệu RC từ người điều khiển, lệnh điều khiển tự động, thông số PID, chế độ bay (Manual, Loiter, Auto…).
- Dữ liệu hệ thống: điện áp và dòng pin, nhiệt độ, trạng thái vi điều khiển, trạng thái các thiết bị ngoại vi , trạng thái hệ thống điện, tình trạng liên kết truyền thông.
- Dữ liệu quỹ đạo bay: vị trí không gian, hướng bay, tốc độ, waypoint và thông tin hành trình.

| | | |
|---|---|---|
| 0:00:35 | INFO | Armed by external command |
| 0:00:35 | INFO | [tone_alarm] arming warning |
| 0:00:37 | INFO | Takeoff detected |
| 0:00:50 | INFO | GNSS data fusion stopped |
| 0:00:52 | WARNING | [timesync] time jump detected. Resetting time synchroniser. |
| 0:01:43 | WARNING | [timesync] time jump detected. Resetting time synchroniser. |

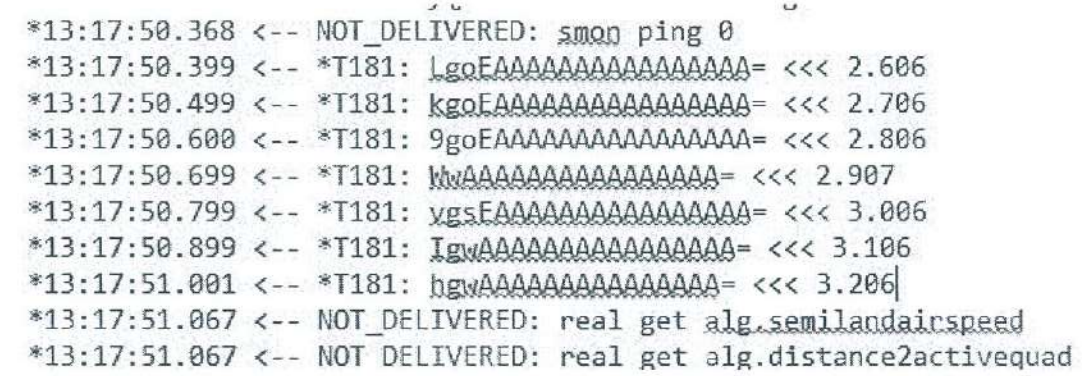
**Hình 1-2 Minh họa log messages cơ bản**

### 1.2.2 Định dạng log bay UAV

Tùy thuộc vào nền tảng phần mềm điều khiển (firmware), log bay sẽ được lưu trữ bởi các định dạng và logic khác nhau, mỗi định dạng sẽ có ưu điểm và hạn chế riêng về kích thước và khả năng đọc:

- Binary Log (BIN/.bin) và DataFlash: Định dạng nhị phân gọn nhẹ, phổ biến trong ArduPilot. Cần được giải mã bằng các công cụ chuyên dụng (như Mission Planner) để đọc được [28].
- ULog (.ulg): Định dạng log mặc định cho hệ điều hành PX4, được thiết kế để có cấu trúc rõ ràng, hỗ trợ ghi dữ liệu theo topic và dễ dàng phân tích [28].
- Tlog (Telemetry Log): Ghi lại dữ liệu truyền qua liên kết giữa UAV và trạm mặt đất (GCS) [28].
- CSV, JSON: Các định dạng văn bản phổ biến, thường kết quả sau khi giải mã là log nhị phân. Chúng dễ đọc và xử lý bằng ngôn ngữ lập trình thông thường như Python [34].

Dữ liệu (data) được ghi với tần số 10–50 Hz, nghĩa là cứ mỗi 20-100 mili giây lại có một bản ghi mới được tạo ra, dẫn đến khối lượng dữ liệu rất lớn ngay cả với những chuyến bay ngắn [28].




**Hình 1-3 Log định dạng Tlog**

| timestamp | gyro_rad[0] | gyro_rad[1] | gyro_rad[2] |
|---|---|---|---|
| 1550930271 | 0.0035212373 | 0.0017454743 | -0.0018244548 |
| 1550932740 | 0.0023754481 | 0.002353345 | -0.0014214296 |
| 1550935211 | 0.001572728 | 0.0016055844 | -0.0005587577 |
| 1550940146 | 0.0010747772 | 0.0009677395 | -0.00035058474 |
| 1550942615 | 0.0025353278 | 0.0035274415 | -0.00049214484 |
| 1550947552 | 0.0013528933 | 0.002496569 | -0.0017445137 |
| 1550950021 | 0.00093987794 | 0.0017055067 | -0.0024123364 |
| 1550952489 | -0.0002242273 | 0.0020019475 | -0.0025488988 |
| 1550957427 | 0.0010781082 | 0.002569843 | -0.0009034951 |
| 1550959896 | 0.0020023957 | 0.0010260248 | -0.0020492827 |
| 1550964833 | 0.0002570711 | 0.0011259523 | -0.0023990148 |
| 1550967302 | 0.0011214096 | 0.0021634875 | -0.0018427749 |
| 1550969771 | 0.0020407066 | 0.003597389 | -0.0005637561 |
| 1550974708 | 0.0012313203 | 0.0002299724 | -0.0012682131 |

**Hình 1-4 Log định dạng CSV**

### 1.2.3 Thách thức trong xử lý log bay

Dữ liệu log bay UAV, mặc dù mang lại giá trị quan trọng trong giám sát và phân tích hoạt động bay, nhưng việc xử lý và khai thác nguồn thông tin từ các log này tồn tại nhiều thách thức đáng kể.

Các UAV hiện đại được trang bị nhiều cảm biến (IMU, GPS, barometer, camera hoặc LiDAR) và ghi log với tần số 10-50 Hz [28]. Điều này dẫn đến khối lượng dữ liệu lớn chỉ trong một chuyến bay ngắn (có thể từ hàng trăm MB đến hàng chục GB), gây ra hạn chế cho công tác lưu trữ, truyền tải và đặc biệt là phân tích thời gian thực [33].

Log bay tổng hợp nhiều luồng dữ liệu khác nhau về định dạng, tần suất và đơn vị đo lường, bao gồm dữ liệu số (như điện áp pin, tốc độ), dữ liệu không gian- thời gian (GPS, waypoint), thậm chí dữ liệu phi cấu trúc (ảnh hoặc video) [28]. Sự không đồng nhất này làm tăng đáng kể chi phí tiền xử lý và hiệu chuẩn dữ liệu.

Data log bay thường xuyên có nhiễu do sai số cảm biến, tín hiệu thiếu hụt hoặc gián đoạn trong quá trình truyền dữ liệu, thậm chí có thể gặp tình trạng dữ liệu sai lệch do lỗi hiệu chuẩn [9], [33]. Nếu không được giải quyết một cách toàn diện, các

vấn đề này có thể ảnh hưởng trực tiếp đến độ tin cậy và hiệu quả của các mô hình phân tích và dự báo.

Quá trình động lực học của UAV vốn rất phi tuyến và nhạy cảm với các nhiễu động nhỏ. Mối quan hệ phức tạp giữa tín hiệu điều khiển, phản hồi cảm biến và quỹ đạo thực tế rất khó để mô hình hóa chính xác thông qua các phương pháp tuyến tính cổ điển hoặc các ngưỡng cố định, dẫn đến hạn chế trong khả năng phát hiện lỗi kịp thời [9].

Nhiều lỗi hệ thống quan trọng thường khởi phát bởi những tín hiệu bất thường rất nhỏ và mờ nhạt, dễ dàng bị che lấp bởi nhiễu nền. Các dấu hiệu cảnh báo sớm này rất dễ bị bỏ sót trong quy trình phân tích thủ công nhưng lại có tầm quan trọng then chốt trong việc ngăn chặn các sự cố nghiêm trọng xảy ra sau này [9], [34].

Một điển hình liên quan đến sai lệch dữ liệu cảm biến là trường hợp UAV sử dụng IMU có hiện tượng drift của gyroscope sau khi cất cánh trong điều kiện rung động mạnh. Sau khoảng 2–3 phút bay, log bay ghi nhận vận tốc góc trục Z tăng bất thường từ 0.02 rad/s lên 0.15 rad/s dù UAV đang bay ổn định. Nếu không xử lý, mô hình phát hiện lỗi có thể hiểu nhầm đây là quay xoắn đột ngột (yaw instability) và đưa ra cảnh báo sai. Tuy nhiên, trong một vài tình huống mà lỗi thực sự xuất hiện (như lệch góc cánh quạt), lực kéo, đẩy bất cân bằng tạo ra dao động thực sự ở vận tốc góc, nhưng tín hiệu lại bị che mờ bởi nhiễu IMU, khiến hệ thống không thể phát hiện kịp thời. Đây là minh chứng rõ nhất cho thấy chất lượng của tiền xử lý dữ liệu, lọc nhiễu và đồng bộ hóa cảm biến trước khi triển khai vào mô hình học máy.

Bên cạnh những thách thức kỹ thuật toàn cầu, hệ sinh thái UAV tại Việt Nam tồn tại một số đặc thù làm phức tạp thêm quy trình xử lý và phân tích log bay. Những đặc thù này xuất phát từ điều kiện môi trường, trình độ phát triển công nghệ và tính mới của thị trường, đòi hỏi các giải pháp được tùy chỉnh và những nghiên cứu chuyên sâu.

Tính không đồng bộ và thiếu chuẩn hóa trong hệ thống thu thập dữ liệu tạo ra rào cản đầu tiên. Tình trạng phổ biến tại Việt Nam là các giải pháp UAV thường kết

hợp linh kiện từ nhiều nguồn gốc khác nhau, từ phần cứng thương mại đến tự chế, cùng với các bộ điều khiển bay sử dụng nhiều nền tảng firmware đa dạng. Hệ quả trực tiếp là sự phân mảnh về định dạng log, khi dữ liệu được ghi lại theo các chuẩn khác nhau với schema không tương thích. Điều này buộc các kỹ sư phải thực hiện quy trình tiền xử lý phức tạp bao gồm giải mã, ánh xạ trường dữ liệu và đồng bộ hóa thời gian, làm tăng đáng kể chi phí, thời gian và tạo ra rào cản kỹ thuật cho việc nâng cao công cụ phân tích tổng hợp.

Tình hình môi trường tại Việt Nam cũng tác động sâu sắc đến chất lượng dữ liệu. Môi trường hoạt động phức tạp với địa hình đô thị dày đặc và vùng núi hiểm trở thường xuyên gây nhiễu tín hiệu GPS, dẫn đến các lỗi mất kết nối, sai số định vị và dữ liệu không ổn định trong log, làm giảm độ chính xác cho các phân tích sau này.

Sự khan hiếm của các bộ dữ liệu lỗi công khai và được gán nhãn cản trở đáng kể công tác tìm hiểu, nghiên cứu và phát triển. Khác với cộng đồng nghiên cứu quốc tế có thể dễ dàng tiếp cận các tài nguyên dữ liệu rộng lớn, các đội ngũ nghiên cứu nội địa phải tự thiết kế mạch dữ liệu riêng với quy mô nhỏ và thiếu tính đa dạng. Việc gán nhãn thủ công cho các sự kiện bất thường không chỉ tốn kém mà còn thiếu tính nhất quán, từ đó hạn chế khả năng huấn luyện các mô hình học máy mạnh mẽ và đối chiếu kết quả giữa các công trình nghiên cứu.

Khung pháp lý chưa hoàn thiện về tiêu chuẩn log và quy trình phân tích gây khó khăn cho việc tổng hợp dữ liệu từ các nguồn khác nhau và ảnh hưởng đến hiệu quả quản lý, giám sát bay của cơ quan nhà nước.

Những thách thức trên đã thúc đẩy nhu cầu áp dụng các phương pháp, mô hình tiên tiến như học máy để khai thác, đánh giá và xử lý dữ liệu log bay UAV. Học máy không chỉ cải thiện khả năng xử lý dữ liệu đa chiều, phi tuyến mà còn có thể trích xuất đặc trưng ẩn, phát hiện bất thường tinh vi và cải thiện liên tục nhờ cơ chế học tập từ dữ liệu mới.

## 1.3 Bài toán phát hiện lỗi hệ thống điều khiển UAV thông qua log bay

### 1.3.1 Giới thiệu bài toán

Bài toán phát hiện lỗi hệ thống điều khiển UAV thông qua phân tích log bay xuất phát từ yêu cầu cấp thiết trong công tác nâng cao mức độ tin cậy và bảo đảm an toàn vận hành của các hệ thống UAV ngày càng phức tạp. Theo nghiên cứu của Wang và cộng sự (2019) [28], dữ liệu log bay được xem như một nguồn thông tin quý giá, ghi chép đầy đủ các trạng thái hoạt động, hành vi điều khiển và sự kiện hệ thống trong suốt quá trình vận hành UAV. Tuy nhiên, việc chuyển đổi số lượng dữ liệu thông tin khổng lồ này thành những dữ kiện có ý nghĩa để dự báo sớm các sự cố tiềm ẩn vẫn đang là một thách thức kỹ thuật đáng kể [19].

Bài toán phát hiện lỗi UAV từ dữ liệu log bay có thể được xem như một bài toán phân loại đa lớp, trong đó mục tiêu là nhận diện và phân biệt các trạng thái vận hành bình thường với các trạng thái bất thường do lỗi hệ thống gây ra. Trong bối cảnh này, các phương pháp học máy cho thấy tiềm năng lớn nhờ khả năng khai thác mối quan hệ phi tuyến và học đặc trưng trực tiếp từ dữ liệu. Tuy nhiên, hiệu quả của mô hình phụ thuộc mạnh mẽ vào quy trình tiền xử lý dữ liệu, chiến lược gán nhãn, cũng như lựa chọn tập đặc trưng đầu vào phù hợp với bản chất vật lý của hệ thống UAV.

Dữ liệu đầu vào là các tham số thiết yếu của hệ thống bao gồm các số liệu kỹ thuật quan trọng được ghi lại trong thời gian hoạt động.

**Thành phần dữ liệu đầu vào( Input):**

Tập đặc trưng đầu vào bao gồm các nhóm dữ liệu kỹ thuật được trích xuất trực tiếp từ log bay:

- Nhóm đặc trưng cảm biến IMU
  + gyro_rad[0], gyro_rad[1], gyro_rad[2]: Vận tốc góc theo 3 trục (rad/s)
  + gyro_integral_dt: Thời gian tích phân gyro

+ accelerometer_m_s2[0],accelerometer_m_s2[1], accelerometer_m_s2[2]: Gia tốc theo 3 trục ($m/s^2$)
+ accelerometer_integral_dt: Thời gian tích phân accelerometer
+ gyro_clipping: Chỉ số clipping của gyro
+ accel_calibration_count, gyro_calibration_count: Số lần hiệu chuẩn cảm biến

- Nhóm đặc trưng định vị
  + latitude_deg, longitude_deg: Tọa độ địa lý (độ)
  + altitude_msl_m: Độ cao so với mực nước biển (m)
  + altitude_ellipsoid_m: Độ cao so với ellipsoid (m)
- Nhóm đặc trưng thời gian
  + timestamp: Dấu thời gian tuyệt đối
  + timestamp_sample: Thời điểm lấy mẫu
  + timestamp_time_relative: Thời gian tương đối
  + time_utc_us: Thời gian UTC (microseconds)

Theo Wang và cộng sự (2019) [28], các đặc trưng IMU đóng vai trò then chốt trong việc nhận diện các sai lệch bất thường về động lực học của UAV. Trong khi đó, Finn và cộng sự (2024) [10] chỉ ra rằng sự kết hợp giữa dữ liệu định vị và cảm biến quán tính cho phép phát hiện các sai số liên quan đến hệ thống định vị và điều hướng.

**Thành phần dữ liệu đầu ra( Output)**:

Data đầu ra được gán nhãn thành 4 lớp chính:

- Lớp 0 - Bình thường: Hệ thống hoạt động ổn định, tất cả các thông số nằm trong ngưỡng cho phép
- Lớp 1 - Lỗi cảm biến: Bao gồm các hiện tượng drift cảm biến, mất tín hiệu đột ngột, giá trị vượt ngưỡng vật lý [10]. Có thể nhận biết qua một số dấu hiệu như sau: IMU Inconsistency tăng bất thường (PX4), gyro_bias,

accel_bias > ngưỡng thiết kế, sensor_status_imu báo lỗi, giá trị gyro_clipping liên tục > 0.

- Lớp 2 - Lỗi động cơ: Thể hiện qua sự mất cân bằng lực đẩy, dao động tốc độ bất thường, quá nhiệt [8]. Nhận biết qua một vài dấu hiệu trong log như: rung động tăng mạnh trong accelerometer, motor RPM giảm bất thường, accelerometer_m_s2 và gyro_rad thay đổi bất thường trong 1 khoảng thời gian dài.
- Lớp 3 - Lỗi kết nối: Mất kết nối liên lạc, delay truyền dữ liệu vượt ngưỡng, packet loss nghiêm trọng [27]. Nhận biết qua một vài dấu hiệu trong log như: cờ báo mất MAVlink / GPS được bật, dữ liệu truyền ngắt quãng, các giá trị latitude_deg, longitude_deg, altitude_msl_m và cả IMU không có sự biến động hay thay đổi gì trong một khoảng thời gian dài, mặc dù UAV đang di chuyển trong khoảng thời gian đó.

Hướng tiếp cận này tập trung vào việc cải thiện khả năng bảo trì, bảo dưỡng, dự đoán lỗi và cải thiện quy trình vận hành UAV trong thực tế, qua đó giảm đáng kể nguy cơ lỗi và tối đa hoá năng suất khai thác thiết bị.

### 1.3.2 Ứng dụng phương pháp học máy giải quyết bài toán

Việc sử dụng các phương pháp học máy cho bài toán dự đoán lỗi UAV dựa trên log bay được phát triển theo hai hướng chính: phương pháp dựa trên mô hình truyền thống và phương pháp học sâu [11].

Các phương pháp học máy thông thường thường yêu cầu bước trích xuất đặc trưng (feature engineering) thủ công từ dữ liệu log thô trước khi đưa vào mô hình. Ngoài SVM vốn đã được chứng minh hiệu suất cao trong việc xử lý bài toán phân loại đa chiều dữ liệu lớn và có nền tảng lý thuyết vững chắc [7], [13], [1], nhiều phương pháp khác cũng được áp dụng như:

- Cây quyết định (Decision Tree) và Random Forest: Các mô hình này có ưu điểm là dễ diễn giải, cho phép hiểu rõ các đặc trưng nào đóng góp nhiều nhất vào việc phát hiện lỗi [19]. Random Forest, một kỹ thuật ensemble

dựa trên cây quyết định, đặc biệt hiệu quả trong việc giảm hiện tượng overfitting và xử lý dữ liệu nhiễu.

- Hồi quy Logistic (Logistic Regression): Mặc dù là một mô hình đơn giản, nó hoạt động có hiệu suất cao như một phương pháp baseline cho các bài toán phân tích nhị phân và số lượng lớp nhiều, đặc biệt khi mối tương quan giữa các tham số là gần như tuyến tính [22].

Các phương pháp Deep Learning nổi bật với khả năng trích xuất đặc trưng tự động (automatic feature extraction) từ dữ liệu nguyên bản hoặc ít qua xử lý, giúp loại bỏ gánh nặng của feature engineering thủ công và nhận diện các cấu trúc phức tạp mà kỹ thuật thông thường khó phát hiện [11]. Bên cạnh ANN cơ bản đã được Iannace và cộng sự (2019) [14] và Al-Haddad and Jaber (2023) [2] chứng minh hiệu quả, các kiến trúc học sâu chuyên biệt hơn có khả năng được áp dụng:

Mạng neural tích chập (CNN): đầu tiên được thiết kế cho xử lý hình ảnh, CNN có thể được điều chỉnh để phân tích dữ liệu chuỗi thời gian (1D-CNN) bằng cách coi các cửa sổ thời gian của dữ liệu cảm biến như các "bức ảnh" một chiều. Chúng rất tối ưu trong việc tìm ra các mẫu hình cục bộ (local patterns) và các đỉnh bất thường trong tín hiệu [28], [6].

Mạng neural hồi tiếp (RNN) và Long Short-Term Memory (LSTM. Các kiến trúc này được thiết kế để xử lý dữ liệu chuỗi thời gian và có khả năng ghi nhớ các phụ thuộc dài hạn. Chúng đặc biệt phù hợp để mô hình hóa hành vi động lực học của UAV và phát hiện các chuỗi sự kiện bất thường có thể dẫn đến lỗi [9], [24].

Mỗi nhóm phương pháp này mang những ưu/khuyết điểm riêng biệt. Các mô hình truyền thống như SVM và Random Forest có ưu điểm về tốc độ huấn luyện, tiêu thụ ít tài nguyên tính toán và khả năng diễn giải mô hình tốt, do đó phù hợp với các tập dữ liệu có quy mô vừa và nhỏ. Ngược lại, các mô hình học sâu như CNN và LSTM sở hữu năng lực biểu diễn mạnh mẽ các mối quan hệ phi tuyến phức tạp và khả năng tự động trích xuất đặc trưng, nhưng đòi hỏi quy mô dữ liệu rất lớn và tài nguyên xử lý mạnh mẽ.

Sau khi xem xét toàn diện, đề án lựa chọn phương pháp SVM và ANN để tập trung nghiên cứu và triển khai dựa trên bằng chứng khoa học. Về mặt đại diện phương pháp luận, SVM là minh chứng nổi bật cho các mô hình truyền thống với nền tảng học thuật vững chắc và hiệu quả đã được kiểm chứng qua nhiều thử nghiệm, trong khi ANN được xem là nền tảng cơ bản của học sâu, tạo cơ sở cho việc mở rộng sang các kiến trúc phức tạp hơn trong tương lai. Trong điều kiện hiện nay ở Việt Nam, nơi nguồn dữ liệu log UAV được gán nhãn còn hạn chế và tài nguyên tính toán không dồi dào, việc lựa chọn SVM với khả năng hiệu quả trên dữ liệu cỡ vừa, ANN với cấu trúc linh hoạt, không yêu cầu lượng dữ liệu lớn như các mô hình sâu phức tạp, thể hiện tính khả thi cao. Mặt khác, việc triển khai song song cả hai phương pháp tạo điều kiện cho một sự so sánh có ý nghĩa giữa hai trường phái học máy trên cùng một bộ dữ liệu, từ đó cung cấp những phân tích khách quan để xác định mức độ thích nghi của từng phương pháp trong bối cảnh ứng dụng cụ thể tại Việt Nam.

### 1.3.3 Tập dữ liệu log bay UAV cho bài toán

Trong lĩnh vực nghiên cứu UAV, một số tập dữ liệu log bay công khai đã được nghiên cứu, thiết kế và ứng dụng rộng rãi. Tập dữ liệu PX4 Flight Review là một trong những kho lưu trữ log bay lớn nhất, được phát triển và gìn giữ bởi cộng đồng PX4 từ năm 2015. Tập dữ liệu này chứa hơn 50.000 chuyến bay từ nhiều phiên bản firmware khác nhau, được lưu trữ dưới định dạng ULog với đầy đủ thông tin về cảm biến và trạng thái hệ thống. Tập dữ liệu này được thiết lập để hỗ trợ đánh giá hiệu suất, gỡ lỗi và xây dựng các giải pháp thuật toán nhằm xác định lỗi trong hệ thống UAV.

Tập dữ liệu UAV Fault and Anomaly Detection (UFAD) do Đại học Stanford xây dựng từ năm 2019. Tập dữ liệu này tập trung chủ yếu vào các tình huống lỗi được tạo ra có chủ đích, với 1.200 chuyến bay và 15 loại lỗi khác nhau. UFAD cung cấp các file log kèm theo nhãn lỗi chi tiết và thời điểm xảy ra sự cố, mang lại giá trị lớn cho việc đào tạo và thẩm định các mô hình học máy trong phát hiện lỗi.

Tập dữ liệu ArduPilot Log Repository cũng đáng chú ý với khoảng 30.000 file log từ nhiều nền tảng hardware khác nhau. Được phát triển từ năm 2016 bởi cộng đồng ArduPilot, tập dữ liệu này chủ yếu phục vụ mục đích kiểm tra, thử nghiệm và phát triển firmware, với định dạng DataFlash tập trung vào các chỉ số hiệu suất và tình trạng hệ thống.

Sau khi phân tích kỹ lưỡng các tập dữ liệu có sẵn, đề án quyết định xây dựng tập dữ liệu riêng dựa trên định dạng chuẩn PX4. Lựa chọn này dựa trên nhiều yếu tố quan trọng, trong đó tính phù hợp được thể hiện qua việc định dạng ULog chuẩn hóa của PX4 tương thích hoàn toàn với mục tiêu nghiên cứu và hệ thống UAV sử dụng trong đề án. Hơn nữa, tập dữ liệu này cung cấp đầy đủ các luồng dữ liệu cần thiết cho bài toán phát hiện lỗi. Về tính đầy đủ, tập dữ liệu PX4 Flight Review cung cấp số lượng mẫu phong phú qua nhiều yếu tố vận hành khác nhau, bao gồm cả dữ liệu bình thường và bất thường. Các metadata đi kèm về cấu hình phần cứng và điều kiện bay được ghi chép đầy đủ, tạo nền tảng vững chắc cho việc phân tích và đánh giá. Đặc biệt, khả năng mở rộng của định dạng ULog cho phép dễ dàng tích hợp thêm các dữ liệu cảm biến mới và gộp với các tập dữ liệu khác. Để tăng cường tính thực tiễn, dữ liệu sẽ được bổ sung thông qua các chuyến bay thử nghiệm trong điều kiện đặc thù của Việt Nam, nhằm phản ánh chân thực các tình huống vận hành thực tế.

## 1.4 Các nghiên cứu liên quan

Ngày càng nhiều nghiên cứu phân tích dữ liệu UAV ứng dụng học máy, dựa trên nhiều cách tiếp cận khác nhau.

Một trong những cách thức nghiên cứu ban đầu chú trọng vào việc giám sát các cảm biến trọng yếu. Nghiên cứu của Meier và cộng sự [23] phát triển một hệ thống phân tích log bay thời gian thực cho nền tảng autopilot PX4 với trọng tâm là phát hiện lỗi cảm biến IMU. Bằng cách đưa ra các giải pháp thống kê để thiết lập đường cơ sở cho hoạt động bình thường, hệ thống này có thể phát hiện các sai lệch tín hiệu ngay lập tức, chứng minh tính khả thi của việc giám sát và chẩn đoán lỗi theo thời gian thực trên các nền tảng UAV thông dụng.

Khi các bài toán trở nên phức tạp hơn, các mô hình học sâu (deep learning) được áp dụng để tự động trích xuất đặc trưng và phát hiện các mẫu lỗi tinh vi từ một loạt các tham số. Thay vì dựa vào một mô hình đơn lẻ, phương pháp học tập kết hợp (ensemble learning) vốn đã có cơ sở lý thuyết vững chắc từ các thuật toán như SVM [7], [13], [3], đã chứng minh hiệu quả trong việc kết hợp nhiều bộ phân loại để cải thiện độ chính xác, như được minh họa trong bài toán phát hiện UAV [19]. Tiến xa hơn, các kiến trúc học sâu chuyên sâu đã được đề xuất để xử lý dữ liệu log. Chẳng hạn, Wang và cộng sự [28] sử dụng mô hình dự đoán học sâu để phát hiện bất thường trong dữ liệu cảm biến, trong khi Debele và cộng sự [9] phát triển một sơ đồ học sâu mạnh mẽ để phát hiện và cách ly lỗi actuator trên UAV đa rotor.

Xu hướng gần đây nhất hướng đến việc tích hợp các nguồn data khác nhau để có được tổng hợp đầy đủ hơn về tình trạng của UAV. Nghiên cứu của Chen và Wang [6] là một bước tiến trong hướng này, khi đề xuất một mô hình kết hợp CNN với cơ chế 'tự chú ý' (Attention) để phân tích đồng thời cả dữ liệu log bay và ảnh nhiệt. Trong đó, CNN đảm nhiệm việc trích xuất đặc trưng trực quan từ ảnh nhiệt, còn cơ chế "chú ý" (Attention) giúp mô hình đặt trọng tâm vào các đặc điểm quan trọng và kết hợp thông tin đa phương thức một cách tối ưu. Từ phương pháp này có thể xác định chuẩn xác hiện tượng quá nhiệt động cơ, một vấn đề mà các phương pháp dựa trên cảm biến đơn lẻ thường khó phát hiện một cách đáng tin cậy, qua đó mở ra một hướng tiếp cận mới nhằm cải thiện hơn nữa mức độ đáng tin cậy của hệ thống giám sát UAV.

## 1.5 Kết luận chương 1

Chương 1 đã hoàn thành việc tạo lập nền tảng lý thuyết toàn diện cho đề án nghiên cứu. Bằng việc phân tích hệ thống điều khiển UAV với các thành phần chính và những dạng lỗi đặc trưng, chương này đã làm rõ yêu cầu cần thiết của việc phát triển giải pháp dự đoán lỗi thông minh. Dữ liệu log bay UAV đã được khẳng định là nguồn thông tin quý giá, dù tiềm ẩn những thách thức về xử lý do tính phức tạp và đa dạng.

Bài toán dự đoán lỗi đã được xác định rõ với đầu vào là các đặc trưng kỹ thuật từ log bay và đầu ra là bốn lớp trạng thái hệ thống. Việc ứng dụng các phương pháp học máy như SVM và ANN được đề xuất như hướng tiếp cận khả thi để tận dụng có hiệu quả nguồn dữ liệu log bay. Các nghiên cứu đưa ra đã cung cấp bằng chứng chứng tỏ một cách thuyết phục rằng hướng nghiên cứu này là khả thi và có giá trị ứng dụng cao.

Những nội dung này tạo cơ sở vững chắc cho việc triển khai các phương pháp cụ thể và đánh giá hiệu quả giải pháp trong các phần tiếp theo của đề án.

# CHƯƠNG 2: NGHIÊN CỨU MỘT SỐ PHƯƠNG PHÁP HỌC MÁY

*Chương này tập trung phân tích hai phương pháp học máy chính là SVM phân loại đa lớp (Multi-class SVM) và mạng neuron nhân tạo cho bài toán phát hiện và dự đoán lỗi hệ thống UAV từ log bay.*

## 2.1 Phương pháp SVM phân loại đa lớp (Multi-class SVM)

Việc phân loại trạng thái hoạt động của UAV dựa trên log bay đòi hỏi mô hình có thể thực hiện khai thác và xử lý dữ liệu phi tuyến mạnh mẽ, phân bố chồng lấp và chứa nhiễu cảm biến đặc trưng của các hệ thống bay thực tế. Trong khuôn khổ đề án này, phương pháp SVM với kernel RBF được lựa chọn nhờ tính chất hiệu quả trong tối ưu hóa biên phân tách và năng lực biểu diễn các quan hệ phi tuyến trong dữ liệu cảm biến. Để mở rộng SVM cho nhiệm vụ nhiều lớp, đề án áp dụng thêm chiến lược One-vs-Rest (OvR), một kỹ thuật phân rã bài toán đa lớp thành nhiều mô hình nhị phân.

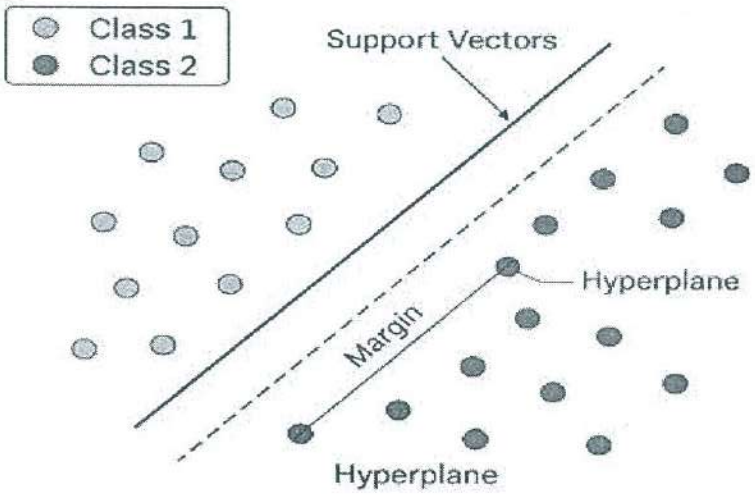
### 2.1.1 Nguyên lý hoạt động

Mô hình Multi-class SVM sử dụng kernel RBF vận hành dựa trên sự tích hợp của ba thành phần cốt lõi: nguyên lý siêu phẳng tối ưu cho bài toán nhị phân, kỹ thuật kernel cho data phi tuyến, và chiến lược mở rộng cho bài toán đa lớp.

Nguyên lý cơ bản của SVM là xác định một siêu phẳng (hyperplane) tối ưu trong không gian đặc trưng sao cho có thể phân chia các lớp dữ liệu. Siêu phẳng này được thiết kế để tối đa hóa khoảng cách (margin) tới các điểm dữ liệu gần nhất thuộc cả hai lớp [5]. Những điểm dữ liệu nằm trên biên của khoảng cách này được gọi là vector hỗ trợ (support vectors), đóng vai trò quan trọng trong việc quyết định vị trí của siêu phẳng phân tách. Nguyên lý này được mô tả trong Hình 2.1.

Khi dữ liệu không thể phân tách tuyến tính trong không gian ban đầu, SVM sử dụng một kỹ thuật gọi là "kernel trick" [7], [2]. Kỹ thuật kernel cho phép biểu diễn dữ liệu trong không gian đặc trưng nhiều chiều hơn, tại đây dữ liệu có khả năng được

phân tách bằng một siêu phẳng tuyến tính. Việc tính toán trong không gian mới này được thực hiện nhờ vào các hàm kernel (như linear, polynomial, hoặc radial basis function - RBF). Điều này giúp SVM có thể xử lý các bài toán phân loại với đường biên phân lớp phi tuyến phức tạp.




**Hình 2-1 Minh họa nguyên lý hoạt động của SVM**

*( Nguồn: Cortes, C., and Vapnik, V. N. (1995). Support-vector networks. Machine Learning, 20(3), 273–297)*

Giả sử tập huấn luyện gồm các cặp dữ liệu:

$$(x_i, y_i), i = 1,2,3, \dots, n \qquad y_i \in \{-1, 1\} \qquad (2.1)$$

Mục tiêu là tìm một siêu phẳng có dạng:

$$w^T x + b = 0 \qquad (2.2)$$

Thỏa mãn:

$$y_i(w^T x_i + b) \geq 1, \ \forall i \qquad (2.3)$$

Trong đó:

- w: vector trọng số xác định hướng của siêu phẳng,
- b: hệ số dịch chuyển,
- $y_i$: nhãn lớp (+1 hoặc -1).

### 2.1.2 Kernel

Trong mô hình SVM, hàm kernel là công cụ giúp ánh xạ dữ liệu đầu vào từ không gian đặc trưng gốc (input space) sang không gian đặc trưng có chiều cao hơn (feature space), nơi mà các mẫu dữ liệu trở nên tuyến tính phân tách được. Một số hàm kernel phổ biến:

Kernel tuyến tính (Linear): Trường hợp đơn giản, được xác định bởi tích vô hướng của hai vector:

$$K(x_i, x_j) = {x_i}^T x_j \quad (2.4)$$

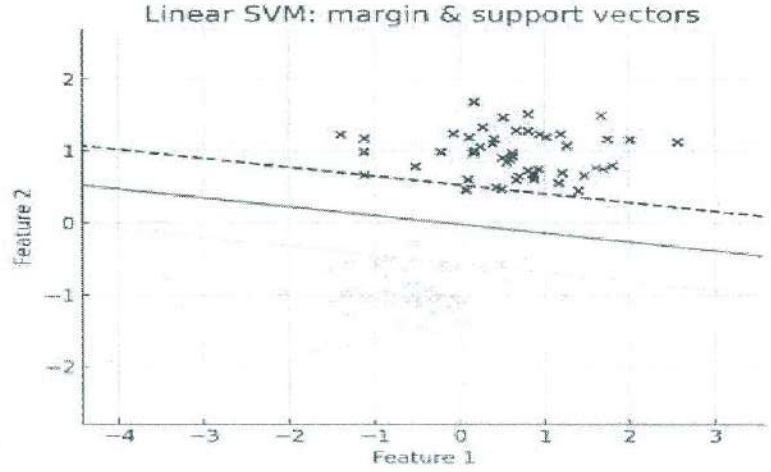
Radial Basic Function (RBF): kernel này được áp dụng rộng rãi trong thực tế và thường được chọn làm mặc định trong thư viện sklearn. Nó được định nghĩa bởi:

$$K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel_2^2), \quad \gamma > 0 \qquad (2.5)$$

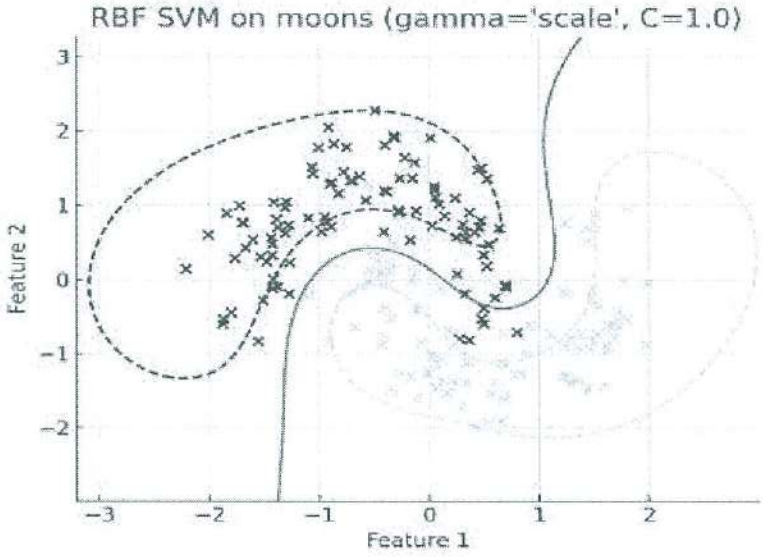
Sigmoid

$$K(x_i, x_j) = \tanh(\gamma {x_i}^T x_j + r) \qquad (2.6)$$

Hình 2.2 và 2.3 mô tả sự khác biệt giữa SVM tuyến tính và phi tuyến (RBF kernel) trong đó RBF cho phép mô hình "uốn cong" biên phân tách để phù hợp hơn với dữ liệu phức tạp:




**Hình 2-2 SVM tuyến tính**




**Hình 2-3 SVM đa lớp sử dụng kernel RBF**

Bài toán phát hiện lỗi dựa trên dữ liệu log bay UAV, các đặc trưng đầu vào (như vận tốc góc, gia tốc, sai lệch điều khiển, tín hiệu IMU) có mối quan hệ phi tuyến mạnh và chịu ảnh hưởng đồng thời của nhiễu cảm biến, điều kiện môi trường và trạng thái điều khiển. Kernel tuyến tính trong SVM chỉ phù hợp khi dữ liệu có thể được phân tách bằng siêu phẳng trong không gian đầu vào ban đầu. Tuy nhiên, đối với dữ liệu log bay UAV, các trạng thái bình thường và bất thường thường chồng lấn mạnh, khiến việc phân tách tuyến tính trở nên kém hiệu quả. Kernel RBF cho phép ánh xạ dữ liệu từ không gian đầu vào sang một không gian đặc trưng có chiều cao hơn, nơi các ranh giới phân tách phi tuyến có thể được xây dựng hiệu quả hơn.

RBF có tính cục bộ cao, nghĩa là ảnh hưởng của mỗi điểm dữ liệu chủ yếu tập trung trong một vùng lân cận nhất định. Đặc điểm này phù hợp với bài toán phát hiện lỗi UAV, trong đó các lỗi thường xuất hiện dưới dạng sai lệch cục bộ hoặc bất thường ngắn hạn trong chuỗi log, thay vì sự thay đổi toàn cục của toàn bộ trạng thái bay.

RBF không yêu cầu giả định trước về dạng phân bố của dữ liệu đầu vào, giúp mô hình thích nghi tốt hơn với dữ liệu log thực tế vốn có phân bố phức tạp và không ổn định. Do đó, việc lựa chọn kernel RBF giúp cân bằng giữa khả năng biểu diễn phi tuyến và độ tổng quát hóa của mô hình, phù hợp với mục tiêu phát hiện lỗi trong đề án.

### 2.1.3 Các siêu tham số

Mô hình SVM đa lớp sử dụng kernel RBF có ba siêu tham số quan trọng cần được tối ưu hóa:

Tham số C (penalty parameter) điều khiển sự đánh đổi giữa việc cực đại hóa lề và việc cho phép các điểm dữ liệu bị phân loại sai. Giá trị C lớn sẽ dẫn đến một mô hình phức tạp hơn với ít điểm bị phân loại sai hơn trên tập huấn luyện, nhưng có thể dẫn đến hiện tượng overfitting [7].

Tham số $\gamma$ điều chỉnh mức độ ảnh hưởng cục bộ của từng điểm huấn luyện: giá trị lớn dẫn tới biên quyết định phức tạp hơn và nhạy với nhiễu, trong khi giá trị nhỏ tạo ra biên mềm mại hơn. Nhờ cơ chế kernel trick, mô hình có thể khai thác các đặc trưng phi tuyến mà không làm tăng chi phí tính toán đáng kể, phù hợp với dữ liệu đa cảm biến và giàu nhiễu của UAV [28], [31].

Do SVM nguyên thủy chỉ hỗ trợ phân loại hai lớp, đề án áp dụng chiến lược One-vs-Rest để xử lý bài toán bốn lớp lỗi UAV. Phương pháp OvR xây dựng K bộ phân loại nhị phân tương ứng với K lớp, trong đó mỗi bộ phân loại tách một lớp ra khỏi toàn bộ các lớp còn lại. Cụ thể, với bốn lớp (bình thường, lỗi cảm biến, lỗi động cơ, lỗi kết nối), mô hình tạo ra bốn bộ phân loại độc lập. Khi dự đoán, mỗi bộ phân loại đưa ra một giá trị hàm quyết định (decision function), và mẫu sẽ được gán vào lớp có giá trị cao nhất, tương ứng với mức độ "tự tin" lớn nhất của mô hình. Cách triển khai này có tính đơn giản, ổn định và đã được chứng minh năng suất trong nhiều ứng dụng nhận dạng đa lớp [13], [25].

### 2.1.4 Ưu nhược điểm

- Ưu điểm:

SVM thể hiện khả năng hoạt động tốt ngay cả khi số lượng đặc trưng vượt quá số lượng mẫu huấn luyện phù hợp với dữ liệu cảm biến UAV vốn có nhiều thông số đầu vào. Theo Cortes và Vapnik [13], mô hình SVM có khả năng tối ưu trong không

gian đặc trưng nhiều chiều nhờ cơ chế xây dựng siêu phẳng phân tách dựa trên các véc-tơ hỗ trợ.

Với nguyên lý maximum margin, SVM tìm được mặt phẳng phân cách tối ưu giữa các lớp dữ liệu, giúp giảm thiểu hiện tượng overfitting và tăng khả năng tổng quát hóa cho dữ liệu chưa từng gặp. Burges [5] đã trình bày cơ sở toán học của nguyên lý này và chứng minh khả năng khái quát hóa mạnh của SVM trong các bài toán nhận dạng mẫu.

Một điểm đáng lưu ý của SVM là vẫn duy trì được hiệu suất phân loại cao khi kích thước tập dữ liệu huấn luyện bị giới hạn. Youseffi và cộng sự.[33] đã chứng minh tính hiệu quả này khi áp dụng SVM vào bài toán phát hiện lỗi UAV dựa trên dữ liệu trên dữ liệu thu thập qua mạng đám mây.

Nền tảng lý thuyết học thống kê giúp SVM duy trì tính ổn định và khả năng kiểm chứng, điều này đặc biệt cần thiết trong các hệ thống điều khiển UAV có yêu cầu độ tin cậy cao.

- Nhược điểm:

Do SVM ban đầu được thiết kế chủ yếu cho bài toán phân loại nhị phân, việc mở rộng sang phân loại nhiều lớp yêu cầu sử dụng các kỹ thuật phức tạp như one-vs-one hoặc one-vs-all, làm tăng độ phức tạp tính toán. Hsu và Lin [6] đã so sánh các phương pháp này và chỉ ra những hạn chế cố hữu của SVM khi xử lý dữ liệu có nhiều lớp.

Độ phức tạp tính toán của SVM tăng nhanh theo kích thước dữ liệu, dẫn đến thời gian huấn luyện dài khi train cho các tập log UAV có quy mô lớn. Yousefi và cộng sự [31] đã chỉ ra việc triển khai SVM trên môi trường xử lý đám mây cho UAV có thể gặp giới hạn về thời gian xử lý khi khối lượng dữ liệu tăng.

Hiệu quả của SVM đa lớp phụ thuộc đáng kể vào việc lựa chọn hàm nhân kernel cũng như các siêu tham số như C và $\gamma$. S.Abe [1] đã thảo luận về sự phụ thuộc này và các phương pháp lựa chọn tham số.

## 2.2 Phương pháp mạng nơ ron nhân tạo

Mạng neuron nhân tạo (ANN) là một mô hình tính toán với thiết kế mô phỏng cơ chế hoạt động của bộ não người. ANN bao gồm nhiều "neuron nhân tạo" (artificial neurol) liên kết với nhau cho phép xử lý dữ liệu, học từ ví dụ và đưa ra tính toán các dự đoán hoặc phân loại [33].

### 2.2.1 Nguyên lý hoạt động

Một ANN cơ bản gồm ba loại tầng:

- Tầng đầu vào (Input Layer): Nhận dữ liệu đầu vào dưới dạng vector đặc trưng (features).
- Tầng ẩn (Hidden Layers): Xử lý tín hiệu thông qua các phép nhân trọng số, cộng thiên lệch, và áp dụng hàm kích hoạt.
- Tầng đầu ra (Output Layer): Cung cấp kết quả dự đoán (ví dụ: phân loại nhãn, giá trị dự báo).

Mỗi neuron trong một tầng được kết nối với các neuron ở tầng phía sau bằng trọng số (weights). Tín hiệu (signal) từ các neuron lớp trước được nạp vào mỗi neuron, tính toán tuyến tính theo trọng số và bias, rồi truyền qua hàm kích hoạt phi tuyến để sinh đầu ra cho lớp tiếp theo [32].

Quá trình lan truyền xuôi (Forward Propagation):

$$z_j = \sum_i w_{ij} x_i + b_j \qquad (2.7)$$

$$a_j = f(z_j) \qquad (2.8)$$

Trong đó:

- $x_i$ là đầu vào
- $w_{ij}$ là trọng số
- $b_j$ là hệ số bù bias

- $f(z_j)$ là hàm kích hoạt, giúp mạng học được các quan hệ phi tuyến.

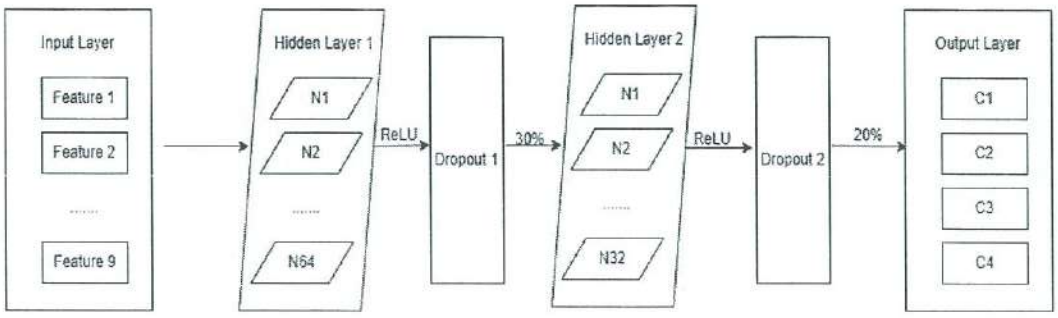
Sau khi tính toán đầu ra, mô hình so sánh kết quả dự đoán với giá trị thật để tính sai số (error). Để giảm sai số này, ANN thực hiện quá trình lan truyền ngược (backpropagation) tức là cập nhật trọng số thông qua thuật toán tối ưu (thường là gradient descent). Trọng số được điều chỉnh theo hướng giảm dần sai số qua từng vòng huấn luyện (epoch).

Quy trình hoạt động tổng thể của ANN bao gồm hai giai đoạn chính:

- Lan truyền xuôi (Forward Propagation): dữ liệu được truyền từ lớp đầu vào qua các lớp ẩn đến lớp đầu ra để tạo dự đoán.
- Lan truyền ngược (Backpropagation): sai số giữa giá trị dự đoán và giá trị thật được tính toán và dùng để cập nhật trọng số, giúp mô hình học và cải thiện dần độ chính xác.

### 2.2.2 Cấu trúc mạng đề xuất

Nhằm giải quyết vấn đề phân loại lỗi từ dữ liệu log bay, đề án đề xuất một kiến trúc mạng nơ ron nhân tạo truyền thống. Mô hình được xây dựng trên nguyên lý mạng truyền thẳng với cấu trúc lớp ẩn đa tầng, cho phép học các biểu diễn phi tuyến từ tập đặc trưng đầu vào được minh họa trên hình 2.4.




**Hình 2-4 Mô hình mạng ANN đề xuất**

Kiến trúc mạng được triển khai theo mô hình tuần tự, bao gồm 4 lớp chính, được mô tả chi tiết như sau:

- Lớp đầu vào (Input Layer): bao gồm 9 đặc trưng được trích xuất từ dữ liệu log. Trước khi được đưa vào mạng, dữ liệu được chuẩn hóa để đảm bảo độ ổn định trong quá trình đào tạo.
- Lớp ẩn thứ nhất: bao gồm 64 nơ ron, sử dụng hàm kích hoạt ReLU nhằm tăng khả năng học và biểu diễn các quan hệ phi tuyến trong dữ liệu. Sau lớp này, một Dropout được áp dụng để hạn chế hiện tượng overfitting bằng cách ngẫu nhiên vô hiệu hóa 30% số neuron trong quá trình học.
- Lớp ẩn thứ hai: bao gồm 32 neuron, dùng hàm kích hoạt ReLU. Sau đó, dùng dropout 20% để tăng cường khả năng mô hình hóa tổng quát trên các dữ liệu mới chưa từng thấy.
- Lớp đầu ra: bao gồm 4 neuron ứng với 4 trạng thái lỗi của UAV. Hàm kích hoạt được sử dụng là Softmax, cho phép mô hình dự đoán xác suất của từng lỗi.

Kiến trúc mạng được thiết kế với ba tầng ẩn, một lựa chọn được cân nhắc kỹ lưỡng dựa trên bản chất của tín hiệu cảm biến UAV. Dữ liệu log bay là một chuỗi thời gian đa chiều với các mối tương quan động lực học phức tạp giữa các biến như gia tốc, vận tốc góc, và vị trí. Một tầng ẩn đơn lẻ thường không đủ sâu để trích xuất các đặc trưng cấp cao từ sự tương tác này. Ngược lại, một kiến trúc quá sâu với số lượng tầng vượt quá bốn, khi được huấn luyện trên tập dữ liệu có quy mô khiêm tốn (vài nghìn mẫu), sẽ dễ dàng học các mẫu nhiễu cụ thể thay vì các quy luật tổng quát, dẫn đến hiệu suất suy giảm nghiêm trọng trên dữ liệu mới. Do đó, ba tầng ẩn đại diện cho điểm cân bằng tối ưu, cho phép mô hình xây dựng một hệ thống biểu diễn phân cấp: tầng đầu trích xuất các đặc trưng cục bộ cơ bản, tầng thứ hai kết hợp chúng thành các mẫu phức tạp hơn, và tầng thứ ba chuẩn bị cho việc ánh xạ cuối cùng sang không gian các lớp lỗi.

### 2.2.3 Hàm kích hoạt

Các hàm kích hoạt giúp mạng học được các quan hệ phi tuyến phức tạp.

- ReLU( Rectified Linear Unit)

$$f(x) = \max(0, x) \qquad (2.9)$$

- Sigmoid ( cho phân loại 2 lớp):

$$f(x) = \frac{1}{1 + e^{-x}} \qquad (2.10)$$

- Softmax (cho phân loại nhiều lớp):

$$f_i(x) = \frac{e^{x_i}}{\sum_j e^{x_j}} \qquad (2.11)$$

Trong mạng neuron nhân tạo, hàm kích hoạt quyết định khả năng học các quan hệ phi tuyến và ảnh hưởng trực tiếp đến tốc độ cũng như độ ổn định của quá trình huấn luyện. Đối với bài toán phân tích log bay UAV, dữ liệu đầu vào có biên độ thay đổi lớn, chứa nhiều dao động ngắn hạn và nhiễu cảm biến, do đó yêu cầu hàm kích hoạt vừa đơn giản vừa hiệu quả.

Các hàm kích hoạt như sigmoid hoặc tanh có miền giá trị bị giới hạn, dễ dẫn đến hiện tượng vanishing gradient khi huấn luyện mạng nhiều lớp, khiến quá trình học trở nên chậm và kém hiệu quả. Trong khi đó, ReLU không bị bão hòa ở miền dương, giúp gradient được duy trì ổn định hơn và tăng tốc độ hội tụ của mô hình.

ReLU chỉ kích hoạt khi đầu vào vượt qua một ngưỡng nhất định, đặc điểm này phù hợp với bài toán phát hiện lỗi UAV, trong đó các trạng thái bất thường thường chỉ xuất hiện khi các tham số bay vượt quá giới hạn vận hành bình thường. Nhờ đó, ReLU giúp mô hình tập trung vào các biến động có ý nghĩa thay vì phản ứng với nhiễu nhỏ.

Với chi phí tính toán thấp và khả năng mở rộng tốt, ReLU là lựa chọn phù hợp cho cấu trúc mạng ANN trong khuôn khổ đề án.

### 2.2.4 Siêu tham số cần tối ưu

Việc xác định bộ siêu tham số tối ưu cho kiến trúc ANN được thực hiện thông qua phân tích có hệ thống, dựa trên đặc tính nội tại của dữ liệu log bay và các nguyên

lý thiết kế mô hình học sâu nhằm đảm bảo hiệu quả phát hiện lỗi. Mỗi quyết định về siêu tham số đều hướng đến sự cân bằng giữa ba yếu tố then chốt: năng lực biểu diễn của mô hình để nắm bắt các quan hệ phi tuyến phức tạp, khả năng tổng quát hóa để tránh hiện tượng học quá khớp (overfitting), và hiệu suất tính toán trong điều kiện tài nguyên có hạn.

Về số lượng neuron trong mỗi lớp, nghiên cứu áp dụng nguyên tắc kiến trúc "hình nón" (64 → 32 → 4). Lớp đầu vào với 64 neuron được thiết kế để có đủ dung lượng tiếp nhận và xử lý sơ bộ 9 đến 12 đặc trưng đầu vào. Lớp ẩn thứ hai với 32 neuron thực hiện nhiệm vụ nén thông tin và trích xuất các đặc trưng cấp cao hơn, đóng vai trò như một cầu nối trung gian. Lớp đầu ra gồm 4 neuron tương ứng trực tiếp với bốn lớp mục tiêu cần phân loại. Cấu trúc thu hẹp dần này phù hợp với luồng xử lý thông tin từ chi tiết đến tổng hợp, đồng thời kiểm soát được độ phức tạp tổng thể của mô hình.

Tốc độ học (learning rate) được thiết lập trong khoảng từ 0.001 đến 0.005. Giá trị này được xác định thông qua thực nghiệm sơ bộ và cân nhắc đặc tính của thuật toán tối ưu Adam. Một tốc độ học quá lớn có thể khiến quá trình tối ưu phân kỳ hoặc dao động quanh điểm tối ưu, trong khi một tốc độ học quá nhỏ sẽ kéo dài thời gian huấn luyện một cách không cần thiết. Phạm vi 0.001- 0.005 đảm bảo sự ổn định và tốc độ hội tụ phù hợp cho bài toán này.

Để giảm thiểu nguy cơ overfitting, kỹ thuật Dropout được áp dụng với tỷ lệ từ 0.3 đến 0.5 giữa các lớp ẩn. Dropout hoạt động bằng cách ngẫu nhiên vô hiệu hóa một tỷ lệ neuron trong quá trình huấn luyện, buộc mạng phải học các biểu diễn dự phòng và tổng quát hơn. Tỷ lệ 0.2- 0.5 là mức độ regularization vừa phải, đủ để ngăn chặn sự phụ thuộc quá mức giữa các neuron mà không làm suy giảm đáng kể khả năng học của mạng.

Về quy trình huấn luyện, kích thước batch được đặt trong khoảng 16 đến 64. Với tập dữ liệu có quy mô vài nghìn mẫu, batch size trong phạm vi này cho phép ước lượng gradient đủ ổn định để hội tụ về nghiệm tốt, đồng thời đảm bảo số lần cập nhật

trọng số đủ nhiều trong mỗi epoch. Số epoch được giới hạn từ 20 đến 50, kết hợp với cơ chế dừng sớm (early stopping) dựa trên hiệu suất tập validation. Giới hạn này ngăn không cho mô hình huấn luyện quá lâu, dẫn đến overfitting trên tập huấn luyện, đặc biệt quan trọng với dữ liệu có tính lặp và chứa nhiễu như log bay.

Thuật toán tối ưu Adam được lựa chọn thay cho SGD truyền thống nhờ khả năng tự động điều chỉnh tốc độ học cho từng tham số và tích hợp cơ chế động lượng (momentum). Điều này đặc biệt hữu ích khi các đặc trưng đầu vào có phạm vi giá trị chênh lệch lớn, giúp quá trình huấn luyện hội tụ nhanh hơn và ổn định hơn.

Bộ siêu tham số được đề xuất là kết quả của việc cân nhắc cẩn trọng giữa độ phức tạp của bài toán phân loại đa lớp từ dữ liệu chuỗi thời gian nhiễu, đặc điểm quy mô vừa phải của tập dữ liệu, và các nguyên tắc thiết kế nhằm tối ưu hóa khả năng tổng quát hóa của mô hình. Các giá trị này đóng vai trò là điểm xuất phát cho quá trình tinh chỉnh dựa trên kết quả thực nghiệm trong chương tiếp theo.

### 2.2.5 Ưu điểm và hạn chế

- Ưu điểm

ANN có thể học trực tiếp các đặc trưng từ dữ liệu phức tạp mà không cần kỹ thuật trích chọn thủ công. Lannace và cộng sự [21] chỉ ra rằng ANN có thể tự nhận diện các đặc trưng quan trọng từ dữ liệu log UAV, hỗ trợ chuẩn đoán lỗi mà không đòi hỏi tác động thủ công trong quá trình trích xuất đặc trưng.

ANN xử lý tốt dữ liệu đa chiều từ nhiều nguồn cảm biến khác nhau (IMU, GPS, cảm biến nhiệt độ, áp suất) trong hệ thống UAV. Theo Al-Haddad and Jaber [17] đã đánh giá hiệu suất của mạng DNN trong chuẩn đoán lỗi dựa trên các đặc trưng đa độ phân giải, đạt độ chính xác 96,7%, góp phần nâng cao độ tin cậy trong giám sát UAV.

ANN có thể được train với lượng dữ liệu lớn từ các chuyến bay và có thể mở rộng linh hoạt khi có thêm dữ liệu mới. Delebe và cộng sự [9] đã huấn luyện ANN trên 50000 mẫu dữ liệu bay, minh chứng cho khả năng mở rộng của mô hình.

- Hạn chế

ANN đòi hỏi một lượng lớn dữ liệu huấn luyện cùng nhãn chính xác để hoạt động hiệu quả [9]. Điều này đặt vấn đề thách thức trong việc tìm kiếm các dữ liệu vận hành lỗi của các hệ thống UAV trong thực tế.

ANN có độ phức tạp tính toán cao, đòi hỏi tài nguyên phần cứng mạnh trong quá trình huấn luyện và suy luận [21]. Đây là yếu tố cần cân nhắc khi triển khai trên các nền tảng UAV có giới hạn về năng lực xử lý.

Mô hình ANN mang tính "hộp đen" khó giải thích kết quả nội tại. Pithode và Patheja [34] chỉ ra rằng đặc tính này làm tăng độ khó trong việc xác định nguyên nhân và mức độ lỗi khi áp dụng cho khả năng phát hiện bất thường từ log UAV.

Hiệu suất của ANN phụ thuộc phần lớn vào việc thiết lập các siêu tham số như learning rate, số lớp ẩn, batch size, dropout, v.v. Nếu không được tinh chỉnh cẩn thận, mô hình dễ rơi vào tình trạng không hội tụ hoặc đạt độ chính xác thấp.

## 2.3 Kết luận chương 2

Chương 2 đã hoàn thiện nền tảng lý thuyết cho hai phương pháp học máy được lựa chọn: SVM đa lớp với kernel RBF và mạng nơ-ron nhân tạo (ANN) sử dụng hàm kích hoạt ReLU và kỹ thuật Dropout. Các phân tích về nguyên lý hoạt động, ưu nhược điểm và cơ sở lựa chọn siêu tham số cho từng phương pháp đã được trình bày hệ thống.

Những quyết định thiết kế và phạm vi tham số được đề xuất trong chương này không chỉ dựa trên đặc thù của dữ liệu log bay UAV- vốn có tính phi tuyến cao và chứa nhiều nhiễu - mà còn hướng đến việc cân bằng giữa khả năng biểu diễn mô hình và nguy cơ học quá khớp. Cụ thể, việc lựa chọn kernel RBF cho SVM xuất phát từ nhu cầu xử lý dữ liệu phi tuyến, trong khi kiến trúc ANN với ba lớp ẩn và cơ chế Dropout được thiết kế để học các đặc trưng phức tạp từ dữ liệu cảm biến đa chiều.

Các kết quả phân tích lý thuyết này sẽ trực tiếp định hướng và làm cơ sở cho toàn bộ quy trình thực nghiệm trong Chương 3. Kiến trúc mô hình và phạm vi siêu

tham số đã đề xuất sẽ được triển khai cụ thể, tối ưu hóa và đánh giá trên bộ dữ liệu log bay thực tế. Sự so sánh bản chất giữa hai phương pháp cũng tạo cơ sở để phân tích và lý giải các kết quả thực nghiệm về sau.

# CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

*Chương 3 triển khai thực nghiệm để đánh giá hiệu quả của hai mô hình SVM và ANN trong bài toán phát hiện lỗi UAV qua log bay. Quy trình bao gồm thiết lập môi trường, tiền xử lý dữ liệu, huấn luyện và tối ưu hóa các mô hình dựa trên các siêu tham số đã đề xuất. Hiệu năng mô hình sẽ được đo lường khách quan thông qua một hệ thống chỉ số đánh giá toàn diện như accuracy, precision, recall và F1-score.*

## 3.1 Thiết lập môi trường thử nghiệm trên Google Colab

Mô hình được đưa vào huấn luyện và đánh giá trong môi trường Google Colab. Nền tảng điện toán đám mây này cung cấp GPU và TPU free, giúp rút ngắn đáng kể thời gian huấn luyện các mô hình học sâu.

Cấu hình phần cứng và phần mềm:

- GPU: NVIDIA Tesla T4 (16GB VRAM).
- RAM hệ thống: 12 GB.
- Phiên bản Python: 3.10.
- Thư viện chính sử dụng:
    + scikit-learn: triển khai mô hình SVM, đánh giá hiệu suất.
    + TensorFlow và Keras: xây dựng, train và đánh giá chất lượng mô hình ANN.
    + NumPy, Pandas: xử lý sơ bộ dữ liệu log.
    + Matplotlib, Seaborn: minh họa kết quả huấn luyện dưới dạng đồ thị,

Quy trình cài đặt thư viện:

```
!pip install scikit-learn tensorflow keras pandas matplotlib seaborn
```

**Hình 3-1 Lệnh cài thư viện trên Google Colab**

Dữ liệu log UAV được lưu trữ trên Google Drive và liên kết trực tiếp với môi trường Colab.

## 3.2 Thu thập dữ liệu:

Tập dữ liệu được sử dụng cho nghiên cứu có nguồn gốc từ cả dữ liệu công khai trên Internet và dữ liệu thực tế được trích xuất từ log bay.

- Nguồn dữ liệu mở:
+ Dữ liệu train được lấy từ PX4 Flight Review, một nền tảng công khai cung cấp hàng ngàn log bay UAV ghi lại thông tin cảm biến, trạng thái hệ thống, và sự kiện lỗi trong quá trình hoạt động thực tế.
+ Địa chỉ truy cập: https://logs.px4.io/browse
+ Dữ liệu được công bố theo giấy phép CC-BY PX4, cho phép dùng cho mục đích nghiên cứu.
+ Mỗi log có cấu trúc gồm: thời gian bay, tốc độ, độ cao, gia tốc, góc nghiêng, điện áp pin, và tín hiệu điều khiển.
- Dữ liệu thực tế
+ Dữ liệu thu thập log bay thực tế từ UAV hạng nhẹ (quadcopter)
+ Thời tiết: gió < 5 m/s, không mưa, nhiệt độ 25–35°C.
+ Thiết bị bay: UAV 4 cánh quạt, điều khiển bằng Pixhawk 4, chạy firmware PX4 1.13.

## 3.3 Triển khai quy trình thực nghiệm

### 3.3.1 Tiền xử lý log bay

Trong bài toán này, dữ liệu log bay thu thập được thường chứa nhiễu, sai lệch đo và sự không đồng nhất giữa các chuyến bay. Vì vậy, trước khi đưa vào các mô hình học máy, dữ liệu cần được tiền xử lý theo một quy trình có hệ thống nhằm bảo đảm tính tin cậy, nhất quán và phù hợp với yêu cầu của các thuật toán sử dụng trong đề án.

Quy trình tiền xử lý log bay được triển khai qua các bước sau:

Bước 1: Chuẩn bị và gán nhãn dữ liệu

Mục tiêu chính của giai đoạn này là xác định chính xác trạng thái hoạt động của UAV tại từng thời điểm ghi nhận trong tệp log bay, qua đó xây dựng tập nhãn đáng tin cậy để huấn luyện và đánh giá mô hình phân loại. Trong thực tế, dữ liệu log thu thập được từ các chuyến bay không đi kèm thông tin về lỗi hệ thống. Vì vậy, luận văn đã tiến hành một quy trình gán nhãn bán thủ công, dựa trên sự kết hợp giữa phân tích tự động trên dữ liệu log và xác minh thủ công từ nhiều nguồn thông tin vận hành.

Dữ liệu log thô từ bộ điều khiển bay được tích hợp và đồng bộ hóa với các nguồn dữ liệu bổ trợ khác như nhật ký cảnh báo hệ thống, báo cáo vận hành từ người điều khiển. Việc đồng bộ được thực hiện dựa trên trường thời gian (timestamp) chung, cho phép liên kết chính xác các sự kiện được ghi nhận từ các nguồn khác nhau. Từ các nguồn thông tin tích hợp, các khoảng thời gian nghi ngờ xảy ra sự cố, gọi là cửa sổ sự cố, được xác định ban đầu. Trong các cửa sổ này, UAV thường biểu hiện các hành vi bất thường như dao động góc nghiêng lớn, sai lệch đáng kể giữa giá trị đặt và giá trị thực của vòng điều khiển, hoặc mất tín hiệu cảm biến và điều khiển. Dựa trên đặc điểm biểu hiện trong log, các sự cố được phân loại theo nhóm nguyên nhân chính, bao gồm lỗi động cơ, lỗi cảm biến và lỗi kết nối. Lỗi động cơ thường thể hiện qua sự sai lệch kéo dài giữa tín hiệu điều khiển (PWM, RPM đặt) và phản hồi thực tế (dòng điện, nhiệt độ động cơ), hoặc qua sự mất cân bằng lực đẩy giữa các rotor. Lỗi cảm biến được nhận diện thông qua sự gián đoạn dữ liệu (data dropout), nhiễu lớn bất thường, hoặc mất tính nhất quán vật lý giữa các tín hiệu đo liên quan (ví dụ: vận tốc GPS khác biệt lớn so với vận tốc ước tính từ IMU). Lỗi kết nối thể hiện qua hiện tượng mất gói tin (packet loss) liên tục, độ trễ (latency) bất thường trong luồng dữ liệu điều khiển từ trạm điều khiển mặt đất (GCS), hoặc sự ngắt quãng trong liên lạc telemetry.

Sau khi xác định các cửa sổ và nguyên nhân sự cố, nhãn nhị phân được áp dụng cho toàn bộ tập dữ liệu để phục vụ mục tiêu phân loại lỗi của đề án. Cụ thể, tất cả các mẫu dữ liệu nằm trong các cửa sổ sự cố được gán nhãn 1 (bất thường -

anomaly), trong khi các mẫu thuộc khoảng thời gian hoạt động ổn định được gán nhãn 0 (bình thường - normal).

Bước 2: Làm sạch dữ liệu

Do đặc thù dữ liệu bay thực tế, tập log thường xuất hiện các giá trị bị thiếu hoặc nhiễu do gián đoạn cảm biến và truyền thông. Đối với các giá trị khuyết, luận văn sử dụng phương pháp Simple Imputer với giá trị trung vị cho từng đặc trưng. Cách tiếp cận này giúp hạn chế ảnh hưởng của các giá trị cực đoan, vốn phổ biến trong dữ liệu động học của UAV.

Các điểm dữ liệu bất thường được xác định thông qua phương pháp IQR (Interquartile Range). Trong bối cảnh bài toán này, không phải mọi giá trị bất thường đều tương ứng với lỗi hệ thống. Do đó, IQR được sử dụng nhằm loại bỏ các nhiễu đo hoặc giá trị phi vật lý rõ rệt, các giá trị cực đoan trong lớp bình thường được xem xét loại bỏ hoặc điều chỉnh, trong khi các giá trị bất thường trong lớp anomaly được bảo toàn như là đặc trưng của trạng thái lỗi. Cách xử lý này giúp cân bằng giữa việc làm sạch dữ liệu và bảo toàn các tín hiệu lỗi tiềm năng.

Bước 3: Trích chọn đặc trưng

Mặc dù log ghi nhận một số lượng lớn tham số, nhưng đề án không sử dụng toàn bộ tập đặc trưng mà tập trung lựa chọn 9 đặc trưng đầu vào tiêu biểu. Việc lựa chọn này dựa trên các cơ sở sau:

+ Các đặc trưng được chọn có mối liên hệ trực tiếp với trạng thái động học và điều khiển bay, phản ánh trạng thái vận động cốt lõi của UAV (vị trí, vận tốc, góc nghiêng, gia tốc), là những đại lượng nhạy cảm nhất khi lỗi điều khiển hoặc lỗi cảm biến xảy ra.
+ Nhiều tham số trong log có mối tương quan cao hoặc chỉ mang tính dẫn xuất. Việc giới hạn số đặc trưng giúp giảm dư thừa thông tin, và cải thiện khả năng tổng quát hóa của mô hình.

+ Các đặc trưng được chọn có ý nghĩa vật lý rõ ràng, giúp kết quả mô hình dễ phân tích và có thể áp dụng cho các hệ UAV tương tự.
+ Với kích thước tập dữ liệu ở mức trung bình( hơn 12000 mẫu), việc sử dụng số lượng đặc trưng vừa phải giúp cân bằng giữa khả năng biểu diễn của mô hình và nguy cơ quá khớp (overfiting).

Sau khi xác định tập đặc trưng, dữ liệu được tách thành tập đầu vào và tập nhãn cho cả hai tập huấn luyện và kiểm tra nhằm bảo đảm tính nhất quán trong toàn bộ quá trình thực nghiệm.

. Bước 4: Chuẩn hóa dữ liệu

Dùng kỹ thuật StandardScaler để thực hiện phép chuẩn hóa Z-score, dựa trên công thức chuyển đổi lấy giá trị gốc trừ giá trị trung bình của đặc trưng rồi chia cho lệch chuẩn tương ứng. Quá trình này thực hiện trên từng đặc trưng.

### 3.3.2 Huấn luyện mô hình

Hai thuật toán được sử dụng trong quá trình huấn luyện và đối chiếu hiệu quả là SVM đa lớp và mạng neuron nhân tạo (ANN)

Tập dữ liệu được phân tách theo tỷ lệ 75% cho giai đoạn huấn luyện và 25% cho quá trình kiểm tra. Các mẫu được xáo trộn ngẫu nhiên trước khi tách để tránh sai lệch thống kê.

Mô hình SVM:

- Sử dụng hàm nhân (kernel) RBF để xử lý quan hệ không tuyến tính giữa các đặc trưng.
- Hệ số phạt $C \in [0.1,1,10]$
- Gamma sử dụng Sacle để tự động tính toán theo đặc điểm dữ liệu
- Tập huấn luyện được đánh giá bằng cross-validation (k=3) để chọn nhóm tham số đạt mức hiệu quả cao nhất.

Mô hình ANN:

- Cấu trúc mạng gồm 3 lớp ẩn với kích thước giảm dần 64->32-> số lớp đầu ra (4 lớp), hàm kích hoạt ReLU.
- Lớp đầu ra, mô hình áp dụng Softmax cho bài toán phân loại nhiều lớp lỗi.
- Huấn luyện bằng thuật toán Adam optimizer, tốc độ học $lr$=0.001, số epoch = 50, batch size = 32, hàm mất mát sử dụng Categorical Crossentropy.
- Cơ chế Dropout được áp dụng để ngăn chặn overfitting.

### 3.3.3 Thử nghiệm

Hoạt động thử nghiệm được vận hành với tập data 12420 bản ghi với 4 nhãn tương ứng:

+ 0: Bản ghi không có lỗi
+ 1: Bản ghi có lỗi GPS
+ 2: Bản ghi có lỗi động cơ
+ 3: Bản ghi có lỗi mất kết nối

Nhãn huấn luyện và kiểm thử được phân bố như trong bảng 3.1..

**Bảng 3-1 Phân bố nhãn**

| Nhãn( Label) | Tập huấn luyện | Tập kiểm thử |
|---|---|---|
| | Số lượng | |
| 0 | 3974 | 994 |
| 1 | 2583 | 447 |
| 2 | 1788 | 646 |
| 3 | 1590 | 397 |

## 3.4 Đánh giá hiệu quả mô hình phát hiện lỗi UAV

Hiệu năng của các mô hình được xem xét bằng cách kiểm tra trên tập dữ liệu riêng biệt để đo lường khả năng nhận dạng và dự báo lỗi UAV. Các chỉ số đo lường hiệu quả bao gồm: Accuracy, Precision, Recall và F1-score. Những chỉ số này được tính toán dựa trên số liệu thực nghiệm gồm True Positive (TP), True Negative (TN), False Positive (FP) và False Negative (FN), nhằm đo lường độ chính xác tổng thể, khả năng xác định đúng các lỗi, cũng như sự cân bằng giữa Precision và Recall [1].

Các công thức được định nghĩa như sau:

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (3.1)$$

Precision

$$Precision = \frac{TP}{TP + FP} \qquad (3.2)$$

Recall

$$Recall = \frac{TP}{TP + FN} \qquad (3.3)$$

F1-score

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad (3.4)$$

Trong đó, TP, TN, FP và FN lần lượt biểu thị số lượng dự đoán đúng dương tính, đúng âm tính, dự đoán sai dương tính và dự đoán sai âm tính, phản ánh sự khác biệt giữa kết quả mô hình và dữ liệu thực tế [1].

Nhằm mục đích đánh giá toàn diện và khách quan năng lực của các mô hình dự đoán lỗi UAV, nghiên cứu triển khai kiểm định dựa trên bốn kịch bản chuyên sâu.

Mỗi kịch bản được thiết kế để nhấn mạnh vào một khía cạnh cụ thể trong hiệu suất và tính ứng dụng của mô hình, từ đó cung cấp cái nhìn đa chiều và có ý nghĩa thực tiễn.

- **Kịch bản 1: Đánh giá hiệu năng tổng quát**

Mục tiêu: Xác định hiệu năng cơ bản và so sánh trực tiếp giữa các mô hình trên một bộ dữ liệu kiểm tra đã được chuẩn hóa, đảm bảo tính khách quan và công bằng trong đánh giá ban đầu.

Phương pháp triển khai:

- Tập dữ liệu kiểm tra được tách ngẫu nhiên từ tập dữ liệu gốc (tỷ lệ 75:25), đảm bảo giữ nguyên phân phối tự nhiên của các lớp lỗi.
- Kiểm tra và thử nghiệm các mô hình trên cùng một bộ dữ liệu chuẩn đã được thiết lập.
- Những chỉ số đánh giá chính được sử dụng là Accuracy (tỷ lệ dự đoán đúng tổng thể), Macro-average F1-score (F1 trung bình trên toàn bộ các lớp) và Weighted-average F1-score (F1 trung bình có cân nhắc trọng số theo số lượng mẫu).

Kịch bản này mang lại một cái nhìn khái quát về độ chính xác và tính ổn định của mô hình trong điều kiện lý tưởng, là cơ sở ban đầu để lựa chọn mô hình tiềm năng.

- **Kịch bản 2: Đánh giá theo từng lỗi**

Mục tiêu: Phân tích chi tiết điểm mạnh và điểm hạn chế của từng mô hình trong việc nhận diện cụ thể từng loại lỗi UAV, từ đó xác định các lỗi thường bị bỏ sót hoặc báo động sai.

Phương pháp triển khai:

- Xác định các chỉ số chỉ số Precision, Recall và F1-score một cách riêng biệt cho từng lớp trong toàn bộ tập hợp lớp.

- Sử dụng Confusion Matrix để minh họa các lỗi phân loại một cách trực quan, phản ánh các lớp mà mô hình hay bị nhầm lẫn.

- **Kịch bản 3: Đánh giá dưới điều kiện dữ liệu mất cân bằng**

Mục tiêu: Kiểm chứng tính vững vàng của mô hình trong điều kiện thực tế, nơi dữ liệu về các lỗi nghiêm trọng thường rất hiếm so với trạng thái bình thường hoặc các lỗi thông thường.

Phương pháp triển khai:

- Giữ nguyên tính mất cân bằng tự nhiên của tập dữ liệu quá trình train và test.
- Áp dụng các chỉ số đánh giá chuyên biệt được xây dựng riêng cho đặc thù của tập dữ liệu mất cân bằng, bao gồm:
    + Macro-average F1-score: Nhấn mạnh vào hiệu suất trên tất cả các lớp, bất kể số lượng mẫu.
    + Weighted-average F1-score: Phản ánh hiệu suất tổng thể có tính đến tỷ lệ của từng lớp.
- So sánh sự khác biệt giữa Accuracy và những chỉ số F1. Một mô hình tốt phải có Macro F1 cao ngay cả khi Accuracy cao, chứng tỏ nó không đơn thuần "học thuộc" lớp đa số.

Kịch bản này đảm bảo mô hình không bị ảnh hưởng bởi sự sai khác số lượng mẫu giữa các lớp và vẫn có thể phát hiện chính xác các lỗi hiếm gặp nhưng nguy hiểm.

- **Kịch bản 4: Đánh giá độ bền với nhiễu**

Thông qua kịch bản này, nghiên cứu hướng đến việc xác định mức độ suy giảm tin cậy của mô hình dưới tác động của nhiễu tín hiệu cảm biến - yếu tố luôn hiện diện trong môi trường vận hành thực tế.

Phương pháp triển khai:

- Mô phỏng nhiễu bằng cách chủ động thêm nhiễu nhân tạo vào các tín hiệu đầu vào của tập kiểm tra. Các loại nhiễu phổ biến được sử dụng bao gồm: nhiễu Gaussian với các mức phương sai (variance) khác nhau và nhiễu xung để mô phỏng các sự cố đột xuất.
- So sánh hiệu suất của mô hình (dựa trên Độ chính xác và Recall tổng thể) giữa tập dữ liệu gốc và tập dữ liệu đã được bổ sung nhiễu nhân tạo. Phép thử này nhằm kiểm tra khả năng duy trì hiệu năng của mô hình trong điều kiện dữ liệu đầu vào không lý tưởng, từ đó phản ánh tính ổn định và độ tin cậy khi áp dụng vào thực tế - nơi tồn tại nhiều yếu tố gây nhiễu.

Các kết quả thu được từ bốn kịch bản kiểm tra sẽ cung cấp cái nhìn tổng thể, làm nổi bật những điểm nổi bật và hạn chế của từng mô hình. Trên cơ sở đó, nghiên cứu sẽ đưa ra căn cứ để lựa chọn giải pháp thích hợp và có hiệu quả tối ưu nhất cho bài toán dự báo lỗi hệ thống UAV trong thực tế vận hành.

❖ **Kết quả đánh giá mô hình SVM đa lớp**

**Kịch bản 1: Đánh giá hiệu năng tổng quát**

Trong kịch bản tổng quát, mô hình SVM được đánh giá bằng tập kiểm tra độc lập nhằm phản ánh khả năng mô phỏng điều kiện vận hành bình thường. Với độ chính xác tổng thể 92.17%, mô hình thể hiện khả năng xử lý hiệu quả bài toán phân loại đa lớp trên tập dữ liệu UAV vốn được biết đến với độ phức tạp và nhiễu cao.

Với thời gian huấn luyện chỉ 8,2 giây, mô hình chứng tỏ ưu thế về chi phí tính toán thấp, từ đó đáp ứng tốt yêu cầu về tốc độ cho các hệ thống phân tích cần xử lý gần thời gian thực (near real-time).

Các chỉ số trung bình (F1-score weighted 92,5%, Macro F1-score 91.14%) cho thấy hiệu năng tổng thể tốt. Tuy nhiên, kết quả tổng quát này chưa phản ánh hết sự khác biệt giữa các lớp lỗi, nhất là trong môi trường dữ liệu có sự phân bố không đồng đều.

**Kịch bản 2: Đánh giá theo từng lớp lỗi**

Phân tích theo từng lớp lỗi được triển khai với mục tiêu đánh giá khả năng nhận dạng chi tiết của mô hình đối với từng loại trạng thái hoặc lỗi. Bảng 3.2 trình bày Precision, Recall và F1-score cho bốn lớp.

Từ kết quả thu được, có thể nhận thấy chênh lệch rõ rệt giữa các lớp:

- Lớp 2 và lớp 3 đạt hiệu năng gần như hoàn hảo (Precision và Recall ≈ 1.0). Điều này cho thấy đặc trưng tín hiệu của hai lớp này có tính phân tách cao và mô hình dễ dàng nhận diện.
- Lớp 0 có Recall cao (0.9525), thể hiện khả năng nhận dạng tốt; tuy nhiên Precision chỉ đạt 0.8660. Điều này gợi ý rằng một phần mẫu từ lớp khác, đặc biệt lớp 1, bị dự đoán nhầm sang lớp 0.
- Lớp 1 là lớp có hiệu suất thấp nhất, với Recall chỉ đạt 65,15%, tức mô hình bỏ sót khoảng một phần ba mẫu thuộc lớp này. F1-score của lớp 1 (0.7390) thấp hơn rõ rệt so với các lớp nhãn còn lại, có thể thấy đây là lớp khó phân loại nhất. Quan sát ma trận nhầm lẫn (hình 3.2) cũng cho thấy lớp 1 bị nhầm lẫn chủ yếu với lớp 0.

Kịch bản này cho thấy mô hình SVM đa lớp có "điểm yếu" ở khả năng nhận dạng một lớp cụ thể, đặt ra nhu cầu cân nhắc tăng cường dữ liệu hoặc điều chỉnh đặc trưng đầu vào cho lớp 1.

**Kịch bản 3: Đánh giá dưới điều kiện dữ liệu mất cân bằng**

Bảng 3.2 cho thấy sự chênh lệch rõ rệt: lớp 0 chiếm 40% tổng mẫu (1,242 mẫu), trong khi lớp 1 chỉ chiếm 17% (528 mẫu). Lớp 2 và 3 lần lượt chiếm 19% và 24%. So với trung bình 29.25%, lớp 0 vượt trung bình 10.75%, còn lớp 1 thấp hơn trung bình 11.25%. Tình trạng mất cân bằng được phản ánh qua cấu trúc phân phối, dù ở mức vừa phải nhưng vẫn gây ra những ảnh hưởng không nhỏ, khiến mô hình khó nắm bắt được các đặc điểm, đặc trưng của những lớp có ít mẫu. Tỷ lệ lớn nhất/nhỏ nhất lên đến 2.35:1, kết hợp với chỉ số Gini = 0.152 cho thấy phân phối dữ liệu có độ lệch đáng kể.

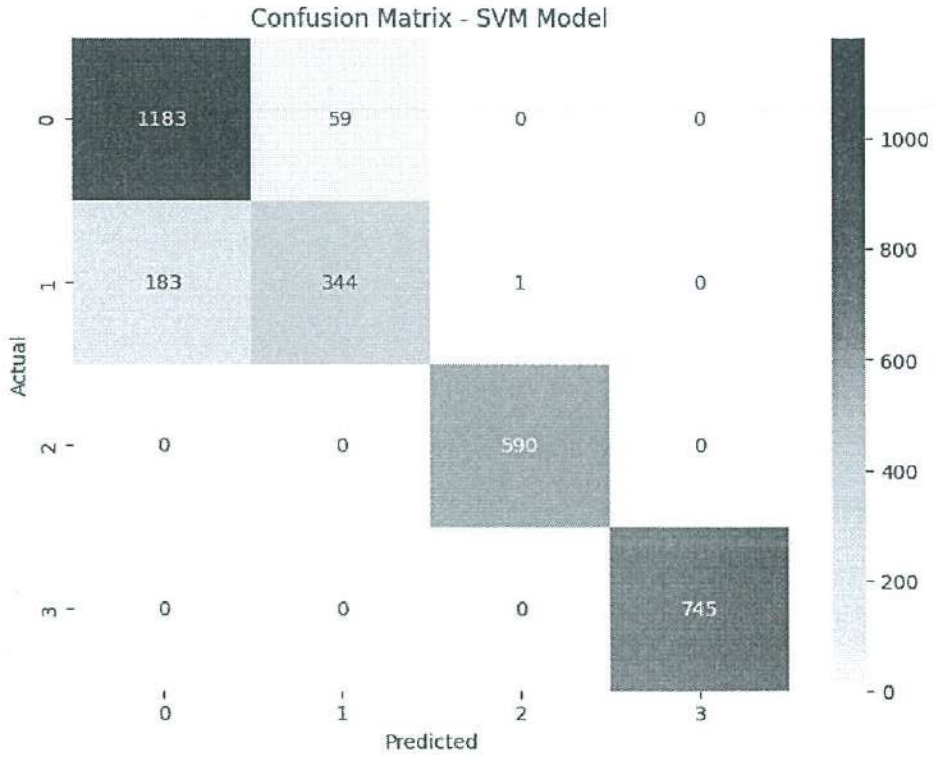
Đánh giá mối liên hệ giữa số lượng mẫu và F1-score cho thấy chỉ số tương quan âm mạnh (r = -0.647), cho thấy lớp càng ít mẫu thì hiệu suất càng thấp. Lớp 1 có số mẫu thấp nhất đạt F1-score thấp nhất, phản ánh khả năng nhận diện lỗi hiếm còn hạn chế. Tuy nhiên, đáng chú ý là lớp 0 (nhiều mẫu nhất) không đạt F1-score cao nhất, chứng tỏ hiệu suất không đơn thuần phụ thuộc vào số lượng mẫu mà còn liên quan đến mức độ tách biệt đặc trưng và mức độ chồng lấn tín hiệu giữa các lớp.

**Kịch bản 4: Đánh giá khi có thêm nhiễu**

Khi thêm nhiễu Gaussian ở nhiều mức độ vào tập kiểm tra, hiệu suất mô hình giảm không đồng đều giữa các lớp:

- Lớp 2 và lớp 3 hầu như không bị ảnh hưởng, do đặc trưng tín hiệu mạnh và ổn định, F1-score giảm không đáng kể.
- Lớp 0 ghi nhận sự suy giảm Precision do số lượng nhầm lẫn từ lớp 1 tăng lên khi tín hiệu trở nên nhiễu hơn.
- Lớp 1 chịu tác động rõ rệt nhất: Recall giảm mạnh hơn các lớp khác, cho thấy đặc trưng của lớp này dễ bị biến dạng dưới nhiễu và không còn giữ được tính phân biệt cao.

Từ những kết quả trên, có thể thấy rằng mô hình SVM có độ bền vững tốt với phần lớn các lớp, nhưng vẫn tồn tại độ nhạy cao đối với lớp 1.




**Hình 3-2 Ma trận nhầm lẫn model SVM**

**Bảng 3-2 Kết quả phân tích từng lớp mô hình SVM**

| Nhãn | Prercision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.8660 | 0.9525 | 0.9072 | 1242 |
| 1 | 0.8536 | 0.6515 | 0.7390 | 528 |
| 2 | 0.9983 | 1.00 | 0.9992 | 590 |
| 3 | 1.00 | 1.00 | 1.00 | 745 |

❖ **Kết quả mô hình ANN**

**Kịch bản 1: Đánh giá hiệu năng tổng quát**

Kết quả thực tế kiểm nghiệm cho thấy mô hình ANN đạt hiệu năng tổng thể cao, phản ánh khả năng đưa ra các đặc trưng phi tuyến trong dữ liệu log UAV. Cụ thể, độ chính xác (Accuracy) đạt 92.5%, cao hơn 0.33% so với mô hình SVM, cho

thấy ANN có thể xử lý bài toán phân loại nhiều lớp một cách linh hoạt và hiệu quả hơn. Macro F1-score đạt 91.47%, cũng nhỉnh hơn SVM (+0.33%), minh chứng rằng cải thiện của ANN không chỉ tập trung ở lớp đa số mà có tác động đồng đều lên toàn bộ hệ thống phân lớp. Bên cạnh đó, Weighted F1-score đạt 92.65%, phản ánh mô hình giữ vững được sự ổn định ngay cả trong điều kiện dữ liệu mất cân bằng.

Tổng thời gian train của mô hình là 41.75 giây, cao hơn so với SVM (tăng 33.55 giây). Tuy thời gian tính toán dài hơn, ANN đạt được mức Loss cuối cùng là 18.73%, cho thấy mô hình đã hội tụ và học được không gian đặc trưng phức tạp của dữ liệu.

Từ góc độ chi phí – hiệu quả (cost–performance), ANN thể hiện sự đánh đổi rõ rệt giữa thời gian train và mức cải thiện hiệu năng. Tỷ lệ "hiệu suất trên đơn vị thời gian" của ANN tính theo Accuracy đạt 0.221%/giây, thấp hơn nhiều so với 1.124%/giây của SVM. Điều này cho thấy ANN chậm hơn khoảng 5.1 lần, nhưng chỉ mang lại mức cải thiện khiêm tốn là 0.33% Accuracy. Kết quả này phản ánh quy luật "hiệu suất biên giảm dần" (diminishing returns) khi tăng mức độ phức tạp của mô hình: mặc dù ANN mạnh hơn trong biểu diễn phi tuyến, lợi ích thu được không tỷ lệ thuận với chi phí tính toán tăng thêm.

Biểu đồ theo dõi quá trình huấn luyện trong 50 epochs (Hình 3.4) cho thấy mô hình ANN hội tụ ổn định. Đường cong Accuracy tăng dần một cách mượt mà qua các epoch, trong khi Loss giảm đều và không ghi nhận các biến động lớn, minh chứng cho một quá trình tối ưu hóa ổn định.

Sự chênh lệch về độ chính xác và hàm mất mát giữa tập train và tập test luôn được duy trì trong ngưỡng cho phép. Điều này chứng tỏ mô hình không mắc phải hiện tượng học quá khớp (overfitting) một cách rõ rệt. Tính ổn định này đặc biệt có ý nghĩa trong bối cảnh xử lý dữ liệu log UAV - vốn là loại dữ liệu có nhiều nhiễu nền và có sự giao thoa đáng kể về đặc trưng giữa các lớp khác nhau.

**Kịch bản 2: Đánh giá theo từng lớp lỗi**

Đánh giá chi tiết từng lớp được mô tả trong bảng 3.3. Lớp 2 và lớp 3 đạt Precision = Recall = F1-score = 1.00, cho thấy mô hình nhận diện hoàn hảo các lỗi này. Nguyên nhân là đặc trưng của hai lớp này có độ phân tách cao và ổn định trong log UAV.

Lớp 0 có F1-score = 0.9109, cho thấy hiệu năng tốt khi số lượng mẫu lớn và phân bố đặc trưng tương đối rõ ràng. Ngược lại, lớp 1 là lớp có hiệu suất thấp nhất với F1-score chỉ đạt 74,81% và Recall ≈ 65.53%. Điều này cho thấy ANN thường xuyên bỏ sót mẫu lớp 1 hoặc nhầm lẫn với lớp 0 – hai lớp có đặc trưng gần nhau theo quan sát từ confusion matrix. Việc nhầm lẫn này phản ánh hạn chế trong khả năng học ranh giới phân lớp giữa các tín hiệu có độ tương đồng cao.

**Kịch bản 3: Đánh giá dưới điều kiện dữ liệu mất cân bằng**

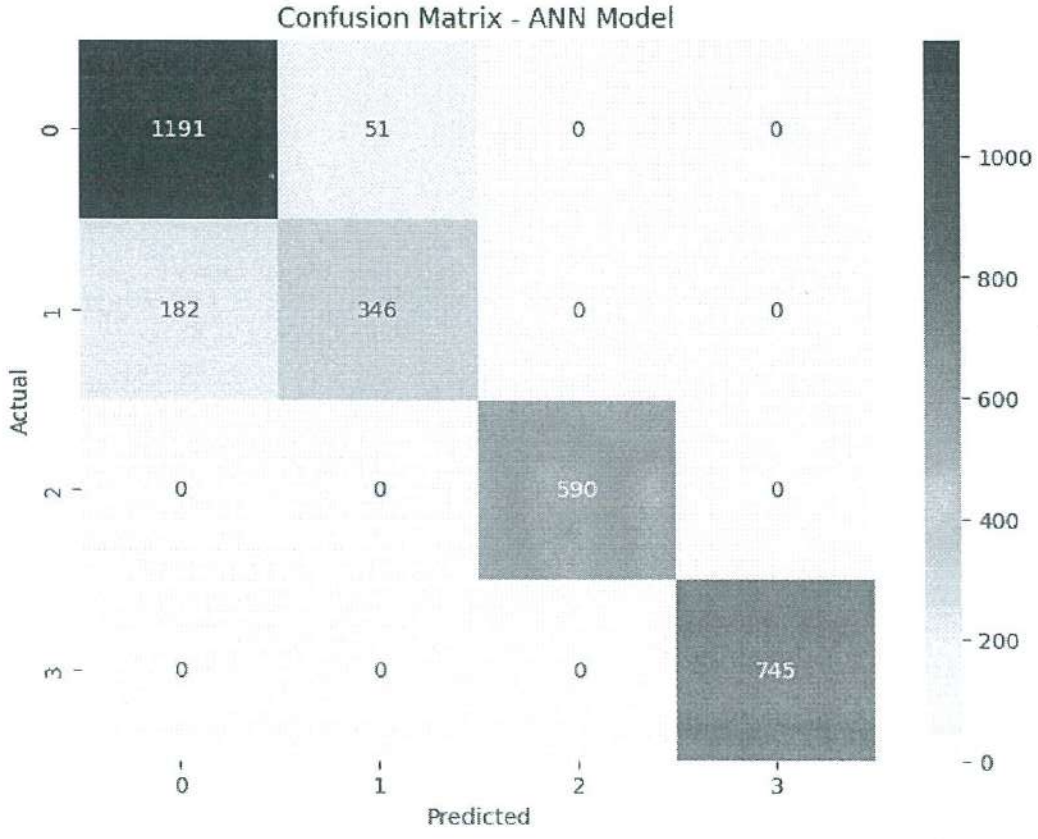
Kết quả thể hiện trên bảng 3.3 cho thấy mô hình ANN vẫn duy trì được hiệu năng cao ở các lớp có đặc trưng rõ ràng và có số lượng mẫu lớn hơn (lớp 2 và lớp 3), với Precision và Recall đều đạt 1.00. Dù vậy, mô hình thể hiện kết quả kém hơn đáng kể trên những nhóm có ít dữ liệu và các đặc trưng khó phân biệt, đặc biệt là lớp 1. Recall chỉ đạt 65.53%, thấp nhất trong bốn lớp. Với mức độ tin cậy thấp như vậy, có thể thấy mô hình khó có thể phát hiện chính xác các lỗi thuộc lớp 1 và dễ bỏ sót (false negative). Trong khi đó, lớp 0 dù là lớp đa số nhưng không đạt hiệu suất cao nhất, điều này phản ánh rằng ANN không bị lệch hoàn toàn về lớp nhiều mẫu, mà bị ảnh hưởng chủ yếu bởi sự tương đồng đặc trưng giữa các lớp 0 và 1.

Đánh giá mối liên hệ giữa số lượng mẫu và hiệu suất ghi nhận hệ số –0.647, biểu thị tương quan âm mạnh: lớp càng ít mẫu thì hiệu suất càng thấp. Điều này xác nhận rằng ANN nhạy cảm với phân phối dữ liệu không đồng đều. Tuy nhiên, tác động đến mô hình không mạnh, chứng tỏ ANN vẫn có hiệu quả trong việc tự động học hỏi các đặc trưng có tính phân biệt, dù bị hạn chế bởi thiếu dữ liệu của lớp thiểu số.

**Kịch bản 4: Đánh giá khi có thêm nhiễu**

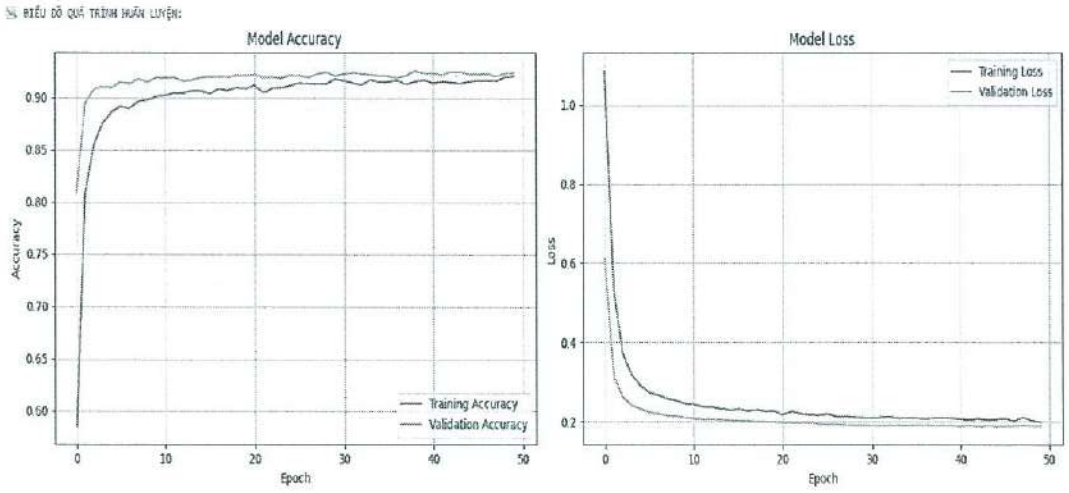
Mô hình ANN thể hiện khả năng ít bị ảnh hưởng của nhiễu khá tốt nhờ cấu trúc phi tuyến và khả năng học đặc trưng theo tầng. Với mức nhiễu thấp, các chỉ số Accuracy và Recall chỉ giảm nhẹ, cho thấy ANN đã học được các biểu diễn ổn định và không quá phụ thuộc vào điểm dữ liệu đơn lẻ. Tại mức nhiễu trung bình, độ chính xác bắt đầu giảm đáng kể ở các lớp gần nhau về đặc trưng (đặc biệt là lớp 1 và lớp 0), trong khi các lớp 2 và 3 chỉ bị ảnh hưởng nhỏ. Điều này củng cố quan sát rằng ANN mạnh hơn trong việc phân tách các lớp có dấu hiệu đặc trưng rõ rệt.

Khi nhiễu tăng lên mức cao, tổng hiệu năng suy giảm nhanh hơn so với mô hình SVM, phản ánh rằng ANN - dù linh hoạt nhưng lại dễ bị ảnh hưởng bởi biến dạng mạnh trong dữ liệu đầu vào.




**Hình 3-3 Ma trận nhầm lẫn mô hình ANN**

Quá trình huấn luyện với epoch 50 thu được kết quả như hình 3.4.




**Hình 3-4 Biểu đồ quá trình huấn luyện model Accuracy và Loss**

**Bảng 3-3 Kết quả từng lớp sau khi chạy mô hình ANN**

| Nhãn | Prercision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.8674 | 0.9589 | 0.9109 | 1242 |
| 1 | 0.8715 | 0.6553 | 0.7481 | 528 |
| 2 | 1.00 | 1.00 | 1.00 | 590 |
| 3 | 1.00 | 1.00 | 1.00 | 754 |

Kết quả thực nghiệm cho thấy cả hai mô hình SVM và ANN đều đạt hiệu suất phân loại tổng thể cao (Accuracy > 92%) trên tập dữ liệu kiểm tra, chứng minh tính khả thi của phương pháp học máy trong bài toán phát hiện lỗi UAV từ log bay. Tuy nhiên, phân tích sâu cho thấy có sự đánh đổi rõ ràng giữa hai phương pháp và tồn tại một điểm yếu cần khắc phục.

Về hiệu quả mô hình, ANN đạt độ chính xác 92.50%, cao hơn SVM (92.17%) một khoảng rất nhỏ (0.33%), nhưng thời gian huấn luyện lại dài hơn đáng kể (41.75 giây so với 8.12 giây). Sự chênh lệch này phản ánh bản chất của từng phương pháp: SVM với kernel RBF là lựa chọn tối ưu khi cần hệ thống nhẹ, có tốc độ suy luận nhanh và ổn định, phù hợp để triển khai trên các thiết bị biên (edge device) như flight

controller hoặc ground station có tài nguyên hạn chế. ANN mang lại lợi thế về khả năng mở rộng và linh hoạt kiến trúc, là nền tảng vững chắc để sau này phát triển lên các mô hình học sâu chuyên biệt hơn (như LSTM, 1D-CNN) cho việc phân tích chuỗi thời gian phức tạp, phù hợp với hệ thống phân tích tập trung trên server hoặc cloud nơi có thể chấp nhận chi phí tính toán cao hơn.

Hiệu suất thấp của cả hai mô hình trong việc phát hiện Lớp 1 - lỗi cảm biến, với chỉ số Recall chỉ đạt khoảng 65%. Điều này đồng nghĩa với việc khoảng 35% lỗi cảm biến thực tế có thể bị bỏ sót, tạo ra rủi ro an toàn nghiêm trọng. Nguyên nhân kỹ thuật được xác định chủ yếu do hai yếu tố: thứ nhất là sự mất cân bằng dữ liệu khi số lượng mẫu lỗi cảm biến trong tập huấn luyện là ít nhất, khiến mô hình không được học đầy đủ; thứ hai là tính chất tín hiệu của lỗi cảm biến (như drift, nhiễu) thường tinh vi, dễ bị che lấp bởi nhiễu nền hoặc nhầm lẫn với trạng thái vận hành biên giới của lớp "Bình thường".

Để khắc phục hạn chế trên, cần áp dụng các kỹ thuật xử lý mất cân bằng dữ liệu, chẳng hạn như tăng cường mẫu cho lớp lỗi cảm biến thông qua SMOTE hoặc điều chỉnh trọng số lớp trong hàm mất mát nhằm tăng mức phạt cho các trường hợp phân loại nhầm Lớp 1. Mở rộng tập đặc trưng theo hướng phản ánh trực tiếp hành vi bất thường của cảm biến, ví dụ như các thống kê trượt theo thời gian (giá trị trung bình, phương sai của gyro bias) hoặc các chỉ số chất lượng tín hiệu GPS, có thể giúp cải thiện khả năng phân biệt của mô hình. Việc giải quyết triệt để vấn đề này là bước then chốt để nâng cấp hệ thống từ một công cụ phân tích hậu kỳ lên thành một hệ thống cảnh báo sớm đáng tin cậy, có thể tích hợp vào quy trình vận hành an toàn của UAV.

## 3.5 Kết luận chương 3

Chương 3 của đề án đã trình bày hệ thống các bước thiết lập môi trường thí nghiệm, tổng hợp và xử lý dữ liệu, đồng thời triển khai hai mô hình học máy SVM và ANN để đánh giá khả năng nhận diện và phân loại sự cố hệ thống.

Toàn bộ quá trình thử nghiệm được thực hiện trong môi trường Google Colab, sử dụng các framework và thư viện chuyên dụng, bảo đảm tính khách quan và tính tin cậy. Dữ liệu sau khi thu thập đã trải qua giai đoạn tiền xử lý, được loại bỏ nhiễu và chuẩn hóa, tạo nền tảng dữ liệu sạch và đồng bộ cho việc huấn luyện mô hình.

Kết quả thử nghiệm cho thấy cả hai mô hình SVM và ANN đều thể hiện hiệu suất ấn tượng và ổn định. Cả hai mô hình đều đạt độ chính xác trên 92%, trong đó ANN đạt mức cao nhất 92.50%. Kết quả này khẳng định khả năng phù hợp môi trường thực tế và hiệu quả của phương pháp tiếp cận dựa trên học máy cho bài toán nghiên cứu.

# KẾT LUẬN

**Kết quả đạt được của đề án**

Đề án "Nghiên cứu ứng dụng học máy trong phát hiện và dự đoán lỗi hệ thống điều khiển UAV thông qua log bay" đạt được những kết quả quan trọng sau:

- *Hệ thống hoá cơ sở lý thuyết về phân tích log bay UAV và dự đoán lỗi*: Đề án đã nghiên cứu tổng quan các phương pháp giám sát UAV, quy trình tạo log theo các firmware phổ biến (PX4, ArduPilot), cấu trúc dữ liệu bay, các dạng lỗi thường gặp và cơ chế phát sinh. Các kỹ thuật học máy áp dụng trong bài toán phân loại (SVM, ANN) được phân tích theo từng khía cạnh: khả năng mô hình hoá phi tuyến, độ nhạy với nhiễu, khả năng khái quát hoá và hiệu suất tính toán.
- *Xây dựng bộ dữ liệu và chuẩn hóa quy trình tiền xử lý*: Đề án đã xử lý và trích xuất đặc trưng từ dữ liệu log bay đa chiều, bao gồm tín hiệu cảm biến, thông số điều khiển và trạng thái hệ thống. Dữ liệu được làm sạch, chuẩn hóa, phân chia train–test theo phương pháp khoa học để đảm bảo tính khách quan. Ngoài ra, đề án thực hiện phân tích phân phối lớp, đánh giá mức độ mất cân bằng và tác động của nó đến hiệu suất mô hình.
- *Triển khai trên 2 mô hình học máy*: mô hình SVM đa lớp và ANN với 50 epochs nhằm học biểu diễn sâu từ đặc trưng log bay.
- *Đánh giá hiệu năng bằng các tiêu chí định lượng*: Các chỉ số accuracy, precision, recall và F1-score đều được phân tích chi tiết. ANN đạt 92.5% accuracy, F1 trung bình trên 91%, trong khi SVM đạt 92.17%. Mô hình ANN cho thấy khả năng khái quát tốt ở các lớp có đặc trưng rõ ràng (lớp 2 và 3), trong khi SVM có tốc độ huấn luyện nhanh hơn (8.2 giây). Các ma trận nhầm lẫn cho thấy lỗi thường tập trung giữa các lớp có tính tương đồng tín hiệu cao (lớp 0 và lớp 1), phản ánh đúng bản chất nhiễu của dữ liệu thực tế.

- *Về tính ứng dụng*: Thay vì việc phân tích log thủ công đòi hỏi chuyên gia có kinh nghiệm, mô hình đề xuất đã chứng minh được khả năng hỗ trợ đắc lực cho người vận hành trong việc chẩn đoán nhanh tình trạng sức khỏe của thiết bị sau mỗi chuyến bay, giúp tối ưu hóa quy trình bảo trì và giảm thiểu rủi ro tai nạn.

**Hạn chế của đề án**

Mặc dù đạt được những kết quả khả quan, đề án vẫn còn một số hạn chế nhất định cần tiếp tục hoàn thiện:

- Tập dữ liệu huấn luyện mới chỉ tập trung vào một số kịch bản lỗi phổ biến, chưa bao quát được toàn bộ các tình huống lỗi hiếm gặp hoặc lỗi kết hợp phức tạp trong thực tế.
- Mô hình hiện tại mới dừng lại ở việc phân tích ngoại tuyến (offline) sau khi UAV đã hạ cánh, chưa được tích hợp trực tiếp để cảnh báo thời gian thực (real-time) trên trạm mặt đất (GCS).

**Hướng phát triển tiếp theo**

Do thời gian thực hiện đề án không nhiều, tác giả chưa có điều kiện nghiên cứu sâu rộng hơn về nhiều phương pháp mới. Trong tương lai, đề tài sẽ tiếp tục tập trung nghiên cứu các hướng phát triển sau:

- Cải tiến mô hình áp dụng các kiến trúc cho dữ liệu tuần tự như LSTM, GRU hoặc Transformer Time-Series nhằm mô tả tốt hơn động học UAV và sự lan truyền lỗi theo thời gian.
- Triển khai hệ thống suy luận thời gian thực. Tích hợp mô hình vào trạm điều khiển mặt đất (GCS) cho phép phát hiện lỗi trực tiếp trong quá trình bay.
- Khắc phục mất cân bằng dữ liệu bằng cách áp dụng SMOTE, ADASYN, cost-sensitive learning hoặc focal loss để nâng cao hiệu suất ở các lớp lỗi hiếm gặp.

# DANH MỤC CÁC TÀI LIỆU THAM KHẢO


[1]. Abe, S. (2005). *Support vector machines for pattern recognition*. Springer.

[2]. Al-Haddad, L. A., and Jaber, A. A. (2023). An intelligent fault diagnosis approach for multirotor UAVs based on deep neural network of multi-resolution transform features. *Drones, 7*(2), 82. https://doi.org/10.3390/drones7020082

[3]. Allwein, E. L., Schapire, R. E., and Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *The Journal of Machine Learning Research, 1*, 113–141.

[4]. Bagchi, T., Samant, R., and Joshi, M. (2013). *SVM classifiers built using imperfect training data*. International Conference on Mathematical Techniques in Engineering Applications (ICMTEA).

[5]. Burges, C. J. C. (2000). *A tutorial on support vector machines for pattern recognition*. Kluwer Academic Publishers.

[6]. Chen, X., and Wang, Y. (2022). Hybrid CNN-attention network for multi-modal UAV health monitoring using flight logs and thermal images. *Robotics and Autonomous Systems, 155*, 104118.

[7]. Cortes, C., and Vapnik, V. N. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297. https://doi.org/10.1007/BF00994018

[8]. Daumé, H., III. (n.d.). *A course in machine learning*. http://ciml.info/

[9]. Debele, Y., Shi, H., Wondosen, A., Ku, T., and Kang, B. S. (2023). Deep learning-based robust actuator fault detection and isolation scheme for highly redundant multirotor UAVs. *Drones, 7*(7), 437. https://doi.org/10.3390/drones7070437

[10]. Finn, A., Jia, M., Li, Y., and Yuan, J. (2024). Detecting stealthy GPS spoofing attack against UAVs using onboard sensors. *2024 IEEE INFOCOM Workshops*, 1–6.

[11]. Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press.

[12]. Gu Quanquan and Han Jiawei (2013). Clustered support vector machines. *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 307–315.

[13]. Hsu Chih-Wei and Lin Chih-Jen (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks, 13*(2), 415–425.

[14]. Iannace, G., Ciaburro, G., and Trematerra, A. (2019). Fault diagnosis for UAV blades using artificial neural network. *Robotics, 8*(3), 59. https://doi.org/10.3390/robotics8030059

[15]. Kobayashi, Y., and Takahashi, M. (2011). Design of intelligent fault-tolerant flight control system for unmanned aerial vehicles. In *Advances in Flight Control Systems*. IntechOpen. https://doi.org/10.5772/14209

[16]. Lei, H., and Govindaraju, V. (2005). Half-against-half multi-class support vector machines. In *Multiple Classifier Systems* (pp. 156–164). Springer

[17]. Liu, H., Zhang, R., Luan, F., Yao, X., Liu, M., Hu, Z., and Fan, B. T. (2003). Diagnosing breast cancer based on support vector machines. *Journal of Chemical Information and Computer Sciences, 43*(3), 900–907.

[18]. Lo, L. Y., Yiu, C. H., Tang, Y., Yang, A. S., Li, B., and Wen, C. Y. (2021). Dynamic object tracking on autonomous UAV system for surveillance applications. *Sensors, 21*(23), 7888.

[19]. McCoy, J., Rawal, A., and Rawat, D. B. (2023). Ensemble learning for UAV detection: Developing a multi-class multimodal dataset. *2023 IEEE 24th International Conference on Information Reuse and Integration for Data Science (IRI)*, 101–106.

[20]. Meier, L., Honegger, D., and Pollefeys, M. (2015). PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 6235–6240.

[21]. Milgram Jonathan, Cheriet Mohamed and Sabourin Réjean (2006). "One against one" or "one against all": Which one is better for handwriting recognition with SVMs? *Tenth International Workshop on Frontiers in Handwriting Recognition*.

[22]. Ng Andrew and Ma Tengyu (2023). *Lecture notes for machine learning*. Stanford University. http://cs229.stanford.edu/materials.html

[23]. Nittala S. K. R., Elkin C. P., Kiker J. M., Meyer R., Curro J., Reiter A. K., Xu K. S. and Devabhaktuni V. K. (2018). Pilot skill level and workload prediction for sliding-scale autonomy. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 1025–1030.

[24]. Pithode, K., and Patheja, P. S. (2022). A study on log anomaly detection using deep learning techniques. *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 290–298.

[25]. Rifkin, R., and Klautau, A. (2004). In defence of one-vs-all classification. *Journal of Machine Learning Research, 5*, 101–141.

[26]. Singh, P., and Sharma, A. (2023). Artificial neural networks: An overview. *International Journal of Advanced Computer Science and Applications, 14*(8), 1–15.

[27]. Vanek B., Bauer P., Gőzse I., Lukátsi M., Réti I. and Bokor J. (2014). *Safety critical platform for mini UAS insertion into the common airspace*. AIAA Guidance, Navigation, and Control Conference.

[28]. Wang, B., Wang, Z., Liu, L., Liu, D., and Peng, X. (2019). Data-driven anomaly detection for UAV sensor data based on deep learning prediction model. *2019 Prognostics and System Health Management Conference (PHM-Paris)*, 286–290.

[29]. Wu, W., and Zhou, H. (2017). Data-driven diagnosis of cervical cancer with support vector machine-based approaches. *IEEE Access, 5*, 25189–25195.

[30]. Yan, H. S., and Xu, D. (2007). An approach to estimating product design time based on fuzzy support vector machine. *IEEE Transactions on Neural Networks, 18*(3), 721–731.

[31]. Yousefi P., Fekriazgomi H., Demir M. A., Prevost J. J. and Jamshidi M. M. (2018). Data-driven fault detection of unmanned aerial vehicles using supervised learning over cloud networks. *2018 World Automation Congress (WAC)*, 1–6.

[32]. Zhang, H. (2021). Research on UAV route optimization method based on double target of confidence and ambiguity. *Frontiers in Neurorobotics, 15*, 673465.

[33]. Zhao, B., Liu, Y., and Xia, S. (2000). Support vector machine and its application in handwritten numeral recognition. *Proceedings 15th International Conference on Pattern Recognition (ICPR)*, 720–723.

[34]. Zhou Qing-Li, Zhang Youmin, Rabbath Camille Alain, Theilliol Didier and Unicersite Nancy (2010). *Reconfigurable control allocation technology using weighted least squares for nonlinear system in unmanned aerial vehicle*. 48th AIAA Aerospace Sciences Meeting.

Tài liệu internet

[35]. https://blog.dronedesk.io/uav-flight-log/. Truy cập ngày 17/09/2025

[36]. https://docs.px4.io/main/en/dev_log/ulog_file_format.html. Truy cập ngày 23/09/2025.

[37]. https://ardupilot.org/copter/docs/common-dataflash-logs.html. Truy cập ngày 23/09/2025

# 16% Tính tương đồng nói chung

Tổng cộng của tất cả các kết quả trùng khớp, bao gồm cả các nguồn trùng lặp, cho mỗi c...

## Đã lọc khỏi Báo cáo

- Mục lục tham khảo
- Văn bản được trích dẫn
- Văn bản được trích dẫn
- Kết quả trùng khớp nhỏ (ít hơn 8 từ)

## Nguồn hàng đầu

11% Nguồn Internet
6% Ấn bản
8% Bài tập được nộp (bài của học sinh)

Học viên

Vũ Thị Hợp

Người hướng dẫn

TS. Vũ Văn Thoá

BỘ KHOA HỌC VÀ CÔNG NGHỆ
**HỌC VIỆN CÔNG NGHỆ**
**BƯU CHÍNH VIỄN THÔNG**

**CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**
**Độc lập - Tự do - Hạnh phúc**

# BÁO CÁO GIẢI TRÌNH
# SỬA CHỮA, HOÀN THIỆN ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Họ và tên học viên: Vũ Thị Hợp

Chuyên ngành: Hệ thống thông tin

Khóa: 2024-2026

Tên đề tài: Nghiên cứu ứng dụng học máy trong phát hiện và dự đoán lỗi hệ thống điều khiển uav thông qua log bay

Người hướng dẫn khoa học: TS.Vũ Văn Thỏa

Ngày bảo vệ: 24/01/2026

Các nội dung học viên đã sửa chữa, bổ sung trong đề án tốt nghiệp theo ý kiến đóng góp của Hội đồng chấm đề án tốt nghiệp:

| TT | Ý kiến hội đồng | Sửa chữa của học viên |
|---|---|---|
| 1 | Làm rõ bài toán và ứng dụng thực tế | Đã bổ sung và làm rõ bài toán ở mục 1.3.1 trang 16.<br>Bổ sung tính ứng dụng thực tế ở phần Kết luận trang 57. |
| 2 | Sửa các lỗi chế bản và chính tả | Đã rà soát và sửa các lỗi chế bản và chính tả. Ví dụ: thay đổi hình ảnh 1-2 trang 11, hình 3-1 trang 37 cho rõ nét hơn, bổ sung nguồn cho hình 2-1. |
| 3 | Cấu trúc và bố cục lại đề án cho logic với bài toán | Bổ sung làm rõ hơn sự liên kết giữa các lý thuyết của 2 phương pháp học máy với bài toán ở mục 2.1.2 trang 26-27, mục 2.2.2 trang 31-32, mục 2.2.3 trang 32-33, mục 2.2.4 trang 33-34.<br>Chỉnh sửa nội dung kết luận chương 2 và mở đầu chương 3.<br>Bổ sung chi tiết quy trình tiền xử lý dữ liệu và giải thích cơ sở chọn |

| | | |
|---|---|---|
| | | đặc trưng đầu vào cho các mô hình ở mục 3.3.1 trang 39-41.<br>Bổ sung phân tích kết quả thực nghiệm, phân tích nguyên nhân recall của lớp 1 thấp ở mục 3.4 trang 53-54. |
| **4** | Kiểm tra tên đề án trong quyển và quyết định | Đã rà soát tên đề án trong quyển và quyết định tương đồng với nhau |

*Hà Nội, ngày..... tháng .... năm 2026*

**Ký xác nhận của**

| CHỦ TỊCH HỘI ĐỒNG CHẤM ĐỀ ÁN | THƯ KÝ HỘI ĐỒNG | NGƯỜI HƯỚNG DẪN KHOA HỌC | HỌC VIÊN |
|---|---|---|---|
| PGS.TSKH. Hoàng Đăng Hải | TS. Phạm Hoàng Duy | TS. Vũ Văn Thỏa | Vũ Thị Hợp |

CNTT9

BỘ KHOA HỌC VÀ CÔNG NGHỆ
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**
**Độc lập - Tự do - Hạnh phúc**

# BIÊN BẢN
# HỌP HỘI ĐỒNG CHẤM ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Căn cứ quyết định số Quyết định số 3299/QĐ-HV ngày 26 tháng 12 năm 2025 của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ. Hội đồng đã họp vào hồi ...9... giờ .30... phút, ngày 24 tháng 01 năm 2025 tại Học viện Công nghệ Bưu chính Viễn thông để chấm đề án tốt nghiệp thạc sĩ cho:

Học viên: **Vũ Thị Hợp**

Tên đề án tốt nghiệp: **Nghiên cứu ứng dụng học máy trong phát hiện và dự đoán lỗi hệ thống điều khiển uav thông qua log bay**

Ngành: **Hệ thống thông tin** Mã số: **8.48.01.04**

Các thành viên của Hội đồng chấm đề án tốt nghiệp có mặt: ..05../ 05

| TT | HỌ VÀ TÊN | TRÁCH NHIỆM TRONG HĐ | GHI CHÚ |
|---|---|---|---|
| 1 | **PGS.TSKH. Hoàng Đăng Hải** | Chủ tịch | |
| 2 | **TS. Phạm Hoàng Duy** | Thư ký | |
| 3 | **PGS.TS. Bùi Thu Lâm** | Phản biện 1 | |
| 4 | **TS. Trần Tiến Công** | Phản biện 2 | |
| 5 | **TS. Trịnh Bảo Ngọc** | Uỷ viên | |

**Các nội dung thực hiện:**

1. Chủ tịch Hội đồng điều khiển buổi họp. Công bố quyết định của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ.
2. Người hướng dẫn khoa học hoặc thư ký đọc lý lịch khoa học và các điều kiện bảo vệ đề án tốt nghiệp của học viên (Có bản lý lịch khoa học và kết quả các môn học cao học của học viên kèm theo).
3. Học viên trình bày tóm tắt đề án tốt nghiệp.
4. Phản biện 1 đọc nhận xét (có văn bản kèm theo)
5. Phản biện 2 đọc nhận xét (có văn bản kèm theo)
6. Các câu hỏi của thành viên Hội đồng:

Thầy Lâm: Giải thích AI AI học tập trung vào ? Làm rõ bài toán phân loại lỗi ? Phân tích kỹ hơn kết quả lỗi ? Cải xếp hạng log ?
Thầy Công: Làm rõ hơn về dữ liệu ? ứng dụng thực tế đề án ? Áp dụng khi xảy ra lỗi khẩn cấp ?
Thầy Ngọc: Cải tiến của đề án của làm khác

7. Trả lời của học viên:

- AUV: học được thêm về lối
- Chưa sử dụng toàn chức năng
- Phần cứng xử phù hợp với trong điều kiện, có thể áp dụng cho drone

8. Thư ký đọc nhận xét về quá trình thực hiện đề án tốt nghiệp của học viên *(có văn bản kèm theo)*.

9. Hội đồng họp riêng:

- Bầu Ban kiểm phiếu:
  1. Trưởng Ban kiểm phiếu: TS. Trần Tiến Lộc
  2. Uỷ viên Ban kiểm phiếu: TS. Trần Bảo Ngọc
  3. Uỷ viên Ban kiểm phiếu: TS. Phạm Hoàng Duy
- Hội đồng chấm đề án tốt nghiệp bằng bỏ phiếu kín.
- Ban kiểm phiếu làm việc:
- Trưởng Ban kiểm phiếu báo cáo kết quả kiểm phiếu *(có Biên bản họp Ban kiểm phiếu kèm theo)*
- **Điểm trung bình của đề án tốt nghiệp:** 8,50

**Kết luận:**

1. Các nội dung cần chỉnh sửa, hoàn thiện sau bảo vệ đề án tốt nghiệp:
   - Làm rõ hơn về nội dung thực tế
   - Sửa các lỗi chính tả + định lý
   - Cấu trúc lại các bài tổ án cho logic với bài toán
   - Kiểm tra lại đề án theo quyển và quyết định
2. Đề nghị Học viện công nhận (hoặc không) và cấp bằng (hoặc không) thạc sĩ cho học viên: Vũ Tài Hợp
3. Đề án tốt nghiệp có thể phát triển thành đề tài nghiên cứu cho NCS.: Không

Buổi làm việc kết thúc vào 12h40 cùng ngày.

| Chủ tịch | Thư ký |
|---|---|
| PGS.TSKH. Hoàng Đăng Hải | TS. Phạm Hoàng Duy |

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
**HỌC VIỆN CÔNG NGHỆ**
**BƯU CHÍNH VIỄN THÔNG**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
**Độc lập – Tự do – Hạnh phúc**
**********

# BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

*(Dùng cho người phản biện)*

Tên đề án: Nghiên cứu ứng dụng học máy trong phát hiện và dự đoán lỗi hệ thống điều khiển UAV thông qua log bay

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04

Tên học viên: VŨ THỊ HỢP

Họ và tên người nhận xét: Trần Tiến Công

Học hàm, học vị: Tiến sĩ, Giảng viên

Chuyên ngành: Trí tuệ nhân tạo

Cơ quan công tác: Khoa Trí tuệ nhân tạo, Học viện Công nghệ BC-VT

## NỘI DUNG NHẬN XÉT

**I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:**

- Đề tài có ý nghĩa thực tiễn cao trong bối cảnh việc ứng dụng UAV ngày càng phổ biến. Việc phân tích log bay để phát hiện lỗi giúp nâng cao độ an toàn và tin cậy của hệ thống, giảm thiểu rủi ro tai nạn do các sự cố kỹ thuật,.
- Hướng tiếp cận sử dụng học máy (Machine Learning) thay vì các phương pháp vật lý truyền thống là phù hợp với xu hướng khai thác dữ liệu lớn hiện nay.

**II/ Về nội dung, chất lượng của đề án, các kết quả đạt được (so với đề cương đã được duyệt):**

*** Nội dung đề án:**

Đề án tốt nghiệp của học viên gồm 3 chương chính:

- **Chương 1:** Giới thiệu bài toán và các vấn đề liên quan: Tổng quan về hệ thống điều khiển UAV, cấu trúc log bay và thách thức trong xử lý dữ liệu.
- **Chương 2:** Nghiên cứu một số phương pháp học máy: Trình bày lý thuyết về SVM phân loại đa lớp và Mạng nơ-ron nhân tạo (ANN).

- **Chương 3:** Thử nghiệm và đánh giá kết quả: Thiết lập môi trường trên Google Colab, tiền xử lý dữ liệu, huấn luyện mô hình và so sánh hiệu năng giữa SVM và ANN,.

*** Chất lượng đề án:** Đạt

*** Các kết quả đạt được:** Cơ bản đáp ứng các yêu cầu của một đề án thạc sĩ theo định hướng ứng dụng. Đã xây dựng được mô hình phân loại 4 lớp lỗi với độ chính xác trên 92%.

**III/ Những vấn đề cần giải thích thêm:**

Đề án có bố cục 3 chương là hợp lý, tuy nhiên nội dung chi tiết và cách trình bày còn một số vấn đề cần làm rõ hoặc chỉnh sửa:

- **Về bố cục và nội dung lý thuyết:**
    - **Chương 2:** Nội dung chương này còn nặng về lý thuyết giáo khoa. Các mục 2.1 và 2.2 trình bày các khái niệm cơ bản về SVM và ANN (nguyên lý, kernel, hàm kích hoạt...),,. Cần làm rõ hơn sự liên kết giữa các lý thuyết này với bài toán cụ thể là log UAV. Ví dụ: Tại sao chọn hàm kích hoạt ReLU hay Kernel RBF *cụ thể cho đặc thù dữ liệu log bay* (dữ liệu chuỗi thời gian, nhiễu cao) thay vì chỉ nêu định nghĩa chung?
    - Phần trích chọn đặc trưng (Feature Engineering) trong Chương 3 (mục 3.3.1) liệt kê 9 đặc trưng đầu vào, nhưng chưa giải thích cơ sở khoa học tại sao lại chọn đúng 9 đặc trưng này trong hàng trăm thông số của log bay PX4? Việc bỏ qua các thông số khác có làm mất mát thông tin quan trọng không?
- **Về phương pháp và dữ liệu (Cần giải thích rõ):**
    - **Dữ liệu thực tế và gán nhãn:** Mục 3.2 đề cập đến việc sử dụng "Dữ liệu thực tế từ UAV hạng nhẹ",. Tuy nhiên, đề án chưa mô tả rõ quy trình **gán nhãn (labeling)** cho dữ liệu thực tế này. Làm thế nào học viên xác định chính xác đâu là bản ghi "Lỗi động cơ" hay "Lỗi cảm biến" trong quá trình bay thực tế? Có thực hiện gây lỗi chủ động (fault injection) không hay chỉ dựa vào phán đoán? Đây là điểm mấu chốt để đánh giá độ tin cậy của tập dữ liệu huấn luyện.
    - **Xử lý ngoại lai (Outliers):** Tại bước 2 mục 3.3.1, học viên nêu sử dụng phương pháp IQR để xác định và xử lý điểm dữ liệu bất thường. Cần giải thích kỹ: Với bài toán phát hiện lỗi, bản thân các "lỗi" thường xuất hiện dưới dạng các giá trị bất thường (outliers). Việc dùng IQR để

lọc/xử lý có vô tình loại bỏ chính các tín hiệu lỗi mà mô hình cần phát hiện hay không?

- **Về kết quả thực nghiệm:**
  - **Hiệu quả giữa SVM và ANN:** Kết quả cho thấy độ chính xác của ANN (92.5%) chỉ nhỉnh hơn SVM (92.17%) rất nhỏ (0.33%), trong khi thời gian huấn luyện lại gấp 5 lần (41.75s so với 8.12s),. Học viên cần biện luận thuyết phục hơn về lý do lựa chọn ANN để triển khai thực tế khi chi phí tính toán cao hơn mà hiệu quả gia tăng không đáng kể.
  - **Độ nhạy thấp với Lớp 1 (Lỗi cảm biến):** Bảng 3-2 và 3-3 cho thấy Recall của Lớp 1 rất thấp (chỉ khoảng 65%) ở cả hai mô hình,. Với một hệ thống an toàn bay, việc bỏ sót 35% lỗi cảm biến là rất nguy hiểm. Đề án chưa đưa ra giải pháp cụ thể để cải thiện vấn đề mất cân bằng dữ liệu hoặc nâng cao độ nhạy cho lớp này.
  - Phần kết luận chương 2 và phần mở đầu chương 3 chưa có sự liên kết chặt chẽ về mặt logic lựa chọn tham số mô hình.
- **Lỗi trình bày:**
  - Một số hình ảnh (Hình 3-1, Hình 1-2) là ảnh chụp màn hình code hoặc log, chất lượng hiển thị chưa chuyên nghiệp, khó đọc,.
  - Tài liệu tham khảo cần rà soát lại định dạng trích dẫn cho thống nhất.

*Câu hỏi:*

1. Giải thích quy trình gán nhãn cho "dữ liệu thực tế" thu thập từ UAV 4 cánh quạt (mục 3.2). Làm sao xác thực được nhãn đó là đúng sự thật?
2. Việc sử dụng phương pháp IQR để xử lý dữ liệu ngoại lai (mục 3.3.1) có ảnh hưởng đến các mẫu dữ liệu chứa lỗi (vốn có tính chất bất thường) hay không?
3. Với kết quả Recall của lớp lỗi cảm biến (Lớp 1) chỉ đạt ~65%, hệ thống có đảm bảo an toàn để ứng dụng thực tế không? Hướng khắc phục là gì?

**IV/ Kết luận:**

Đề án đáp ứng các tiêu chuẩn của Đề án tốt nghiệp thạc sĩ kỹ thuật.

**Đồng ý cho phép học viên bảo vệ Đề án tốt nghiệp.**

*Hà Nội, ngày 20 tháng 01 năm 2026*

**Người phản biện**
(Ký ghi rõ họ tên)



**Trần Tiến Công**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

# BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

*(Dùng cho người phản biện)*

**Tên đề tài đề án tốt nghiệp:** Nghiên cứu ứng dụng học máy trong phát hiện và dự đoán lỗi hệ thống điều khiển UAV thông qua log bay

Chuyên ngành: Hệ thống thông tin
Mã chuyên ngành: 8.48.01.04

Họ và tên học viên: Vũ Thị Hợp

Họ và tên người nhận xét: PGS TS Bùi Thu Lâm
Học hàm, học vị: PGS TS
Chuyên ngành: KHMT
Cơ quan công tác: Học viện Kỹ thuật Mật mã
Số điện thoại: 0968699975 E-mail: lambt.actvn@gmail.com

## NỘI DUNG NHẬN XÉT

### I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài

Đề tài được lựa chọn có cơ sở khoa học và thực tiễn, bám sát xu hướng nghiên cứu hiện nay trong lĩnh vực đảm bảo độ tin cậy, an toàn và vận hành thông minh cho hệ thống UAV. Việc khai thác dữ liệu log bay để phát hiện và dự đoán lỗi là một hướng tiếp cận hợp lý, phù hợp với sự phát triển của các hệ thống UAV hiện đại có khả năng ghi nhận dữ liệu đa chiều với tần suất cao.

Về mặt khoa học, đề tài dựa trên nền tảng của học máy có giám sát, đặc biệt là các mô hình SVM và ANN, là những phương pháp đã được chứng minh hiệu quả trong bài toán phân loại, nhận diện bất thường và chẩn đoán lỗi hệ thống. Việc tổng hợp các nghiên cứu liên quan trong và ngoài nước cho thấy học viên đã nắm được bối cảnh nghiên cứu và vị trí của đề tài trong tổng thể lĩnh vực.

Về mặt thực tiễn, đề tài có ý nghĩa ứng dụng rõ rệt, nhất là trong bối cảnh hệ sinh thái UAV tại Việt Nam còn thiếu các công cụ phân tích log bay tự động, chủ yếu dựa vào kinh nghiệm thủ công. Kết quả nghiên cứu có tiềm năng hỗ trợ công tác giám sát, bảo trì dự đoán và nâng cao độ an toàn vận hành UAV, đặc biệt đối với UAV cỡ nhỏ sử dụng trong khảo sát, giám sát và dân sự.

Nhìn chung, đề tài có tính cấp thiết, phù hợp với định hướng đề án thạc sĩ theo hướng ứng dụng.

### II/ Nội dung của đề án tốt nghiệp, các kết quả đã đạt được

Nội dung đề án được trình bày có cấu trúc hợp lý, gồm ba chương chính:

Chương 1 trình bày tổng quan về hệ thống điều khiển UAV, log bay, các dạng lỗi thường gặp và các phương pháp phát hiện – khắc phục lỗi. Phần này được xây dựng công phu, có nhiều dẫn chứng thực tế, cho thấy học viên hiểu khá rõ đặc thù kỹ thuật của UAV và bài toán phân tích log bay.

Chương 2 giới thiệu hai phương pháp học máy được sử dụng trong nghiên cứu là SVM đa lớp và mạng nơ-ron nhân tạo (ANN). Các nguyên lý cơ bản, siêu tham số và ưu nhược điểm của từng phương pháp được trình bày tương đối rõ ràng, phù hợp với trình độ thạc sĩ.

Chương 3 triển khai thực nghiệm, bao gồm thu thập dữ liệu log bay, tiền xử lý, huấn luyện mô hình và đánh giá kết quả. Học viên đã xây dựng được quy trình thử nghiệm tương đối đầy đủ, sử dụng các chỉ số đánh giá như accuracy, confusion matrix và phân tích kết quả theo từng lớp lỗi.

Kết quả thực nghiệm cho thấy các mô hình học máy có khả năng phân loại và phát hiện lỗi UAV ở mức chấp nhận được, trong đó mô hình ANN thể hiện xu hướng cải thiện so với SVM ở một số lớp dữ liệu. Điều này chứng minh tính khả thi của hướng tiếp cận dựa trên học máy trong bài toán nghiên cứu.

**III/ Những vấn đề cần giải thích thêm (các hạn chế, thiếu sót)**

Mặc dù đề án đạt được các kết quả nhất định, vẫn tồn tại một số vấn đề học thuật và kỹ thuật cần được làm rõ thêm, cụ thể:

1. Bài toán dự đoán lỗi chưa được phân biệt rõ với bài toán phân loại lỗi
   Trong nhiều phần của luận văn, khái niệm *phát hiện*, *phân loại* và *dự đoán* lỗi được sử dụng xen kẽ. Tuy nhiên, về mặt học thuật:
   - Mô hình được xây dựng chủ yếu là phân loại trạng thái/lỗi dựa trên dữ liệu đã quan sát,
   - Chưa làm rõ yếu tố dự đoán theo thời gian (fault prediction / prognosis), tức là dự báo lỗi trước khi xảy ra trong tương lai.
     Học viên cần làm rõ phạm vi bài toán: đây là *fault detection & classification* hay *fault prediction*, và nếu là dự đoán thì yếu tố thời gian đã được khai thác ở mức nào.
2. Thiếu mô hình hóa bản chất chuỗi thời gian của log bay
   Dữ liệu log bay UAV về bản chất là chuỗi thời gian đa biến, có tương quan mạnh theo thời gian. Tuy nhiên, trong thực nghiệm:
   - Các mô hình SVM và ANN được sử dụng chủ yếu dưới dạng xử lý mẫu tĩnh (snapshot),
   - Chưa phân tích rõ cách xây dựng cửa sổ thời gian (sliding window), độ dài chuỗi, hoặc ảnh hưởng của lịch sử tín hiệu đến kết quả.
     Điều này làm hạn chế khả năng khai thác động học hệ thống và chưa tận dụng hết đặc trưng của dữ liệu log bay.
3. Tập dữ liệu còn hạn chế về quy mô và tính đa dạng
   Tập dữ liệu sử dụng trong thực nghiệm có quy mô tương đối nhỏ, phân bố nhãn chưa cân bằng, và chủ yếu tập trung vào một số dạng lỗi nhất định. Điều này dẫn đến:
   - Khả năng tổng quát hóa của mô hình chưa được kiểm chứng đầy đủ,

- Chưa có đánh giá về hiệu năng mô hình khi gặp các dạng lỗi hiếm hoặc chưa xuất hiện trong tập huấn luyện.

4. Thiếu so sánh với các phương pháp hiện đại hơn
   Luận văn mới dừng lại ở việc so sánh SVM và ANN cơ bản, trong khi:
   - Chưa có đối chứng với các phương pháp phổ biến cho chuỗi thời gian như LSTM, GRU, Autoencoder,
   - Chưa phân tích rõ lý do lựa chọn SVM và ANN thay vì các mô hình học sâu chuyên biệt cho log bay.
     Điều này làm cho đóng góp phương pháp luận của đề án chưa thật sự nổi bật.
5. Phần đánh giá kết quả còn thiên về mô tả, thiếu phân tích sâu
   Việc đánh giá chủ yếu dựa trên accuracy và confusion matrix, trong khi:
   - Chưa phân tích nguyên nhân gây nhầm lẫn giữa các lớp lỗi,
   - Chưa đánh giá độ ổn định mô hình khi thay đổi siêu tham số hoặc tập dữ liệu,
   - Chưa bàn luận về khả năng triển khai thời gian thực trên hệ thống UAV thực tế (độ trễ, tài nguyên tính toán).

Những điểm trên không làm mất giá trị của đề án, nhưng cần được học viên giải thích rõ khi bảo vệ để thể hiện mức độ hiểu biết và làm chủ vấn đề nghiên cứu.

**IV/ Kết luận**

Đề án tốt nghiệp được thực hiện nghiêm túc, có nội dung phù hợp với chuyên ngành Hệ thống thông tin và định hướng ứng dụng.

Học viên đã thể hiện được khả năng tổng hợp tài liệu, xây dựng mô hình và triển khai thực nghiệm ở mức đạt yêu cầu của luận văn thạc sĩ.

Đồng ý cho phép học viên bảo vệ đề án tốt nghiệp.

Ngày..14..tháng..1.....năm 2026

NGƯỜI NHẬN XÉT

Bùi Thu Lâm