BỘ GIÁO DỤC VÀ ĐÀO TẠO TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



KHOA CÔNG NGHỆ THÔNG TIN

NHẬP MÔN HỌC MÁY

**BÁO CÁO BÀI TẬP MÔN HỌC**

**NỘI DUNG: PRACTICE COMPETITION PHASE 03- ADVANCED ARCHITECTURES NEURAL NETWORK**

**Giảng viên hướng dẫn** :Nguyễn Thanh Tình Huỳnh Lâm Hải Đăng

**Lớp** :CQ2022/24

**Sinh viên thực hiện** :Đinh Viết Lợi-22120188 Nguyễn Trần Lợi- 22120190 Nguyễn Nhật Long-22120194

*Hồ Chí Minh, ngày 7 tháng 6 năm 2025*

# MỤC LỤC

Contents

[MỤC LỤC 1](#_Toc200272849)

[PHẦN 1: TỔNG QUAN ĐỒ ÁN 3](#_Toc200272850)

[I. Tổng quan đồ án 3](#_Toc200272851)

[1. Giới thiệu đồ án 3](#_Toc200272852)

[2. Yêu cầu đồ án 3](#_Toc200272853)

[II. Kết quả đạt được 4](#_Toc200272854)

[PHẦN 2: BÁO CÁO NHÓM 5](#_Toc200272855)

[I. Thông tin nhóm 5](#_Toc200272856)

[II. Quy trình thực hiện 5](#_Toc200272857)

[PHẦN 3: MODEL DEVELOPMENT 6](#_Toc200272858)

[I. Phân tích bộ dữ liệu 6](#_Toc200272859)

[II. Chi tiết quá trình xây dựng mô hình 6](#_Toc200272860)

[1. Tiền xử lý dữ liệu 6](#_Toc200272861)

[2. Transfomer 6](#_Toc200272862)

[3. Convolutional neural network (CNN) 6](#_Toc200272863)

[4. Recurrent neural network (RNN) 7](#_Toc200272864)

[III. Các phương pháp xử lý khác 7](#_Toc200272865)

[PHẦN 4: MODEL EVALUATION 8](#_Toc200272866)

[I. Các bộ chỉ số đánh giá 8](#_Toc200272867)

[II. Đánh giá và cải thiện bộ chỉ số 9](#_Toc200272868)

[1. Transformer 9](#_Toc200272869)

[2. Convolutional neural network (CNN) 9](#_Toc200272870)

[3. Recurrent neural network (RNN) 9](#_Toc200272871)

[PHẦN 5: ĐÁNH GIÁ PHƯƠNG PHÁP 9](#_Toc200272872)

[I. Phương pháp tiền xử lý dữ liệu 9](#_Toc200272873)

[II. Phương pháp lựa chọn và xây dựng mô hình 10](#_Toc200272874)

[1. Transfomer 10](#_Toc200272875)

[2. Convolutional neural network (CNN) 10](#_Toc200272876)

[3. Recurrent neural network (RNN) 10](#_Toc200272877)

[PHẦN 6: TÀI LIỆU THAM KHẢO 10](#_Toc200272878)

[I. Tài liệu nhóm 10](#_Toc200272879)

[II. Tài liệu tham khảo 10](#_Toc200272880)

# PHẦN 1: TỔNG QUAN ĐỒ ÁN

## Tổng quan đồ án

## Giới thiệu đồ án

* + - Đồ án Practice Competition là đồ án kết thúc môn học của bộ môn Nhập môn Học máy của lớp CQ2022/24 Trường Đại học Khoa học Tự nhiên học kỳ II năm học 2024-2025.
    - Đồ án tập trung vào thực hành quy trình xây dựng các mô hình học máy phục với 3 giai đoạn chính: Exploratory Data, Model Development, and Model Evaluation. Trọng tâm quy trình sẽ nằm ở hai giai đoạn sau.
    - Hình thức thực hiện đồ án được tổ chức dưới dạng một cuộc thi trên Kaggle với thể thức cá nhân hoặc theo nhóm. Các thông tin về cuộc thi được trình bày tại Kaggle competition.
    - Bài toán được đặt ra bởi cuộc thi sẽ được yêu cầu giải quyết trong vòng 3 tuần với 3 giai đoạn (phase) khác nhau. Tuần thứ ba cũng là tuần cuối cùng của cuộc thi được yêu cầu sử dụng các kiến trúc deep learning nâng cao hơn như CNN, RNN, Transformer.

## Yêu cầu đồ án

* + - Các sinh viên được mong đợi tuân thủ một quy trình phát triển cho dự án Học máy thực tế, được cấu trúc với ba bước chính: Phân tích Dữ liệu Khám phá (Exploratory Data Analysis), Phát triển Mô hình (Model Development) và Đánh giá Mô hình (Model Evaluation).
    - Bài tập được thực hiện dưới hình thức một cuộc thi Kaggle, sinh viên cần tham gia theo thể thức cá nhân hoặc theo đội nhóm.
    - Dự án được lên kế hoạch diễn ra trong tổng cộng 3 tuần, được cấu trúc thành 3 giai đoạn riêng biệt. Mỗi giai đoạn kéo dài một tuần và bao gồm toàn bộ chu trình của hai bước chính: Phát triển Mô hình, và Đánh giá và Phân tích Lỗi.
    - Mỗi sinh viên/ nhóm sinh viên được yêu cầu nộp quá trình thực hiện hàng tuần bao gồm mã nguồn Jupyter notebook và báo cáo cho từng Phase.
    - Mã nguồn Jupyter Notebook cần có: thông tin sinh viên/ nhóm sinh viên, giải thích chi tiết từng bước bằng hình ảnh minh hoạ, biểu đồ, công thức, chú thích (comment) từng bước xử lý và trực quan, notebook được format tốt.
    - Báo cáo bao gồm: thông tin sinh viên/ nhóm sinh viên, tự đánh giá kết quả, ưu và nhược điểm của các phương pháp xử lý, mô hình và frameworks khác nhau. Thông tin chi tiết mỗi giai đoạn cần đảm bảo:
      * **Model development**: Trình bày việc lựa chọn mô hình, thiết kế kiến trúc và thiết lập huấn luyện cùng với lý do đằng sau các lựa chọn của bạn.
      * **Đánh giá Mô hình:** Báo cáo về hiệu suất mô hình bằng cách sử dụng các chỉ số phù hợp, đồng thời bao gồm các quan sát về thời gian chạy, mức sử dụng tài nguyên tính toán. Cung cấp phân tích chi tiết về bất kỳ loại lỗi nào mà sinh viên và/hoặc nhóm đã xác định là điểm yếu của mô hình.

## Kết quả đạt được

* Quá trình xây dựng mô hình của nhóm được áp dụng mạnh mẽ và đa dạng các thư viện, framwork được hỗ trợ sẵn, tận dụng tối đa được sức mạnh từ các kiến trúc phức tạp để giải quyết bài toán.
* Nhóm áp dụng nhiều các phương pháp xử lý khác nhau nhằm chuẩn bị được bộ dữ liệu tốt nhất, phân tích các đặc điểm của dữ liệu hỗ trợ, gây khó khăn cho quá trình huấn luyện.
* Thực hiện huấn luyện nhiều lần với các bộ siêu tham số, kiến trúc mạng neural và các phương pháp khác nhau nhằm đánh giá quy luật, tìm ra bộ kết hợp tốt nhất để lựa chọn được phương thức huấn luyện mô hình phù hợp.
* Kết quả mô hình cho ra kết quả tốt, có thể được ứng dụng để giải quyết tốt vấn đề được đặt ra.
* Thông tin cụ thể về quá trình làm cũng như động lực được trình bày rõ ràng trong notebook và báo cáo.

# PHẦN 2: BÁO CÁO NHÓM

## Thông tin nhóm

|  |  |
| --- | --- |
| **Thành viên** | **Vai trò** |
| Đinh Viết Lợi | CNN |
| Nguyễn Trần Lợi | RNN |
| Nguyễn Nhật Long | Transforme |

## Quy trình thực hiện

Tận dụng số lượng submit có hạn hằng ngày trong thời gian ngắn, nhóm liên tục cập nhật source code hằng ngày để kiểm tra mức độ cải thiện, cũng như xác định độ hiệu quả của các phương pháp qua từng phiên bản:

* + Ngày 1:
  + Ngày 2:
  + Ngày 3:
  + Ngày 4:
  + Ngày 5:
  + Ngày 6: Tổng hợp phương án tối ưu cho việc huấn luyện mô hình, viết báo cáo tổng hợp.

# PHẦN 3: MODEL DEVELOPMENT

## Phân tích bộ dữ liệu

* + Bộ dữ liệu là tập các record được lưu dưới dạng bảng của loại dữ liệu supervised với nhãn is\_turkey xác định xem liệu một audio âm thanh có phải là tiếng gà tây hay không.
  + Các thuộc tính của dữ liệu bao gồm: các đặc trưng âm thanh của từng frame ~ 1s được embedding thành một features vector gồm 128 chiều thông thường đoạn audio n giây sẽ có ~n frame, các thuộc tính còn lại bao gồm thời gian bắt đầu và kết thúc của video.
  + Quan sát đặc tính của bộ dữ liệu cụ thể là các thuộc tính cho thấy chỉ có thuộc tính audio\_embedding sẽ là trọng tâm xác định nhãn của dữ liệu, các thuộc tính còn lại không đóng vai trò trong việc huấn luyện mô hình.

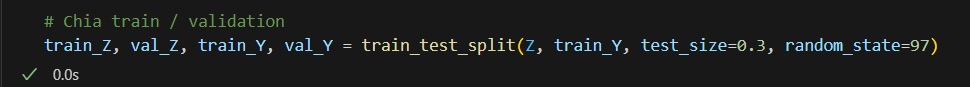
## Chi tiết quá trình xây dựng mô hình

1. **Tiền xử lý dữ liệu**
   * **Trích xuất đặc trưng cố định từ audio\_embedding:** Do mỗi audio\_embedding có số lượng frame khác nhau (tức là chiều thứ 2 của embedding không cố định), để chuẩn hóa đầu vào cho mô hình học máy, ta thực hiện việc **lấy trung bình theo trục thời gian** cho từng audio\_embedding. Cụ thể, với mỗi mẫu âm thanh, ta tính trung bình theo từng cột (tức là trung bình vector theo từng chiều đặc trưng), kết quả là mỗi mẫu sẽ được biểu diễn bởi một vector có chiều cố định.
   * **Chuẩn hóa dữ liệu**: Sau khi rút trích đặc trưng cố định, ta sử dụng **StandardScaler** để chuẩn hóa dữ liệu đầu vào. Phương pháp này chuẩn hóa mỗi đặc trưng sao cho chúng có **trung bình bằng 0 và độ lệch chuẩn bằng 1**, giúp mô hình học máy hội tụ nhanh hơn và tránh bị lệ thuộc vào độ lớn của các đặc trưng.

scaler= StandardScaler()

# Chuyển dữ liệu thành dạng chuẩn Z = scaler.fit\_transform(train\_X) test\_Z = scaler.transform(test\_X)

* + **Chia tập dữ liệu train và validation:** Dữ liệu sau chuẩn hóa được chia thành hai tập: **tập huấn luyện (training set)** và **tập xác thực (validation set)** để đánh giá khả năng tổng quát hóa của mô hình. Việc chia dữ liệu này đảm bảo mô hình không bị đánh giá trên chính dữ liệu nó đã học. Với dataset cụ thể này, tỉ lệ 70/30 (train/test) cho hiệu quả tốt và hợp lí nhất.



1. **Transfomer**
2. **Convolutional neural network (CNN)**
3. **Recurrent neural network (RNN)**

## Các phương pháp xử lý khác

* Data Regularization:
  + Một hiện tượng phổ biến xảy ra khi xây dựng mô hình là việc thường xuyên gặp phải overfitting với hầu hết các cấu hình siêu tham số hay kiến trúc mạng neural với nguyên nhân dễ thấy nhất là việc thiếu dữ liệu. Nhóm áp dụng một số phương pháp data regularization như Data Augmentation, Mixup, Soft Voting,... với mục đích tạo thêm dữ liệu huấn luyện, giúp mô hình tránh bị overfitting.
* r

# PHẦN 4: MODEL EVALUATION

## Các bộ chỉ số đánh giá

* + AUC:
    - Ý nghĩa: AUC đo lường khả năng của mô hình trong việc phân biệt giữa hai lớp. AUC càng gần 1 thì mô hình phân biệt càng tốt. AUC = 0.5 tương đương với mô hình ngẫu nhiên.
    - Dữ liệu đầu vào: Nhãn thật (y\_true) và xác suất dự đoán (y\_prob).
    - AUC xét toàn bộ các giá trị ngưỡng từ 0 đến 1, không cố định tại một giá trị nào.
    - Thư viện hỗ trợ:

from sklearn.metrics import roc\_auc\_score auc = roc\_auc\_score(y\_true, y\_prob)

* + Accuracy:
    - Ý nghĩa: Tỷ lệ dự đoán đúng trên tổng số mẫu. Phản ánh mức độ tổng quát mà mô hình dự đoán chính xác.
    - Dữ liệu đầu vào: Nhãn thật (y\_true) và xác suất dự đoán (y\_prob).
    - Cần chọn ngưỡng (ví dụ: 0.5) để chuyển xác suất (y\_prob) thành nhãn (y\_pred).
    - Thư viện hỗ trợ:

from sklearn.metrics import accuracy\_score accuracy = accuracy\_score(y\_true, y\_pred)

* + Precision:
    - Ý nghĩa: Tỷ lệ các mẫu được mô hình dự đoán là dương (positive) mà thực sự là dương. Phản ánh mức độ tin cậy khi mô hình dự đoán 1.
    - Dữ liệu đầu vào: Nhãn thật (y\_true) và xác suất dự đoán (y\_prob).
    - Cần chọn ngưỡng (ví dụ: 0.5) để chuyển xác suất (y\_prob) thành nhãn (y\_pred).
    - Thư viện hỗ trợ:

from sklearn.metrics import precision\_score precision = precision\_score(y\_true, y\_pred)

* + Recall:
    - Ý nghĩa: Tỷ lệ các mẫu thực sự dương được mô hình dự đoán đúng là dương. Quan trọng trong các bài toán nhạy cảm như y tế, an ninh.
    - Dữ liệu đầu vào: Nhãn thật (y\_true) và xác suất dự đoán (y\_prob).
    - Cần chọn ngưỡng (ví dụ: 0.5) để chuyển xác suất (y\_prob) thành nhãn (y\_pred).
    - Thư viện hỗ trợ:

from sklearn.metrics import recall\_score recall = recall\_score(y\_true, y\_pred)

* + F1-score:
    - Ý nghĩa: Là chỉ số cân bằng giữa Precision và Recall. F1-score cao nghĩa là mô hình vừa chính xác vừa phát hiện tốt lớp dương.
    - Dữ liệu đầu vào: Nhãn thật (y\_true) và xác suất dự đoán (y\_prob).
    - Cần chọn ngưỡng (ví dụ: 0.5) để chuyển xác suất (y\_prob) thành nhãn (y\_pred).
    - Thư viện hỗ trợ:

from sklearn.metrics import f1\_score f1 = f1\_score(y\_true, y\_pred)

## Đánh giá và cải thiện bộ chỉ số

* + **AUC Score =**
  + **Accuracy =**
  + **Precision = 0.97**
  + **Recall = 0.9604**
  + Mô hình đang thể hiện **rất tốt**, với **AUC gần tuyệt đối**, cùng **precision, recall, F1 đều vượt ngưỡng**
  + **THỜI GIAN SỬ DỤNG VÀ TÀI NGUYÊN THÊM VÀO ĐÂY LUÔN NHÉ**
  + Tuy nhiên khi thực hiện với bộ dữ liệu test, mô hình cho evaluation metrics càng cao quá mức thì càng đạt được độ chính xác thấp hơn điều này có thể được giải thích bởi hiện tượng overfitting. # Mẫu

1. **Transformer**
2. **Convolutional neural network (CNN)**
3. **Recurrent neural network (RNN)**

# PHẦN 5: ĐÁNH GIÁ PHƯƠNG PHÁP

## Phương pháp tiền xử lý dữ liệu

* + Trích xuất đặc trưng bằng mean:
    - **Ưu điểm:** Đơn giản, nhanh, dễ tính toán. Rút gọn được chiều dữ liệu → giảm kích thước đầu vào cho mô hình.
    - **Nhược điểm:** Thiếu cơ sở lý thuyết làm mất thông tin chi tiết theo từng đặc trưng. Có thể làm mất các mẫu âm thanh đặc trưng quan trọng.
  + Chuẩn hoá dữ liệu:
    - **Ưu điểm:** Giúp đưa các đặc trưng về cùng một thang đo, giúp quá trình lan truyền tiến (forward) và lan truyền ngược (backpropagation) trong mạng MLP trở nên ổn định hơn. Việc này hỗ trợ tăng tốc độ hội tụ, giúp mô hình học hiệu quả hơn và tránh hiện tượng gradient biến mất.
    - **Nhược điểm:** Nhạy cảm với outliers nếu tồn tại nhiều outliers sẽ dẫn đến hiệu suất kém cho mô hình.
  + Data regularization là một phương pháp phổ biến để tránh overfitting nhưng thực tế khi ứng dụng với bộ dữ liệu này lại không mang lại kết quả tốt, điều này có thể do phương pháp tạo thêm dữ liệu mới không phù hợp, pha trộn và sáng tạo nhưng không thể đánh nhãn phù hợp cho dữ liệu.

## Phương pháp lựa chọn và xây dựng mô hình

1. **Transfomer**
2. **Convolutional neural network (CNN)**
3. **Recurrent neural network (RNN)**

# PHẦN 6: TÀI LIỆU THAM KHẢO

## Tài liệu nhóm

Github: [Dzivilord/Machine\_Learning\_Project](https://github.com/Dzivilord/Machine_Learning_Project)

## Tài liệu tham khảo