

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH



BÁO CÁO ĐÒ ÁN CUỐI KỲ
MÔN: PHÁT TRIỂN ỨNG DỤNG TRÊN NỀN TẢNG
DỮ LIỆU LỚN

**PHÁT HIỆN NỘI DUNG ĐỘC HẠI TRÊN TIKTOK
BẰNG PHÂN TÍCH VIDEO – VĂN BẢN ĐỒNG THỜI**

Giảng viên hướng dẫn: Ts. Đỗ Trọng Hợp

Mã lớp: SE363.Q11

Sinh viên thực hiện:

Bùi Nhật Anh Khôi - 23520761

Đinh Lê Bình An - 23520004

Phạm Quốc Nam - 23520984

TPHCM, ngày 02 tháng 2 năm 2026

Mục lục

Chương 1: Giới thiệu.....	3
1.1. Bối cảnh và tầm quan trọng của vấn đề.....	3
1.2. Khoảng trống nghiên cứu hiện tại.....	3
1.3. Mục tiêu nghiên cứu và câu hỏi nghiên cứu.....	4
1.4. Đóng góp chính của đề tài.....	4
Chương 2: TỔNG QUAN TÀI LIỆU VÀ CÔNG TRÌNH LIÊN QUAN.....	5
2.1. Các nghiên cứu về phát hiện nội dung độc hại trên nền tảng video ngắn.....	5
2.2. Các phương pháp học sâu đa phương thức cho kiểm duyệt nội dung.....	5
2.3. Các hệ thống xử lý dữ liệu lớn cho kiểm duyệt nội dung.....	6
2.4. Khoảng trống nghiên cứu và hướng tiếp cận của đề tài.....	6
Chương 3: PHƯƠNG PHÁP ĐỀ XUẤT VÀ KIẾN TRÚC HỆ THỐNG.....	7
3.1. Tổng quan kiến trúc hệ thống.....	7
3.2. Quy trình thu thập dữ liệu.....	8
3.3. Kiến trúc mô hình AI đa phương thức.....	9
3.3.1. Nhánh xử lý văn bản.....	9
3.3.2. Nhánh xử lý video.....	10
3.3.3. Mô hình kết hợp đa phương thức (Multimodal Fusion).....	11
3.4. Quy trình xử lý dòng thời gian thực.....	12
3.5. Tích hợp vận hành máy học (MLOps).....	14
Chương 4: THỰC NGHIỆM VÀ KẾT QUẢ.....	15
4.1. Tập dữ liệu.....	15
4.1.1. EDA tần số phân bố nhãn.....	16
4.1.2. EDA phân tích độ dài văn bản.....	16
4.1.3. EDA thống kê các ngôn ngữ trong bộ dữ liệu.....	17
4.1.4. EDA phân tích dữ liệu video.....	17
4.1.5. EDA phân tích các hashtag trong dữ liệu video.....	17
4.2. Môi trường thực nghiệm.....	17
4.3. Các chỉ số đánh giá.....	18
4.4. Kết quả so sánh mô hình văn bản.....	18
4.5. Kết quả so sánh mô hình video.....	18
4.6. Kết quả mô hình kết hợp đa phương thức.....	19
4.7. Phân tích hiệu năng hệ thống.....	19
Chương 5: Thảo luận.....	20
5.1. Phân tích ý nghĩa kết quả.....	20
5.2. Các trường hợp nghiên cứu điển hình.....	21
5.3. Hạn chế và thách thức.....	21
Chương 6: Kết luận và hướng phát triển.....	22
6.1. Tổng kết.....	22
6.2. Hướng phát triển trong tương lai.....	22
Chương 7: TÀI LIỆU THAM KHẢO.....	23

Danh mục hình ảnh

Hình 1: Kiến trúc hệ thống phân tầng dựa trên mô hình xử lý dữ liệu lai.....	7
Hình 2: Sơ đồ luồng dữ liệu từ TikTok đến Kafka.....	8
Hình 3: Kiến trúc và cấu hình thực nghiệm các mô hình cho nhánh xử lý văn bản.....	10
Hình 4: Kiến trúc và cấu hình thực nghiệm các mô hình cho nhánh xử lý video.....	10
Hình 5: Kiến trúc và cấu hình thực nghiệm các mô hình kết hợp đa phương thức.....	11
Hình 6: Sơ đồ pipeline Spark Structured Streaming.....	12
Hình 7: Quy trình vận hành máy học (MLOps) tự động tích hợp Apache Airflow & MLflow..	14

Danh mục bảng

Bảng 1: Kết quả test các mô hình văn bản.....	18
Bảng 2: Kết quả test các mô hình video.....	19
Bảng 3: Kết quả test mô hình kết hợp đa phương thức.....	19

Chương 1: Giới thiệu

1.1. Bối cảnh và tầm quan trọng của vấn đề

Trong thập kỷ qua, bối cảnh truyền thông kỹ thuật số đã chứng kiến sự chuyển dịch mạnh mẽ từ văn bản thuần túy sang các định dạng đa phương tiện, trong đó video ngắn đã trở thành hình thức tiêu thụ nội dung chủ đạo. Sự trỗi dậy của TikTok với thuật toán gợi ý nội dung siêu cá nhân hóa đã thu hút hàng tỷ người dùng trên toàn cầu, đặc biệt là nhóm đối tượng thanh thiếu niên.

Tại Việt Nam, nền tảng này cũng đã nhanh chóng chiếm lĩnh thị trường và trở thành một phần không thể thiếu trong đời sống giải trí hàng ngày. Tuy nhiên, cơ chế lan truyền video dựa trên xu hướng (trend) và khả năng tạo nội dung dễ dàng đã biến TikTok thành một môi trường lý tưởng cho việc phát tán các nội dung độc hại. Các video chứa ngôn từ kích động thù địch, bạo lực, khiêu dâm, hay thông tin sai lệch có thể tiếp cận hàng triệu người xem chỉ trong thời gian ngắn trước khi bị gỡ bỏ. Hậu quả của việc tiếp xúc với các nội dung này là vô cùng nghiêm trọng, gây ra những tác động tiêu cực đến sức khỏe tâm thần, nhận thức xã hội và sự an toàn của cộng đồng người dùng.

Do đó, nhu cầu cấp thiết hiện nay là phải phát triển các hệ thống kiểm duyệt nội dung tự động có khả năng hoạt động theo thời gian thực với độ chính xác cao, nhằm hỗ trợ con người trong việc làm sạch môi trường mạng.

1.2. Khoảng trống nghiên cứu hiện tại

Mặc dù lĩnh vực tự động phát hiện nội dung độc hại đã đạt được những tiến bộ đáng kể, việc áp dụng vào thực tế cho các nền tảng video ngắn như TikTok vẫn đối mặt với nhiều thách thức chưa được giải quyết triệt để. Một hạn chế lớn của hầu hết các nghiên cứu hiện tại là sự tập trung đơn lẻ vào một dạng dữ liệu, hoặc chỉ phân tích văn bản trong phần bình luận và mô tả, hoặc chỉ phân tích hình ảnh trong video.

Cách tiếp cận đơn phương thức này thường thất bại trong việc nhận diện các nội dung độc hại tinh vi, nơi mà sự độc hại không nằm ở từng thành phần riêng lẻ mà nằm ở sự kết hợp ngữ cảnh giữa chúng. Ví dụ, một câu nói bình thường ghép với một hình ảnh nhạy cảm có thể tạo thành nội dung quấy rối, điều mà các mô hình đơn lẻ sẽ dễ dàng bỏ qua.

Bên cạnh đó, các phương pháp dựa trên từ khóa hoặc học máy truyền thống không còn đủ khả năng đối phó với sự biến đổi ngôn ngữ liên tục của cộng đồng mạng (teencode, tiếng lóng). Đặc biệt, đối với ngôn ngữ tiếng Việt, sự thiếu hụt các bộ dữ liệu chuẩn và các mô hình ngôn ngữ chuyên biệt được tinh chỉnh cho bài toán an toàn nội dung là một khoảng trống lớn.

Cuối cùng, khía cạnh triển khai hệ thống ở quy mô lớn (scalability) thường bị bỏ ngỏ trong các nghiên cứu học thuật thuần túy, dẫn đến thiếu vắng các tài liệu tham khảo về kiến trúc đường ống xử lý dữ liệu từ đầu đến cuối (end-to-end data pipeline).

1.3. Mục tiêu nghiên cứu và câu hỏi nghiên cứu

Đề tài này được thực hiện với mục tiêu xây dựng một hệ thống hoàn chỉnh có khả năng thu thập, xử lý và phát hiện nội dung độc hại tiếng Việt trên TikTok theo thời gian thực. Để đạt được mục tiêu này, nghiên cứu tập trung giải quyết ba câu hỏi cốt lõi.

1. **Câu hỏi thứ nhất liên quan đến kiến trúc hệ thống:** Làm thế nào để thiết kế một hạ tầng dữ liệu lớn có khả năng chịu tải cao, đảm bảo thu thập và xử lý luồng dữ liệu video liên tục với độ trễ thấp nhất có thể?

2. **Câu hỏi thứ hai tập trung vào khía cạnh mô hình trí tuệ nhân tạo:** Liệu việc kết hợp các đặc trưng từ văn bản và video thông qua các cơ chế học sâu đa phương thức có mang lại hiệu suất vượt trội so với các phương pháp đơn phương thức truyền thống hay không?
3. **Câu hỏi thứ ba hướng tới quy trình vận hành:** Làm sao để xây dựng một quy trình tự động hóa việc theo dõi, đánh giá và cập nhật mô hình (MLOps) nhằm đảm bảo hệ thống luôn thích ứng được với các dạng nội dung độc hại mới xuất hiện theo thời gian?

1.4. Đóng góp chính của đề tài

Nghiên cứu này đóng góp vào công đồng kỹ thuật và học thuật trên năm phương diện chính. Đầu tiên, đề tài đề xuất và hiện thực hóa một kiến trúc xử lý dữ liệu lai (Hybrid Architecture), kết hợp giữa xử lý theo lô và xử lý dòng thời gian thực, tích hợp các công nghệ mã nguồn mở hàng đầu. Thứ hai, nghiên cứu phát triển một phương pháp kết hợp đa phương thức tiên tiến sử dụng cơ chế chú ý chéo, chứng minh tính hiệu quả trong việc phát hiện nội dung độc hại tiếng Việt. Thứ ba, đề tài xây dựng một quy trình MLOps hoàn chỉnh, khép kín từ khâu gán nhãn, huấn luyện đến triển khai, đảm bảo tính bền vững của hệ thống. Thứ tư, nghiên cứu cung cấp một bộ công cụ thu thập dữ liệu chuyên sâu có khả năng vượt qua các cơ chế phòng vệ của nền tảng, đóng góp vào nguồn tài nguyên dữ liệu nghiên cứu. Cuối cùng, hệ thống đã được kiểm nghiệm thực tế và chứng minh khả năng hoạt động ổn định, mang lại giá trị ứng dụng cao trong việc hỗ trợ công tác quản lý và kiểm duyệt nội dung số.

Chương 2: TỔNG QUAN TÀI LIỆU VÀ CÔNG TRÌNH LIÊN QUAN

2.1. Các nghiên cứu về phát hiện nội dung độc hại trên nền tảng video ngắn

Trong những năm gần đây, sự bùng nổ của các nền tảng video ngắn như TikTok, YouTube Shorts và Instagram Reels đã thúc đẩy cộng đồng nghiên cứu tập trung vào bài toán kiểm duyệt nội dung tự động. Nghiên cứu của Cools, Wenniger và Maathuis được công bố tại hội nghị DPSH 2024 với tiêu đề "Modeling Offensive Content Detection for Tik Tok" là một trong những công trình tiên phong trực tiếp nhắm vào nền tảng TikTok. Nhóm tác giả đã thu thập dữ liệu gồm hơn 120.000 bình luận TikTok và xây dựng các mô hình học máy cũng như học sâu để phân loại nội dung xúc phạm. Kết quả thực nghiệm cho thấy mô hình đạt chỉ số F1 là 0,863 trên bài toán phân loại nhị phân cân bằng.

Tuy nhiên, nghiên cứu này tập trung hoàn toàn vào văn bản bình luận mà bỏ qua nội dung thị giác của video, đồng nghĩa với việc các hành vi độc hại thể hiện qua hình ảnh hoặc hành động trong video không được phát hiện. Hơn nữa, phương pháp được đề xuất không đề cập đến khả năng triển khai ở quy mô lớn hay xử lý theo thời gian thực, điều cần thiết cho các hệ thống kiểm duyệt thực tế.

2.2. Các phương pháp học sâu đa phương thức cho kiểm duyệt nội dung

Xu hướng nghiên cứu gần đây trong lĩnh vực kiểm duyệt nội dung đang chuyển dịch mạnh mẽ từ các phương pháp đơn phương thức sang các kiến trúc đa phương thức tích hợp. Hội thảo "Workshop on Multimodal Content Moderation" được tổ chức song song với hội nghị CVPR 2024 đã thu hút sự quan tâm đáng kể từ cộng đồng thị giác máy tính, phản ánh tầm quan trọng ngày càng tăng của hướng nghiên cứu này. Các công trình trình bày tại hội thảo tập trung vào ba chiến lược tích hợp chính. Chiến lược đầu tiên là early fusion, thực hiện nối các vector đặc trưng từ các phương thức khác nhau ngay từ giai đoạn đầu. Chiến lược thứ hai là late fusion, kết hợp kết quả dự đoán từ các mô hình riêng lẻ ở giai đoạn cuối. Chiến lược thứ ba và cũng là tiên tiến nhất là sử dụng cơ chế cross-modal attention, cho phép một phương thức "truy vấn" thông tin có liên quan từ phương thức khác, từ đó nắm bắt được các ngữ cảnh tinh vi mà sự độc hại được phân tán qua nhiều kênh thông tin.

Về khía cạnh mô hình ngôn ngữ, BERT và các biến thể của nó đã trở thành xương sống cho hầu hết các hệ thống phân tích văn bản. Mô hình XLM-RoBERTa với khả năng xử lý hơn 100 ngôn ngữ đã được chứng minh hiệu quả trong các nghiên cứu xử lý nội dung đa ngôn ngữ trên mạng xã hội. Đối với phân tích video, các kiến trúc Video Transformer như VideoMAE và TimeSformer đang dẫn đầu về hiệu suất. VideoMAE sử dụng phương pháp học tự giám sát thông qua kỹ thuật masking, buộc mô hình phải học các biểu diễn ngữ nghĩa mạnh mẽ từ dữ liệu không gán nhãn. TimeSformer áp dụng cơ chế divided space-time attention, tính toán attention lần lượt theo chiều không gian và thời gian để giảm độ phức tạp. Tuy nhiên, các nghiên cứu này chủ yếu tập trung vào việc cải thiện độ chính xác mô hình mà ít quan tâm đến khía cạnh triển khai và vận hành ở quy mô lớn.

2.3. Các hệ thống xử lý dữ liệu lớn cho kiểm duyệt nội dung

Một khía cạnh quan trọng nhưng thường bị bỏ qua trong các nghiên cứu học thuật là kiến trúc hệ thống cho việc xử lý luồng dữ liệu liên tục theo thời gian thực. Trong khi các công trình về mô hình AI thường được đánh giá trên các tập dữ liệu tĩnh đã được tải sẵn, các hệ thống thực tế phải đổi mới với yêu cầu xử lý hàng triệu video mới mỗi ngày. Apache Kafka đã trở thành tiêu chuẩn công nghiệp cho lớp hàng đợi thông điệp nhờ khả năng chịu lỗi cao và thông lượng lớn. Đối với lớp xử lý tính toán, Apache Spark Structured Streaming và Apache Flink là hai lựa chọn hàng đầu. Flink nổi bật với mô hình xử lý dòng thuận túy và độ trễ cực thấp, trong khi Spark với mô hình micro-batch có ưu thế về hệ sinh thái phong phú và khả năng tích hợp tốt với các thư viện Python như PyTorch và TensorFlow.

Tuy nhiên, các tài liệu nghiên cứu hiện tại về kiểm duyệt nội dung rất ít đề cập đến việc tích hợp các mô hình AI phức tạp vào trong các đường ống xử lý dữ liệu lớn, tạo nên một khoảng trống đáng kể giữa nghiên cứu và ứng dụng thực tiễn.

2.4. Khoảng trống nghiên cứu và hướng tiếp cận của đề tài

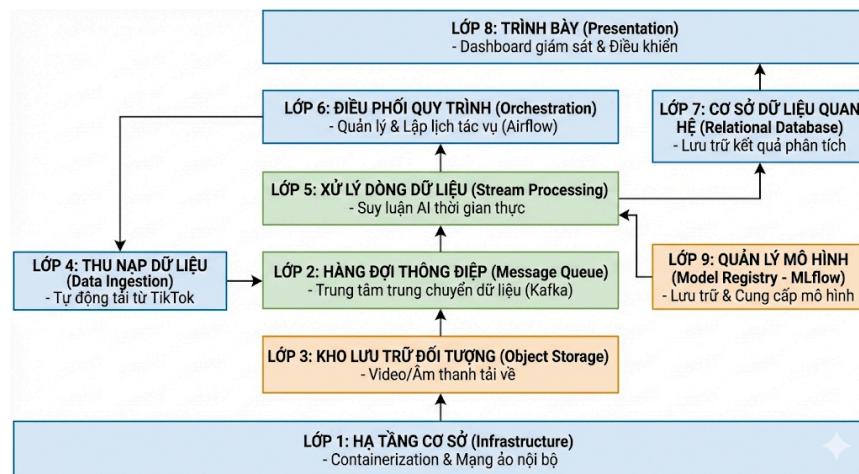
Phân tích tổng hợp các công trình liên quan cho thấy ba khoảng trống nghiên cứu chính cần được giải quyết. Khoảng Trống thứ nhất liên quan đến sự thiếu vắng các nghiên cứu về kiểm duyệt nội dung đa phương thức cho ngôn ngữ tiếng Việt. Phần lớn các mô hình hiện tại được huấn luyện và đánh giá trên dữ liệu tiếng Anh, trong khi tiếng Việt với các đặc thù về từ vựng, ngữ pháp và đặc biệt là các hình thức viết tắt, tiếng lóng trên mạng xã hội đòi hỏi các mô hình được tinh chỉnh chuyên biệt. Khoảng trống thứ hai nằm ở sự thiếu kết nối giữa các mô hình AI tiên tiến và kiến trúc hệ thống xử lý dữ liệu lớn. Các nghiên cứu hiện tại thường tập trung vào một trong hai khía cạnh này một cách riêng lẻ mà không trình bày cách tích hợp chúng thành một hệ thống hoàn chỉnh từ đầu đến cuối. Khoảng trống thứ ba liên quan đến quy trình vận hành máy học (MLOps), bao gồm việc theo dõi thực nghiệm, quản lý phiên bản mô hình và tự động hóa việc huấn luyện lại để thích ứng với các dạng nội dung độc hại mới.

Để giải quyết các khoảng trống trên, đề tài này đề xuất một hệ thống toàn diện kết hợp ba thành phần chính. Thành phần đầu tiên là mô hình AI đa phương thức sử dụng CafeBERT cho văn bản tiếng Việt và VideoMAE cho video, được kết nối thông qua cơ chế cross-attention để phát hiện các mối tương quan giữa nội dung văn bản và hình ảnh. Thành phần thứ hai là kiến trúc xử lý dữ liệu song song Lô và Dòng (Dual-layer Batch & Stream Architecture), tích hợp Apache Kafka và Spark Structured Streaming để xử lý luồng video theo thời gian thực với khả năng mở rộng cao. Thành phần thứ ba là quy trình MLOps dựa trên Apache Airflow và MLflow, đảm bảo hệ thống có khả năng tự cải thiện liên tục. Với cách tiếp cận này, đề tài hướng đến việc thu hẹp khoảng cách giữa nghiên cứu học thuật và ứng dụng thực tiễn trong lĩnh vực kiểm duyệt nội dung số.

Chương 3: PHƯƠNG PHÁP ĐỀ XUẤT VÀ KIẾN TRÚC HỆ THỐNG

3.1. Tổng quan kiến trúc hệ thống

Hệ thống được xây dựng dựa trên kiến trúc xử lý dữ liệu lai (Hybrid Data Processing Architecture), hay còn được biết đến là mô hình Lambda, một kiến trúc tham chiếu phổ biến trong các hệ thống xử lý dữ liệu lớn, cho phép kết hợp hài hòa giữa khả năng xử lý theo lô (batch processing) và xử lý thời gian thực (stream processing). Triết lý của kiến trúc Lambda là duy trì một đường ống dữ liệu liên tục từ nguồn đến người dùng cuối, đồng thời cung cấp cơ chế huấn luyện lại mô hình một cách định kỳ để mô hình luôn được cập nhật với các xu hướng nội dung mới. Để hiện thực hóa triết lý này, chúng tôi đã thiết kế một kiến trúc phân tầng gồm chín lớp, mỗi lớp đảm nhận một vai trò cụ thể trong chuỗi xử lý dữ liệu.



Hình 1: Kiến trúc hệ thống phân tầng dựa trên mô hình xử lý dữ liệu lai (Lambda Architecture)

Lớp thứ nhất đóng vai trò hạ tầng cơ sở, cung cấp nền tảng container hóa và mạng ảo nội bộ để các thành phần có thể giao tiếp với nhau một cách an toàn và hiệu quả. Tất cả các thành phần của hệ thống đều được đóng gói theo phương pháp container hóa, đảm bảo tính nhất quán khi triển khai trên các môi trường khác nhau.

Lớp thứ hai là lớp hàng đợi thông điệp, hoạt động như trung tâm trung chuyển dữ liệu giữa các thành phần. Lớp này sử dụng hệ thống hàng đợi phân tán có khả năng xử lý thông lượng cao và chịu lỗi tốt, đảm bảo không mất mát dữ liệu ngay cả khi một số thành phần tạm thời ngưng hoạt động.

Lớp thứ ba là kho lưu trữ đối tượng, nơi lưu trữ các tập tin video và âm thanh đã tải về. Kho lưu trữ này được thiết kế tương thích với giao thức lưu trữ đám mây phổ biến, cho phép truy xuất nhanh chóng từ các thành phần xử lý.

Lớp thứ tư đảm nhận vai trò thu nạp dữ liệu, bao gồm các thành phần tự động tìm kiếm và tải video từ nền tảng Tik Tok. Quá trình này được thiết kế để mô phỏng hành vi người dùng thực, vượt qua các cơ chế phát hiện robot của nền tảng đích.

Lớp thứ năm là lớp xử lý dòng dữ liệu, nơi diễn ra quá trình suy luận của các mô hình trí tuệ nhân tạo. Lớp này tiếp nhận dữ liệu từ hàng đợi thông điệp, thực hiện phân tích đa phương thức, và đưa ra kết luận về tính chất của nội dung theo thời gian gần thực.

Lớp thứ sáu là lớp điều phối quy trình, quản lý và lập lịch cho các tác vụ thu thập, xử lý và huấn luyện mô hình. Lớp này đảm bảo các quy trình tự động được thực thi đúng thứ tự và có khả năng xử lý các tình huống lỗi.

Lớp thứ bảy là cơ sở dữ liệu quan hệ, lưu trữ kết quả phân tích, bao gồm điểm số dự đoán, nhãn phân loại, và các thông tin thống kê phục vụ cho việc theo dõi và đánh giá hiệu năng hệ thống.

Lớp thứ tám là lớp trình bày, cung cấp giao diện đồ họa trực quan cho phép người vận hành theo dõi trạng thái hệ thống, xem kết quả phân tích, và điều khiển các tác vụ thông qua bảng điều khiển tương tác.

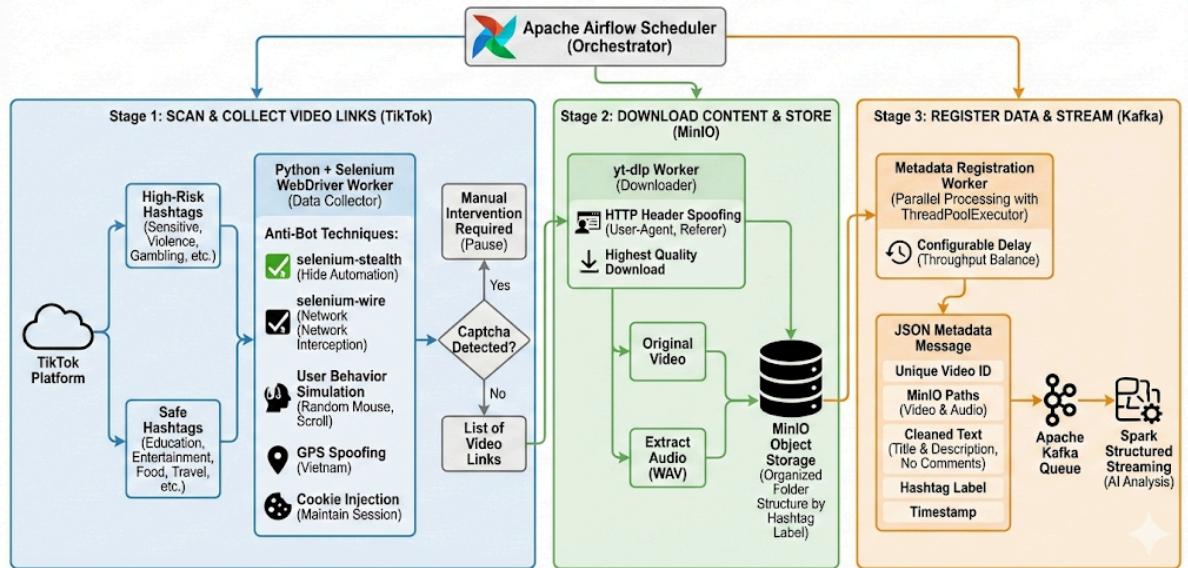
Lớp thứ chín là lớp quản lý mô hình (Model Registry - MLflow), đóng vai trò trung tâm trong quy trình MLOps. Lớp này chịu trách nhiệm lưu trữ phiên bản, theo dõi hiệu năng huấn luyện, và tự động cung cấp các mô hình tối ưu nhất cho lớp xử lý dòng dữ liệu mà không cần dừng hệ thống.

Kiến trúc tám lớp này tạo nên một hệ thống hoàn chỉnh, có khả năng tự động hóa cao từ khâu thu thập dữ liệu đến chia sẻ kết quả, đáp ứng yêu cầu giám sát nội dung theo thời gian thực với quy mô lớn.

3.2. Quy trình thu thập dữ liệu

Mục tiêu của quy trình thu thập dữ liệu là xây dựng một tập dữ liệu có gán nhãn cân bằng giữa nội dung độc hại và an toàn để phục vụ cho việc huấn luyện mô hình. Quy trình này được thiết kế hoàn toàn tự động và được điều phối bởi hệ thống lập lịch Apache Airflow, cho phép thu thập dữ liệu liên tục mà không cần can thiệp thủ công.

Automated TikTok Data Collection and Streaming Pipeline for Balanced Dataset Construction (Orchestrated by Apache Airflow)



Hình 2: Sơ đồ luồng dữ liệu từ TikTok đến Kafka

Giai đoạn đầu tiên của quy trình là quét và thu thập liên kết video từ nền tảng TikTok. Phương pháp thu thập chính được áp dụng là kỹ thuật quét theo từ khóa thông qua các hashtag mục tiêu. Nhóm đã xây dựng hai bộ hashtag phục vụ cho việc thu thập dữ liệu cân bằng. Bộ hashtag thứ nhất tập trung vào các nội dung có nguy cơ cao, bao gồm các chủ đề liên quan đến nội dung nhạy cảm, bạo lực, cờ bạc và các tệ nạn xã hội thường xuất hiện trên mạng xã hội Việt Nam. Bộ hashtag thứ hai bao gồm các nội dung giáo dục và giải trí lành mạnh như âm thực, du lịch, học ngoại ngữ và thú cưng.

Công cụ thu thập được xây dựng dựa trên nền tảng Python kết hợp Selenium WebDriver để tự động hóa trình duyệt. Để vượt qua các cơ chế chống bot của TikTok, hệ thống tích hợp nhiều kỹ thuật tiên tiến. Thư viện selenium-stealth được sử dụng để che giấu các dấu vết tự động hóa, khiến trình duyệt được điều khiển tự động có hành vi giống với trình duyệt thông thường. Hệ thống cũng giả lập hành vi người dùng thực thông qua các chuyển động chuột ngẫu nhiên và thời gian cuộn trang không đều đặn. Ngoài ra, tọa độ GPS được giả lập tại Việt Nam để TikTok để xuất nội dung phù hợp với khu vực địa lý mục tiêu. Để duy trì phiên đăng nhập, hệ thống sử dụng kỹ thuật tiêm cookie từ phiên đăng nhập thực trên trình duyệt thật. Hệ thống sử dụng thư viện selenium wire kết hợp với selenium-stealth để giảm thiểu khả năng bị phát hiện là robot. Trong trường hợp gặp captcha, quá trình thu thập sẽ tạm dừng và cần can thiệp thủ công để tiếp tục.

Giai đoạn thứ hai là tải nội dung chi tiết về máy chủ. Sau khi thu thập được danh sách liên kết video, hệ thống sử dụng công cụ yt-dlp để tải video với chất lượng cao nhất có thể. Để vượt qua các rào cản truy cập, hệ thống áp dụng kỹ thuật giả mạo header HTTP bằng cách mô phỏng User-Agent và Referer của trình duyệt thực. Mỗi video sau khi tải về sẽ được trích xuất phần âm thanh thành định dạng WAV để phục vụ cho nhánh xử lý âm thanh. Tiếp theo, cả video và âm thanh được tải lên kho lưu trữ đối tượng MinION với cấu trúc thư mục được tổ chức theo nhãn phân loại gợi ý từ hashtag.

Giai đoạn cuối cùng là đăng ký dữ liệu vào hệ thống xử lý dòng. Sau khi tải lên kho lưu trữ thành công, worker tạo một thông điệp JSON chứa đầy đủ metadata bao gồm định danh duy nhất của video, đường dẫn lưu trữ trên MinIOcho cả video và âm thanh, nội dung văn bản đã được làm sạch từ tiêu đề và mô tả video (không bao gồm bình luận trong luồng streaming), nhãn gợi ý từ hashtag, và dấu thời gian xử lý. Thông điệp này được gửi vào hàng đợi Apache Kafka để lớp xử lý dòng Spark Structured Streaming Tiếp nhận và thực hiện phân tích bằng mô hình trí tuệ nhân tạo. Để tránh bị nền tảng TikTok phát hiện và chặn, hệ thống áp dụng chiến lược xử lý song song với ThreadPoolExecutor. Khoảng nghỉ giữa các lần tải có thể được cấu hình qua biến môi trường nhưng mặc định không được kích hoạt để tối đa hóa throughput. Chiến lược này tạo nên sự cân bằng giữa tốc độ cập nhật dữ liệu cho Dashboard giám sát và việc tránh bị hạn chế truy cập.

3.3. Kiến trúc mô hình AI đa phương thức

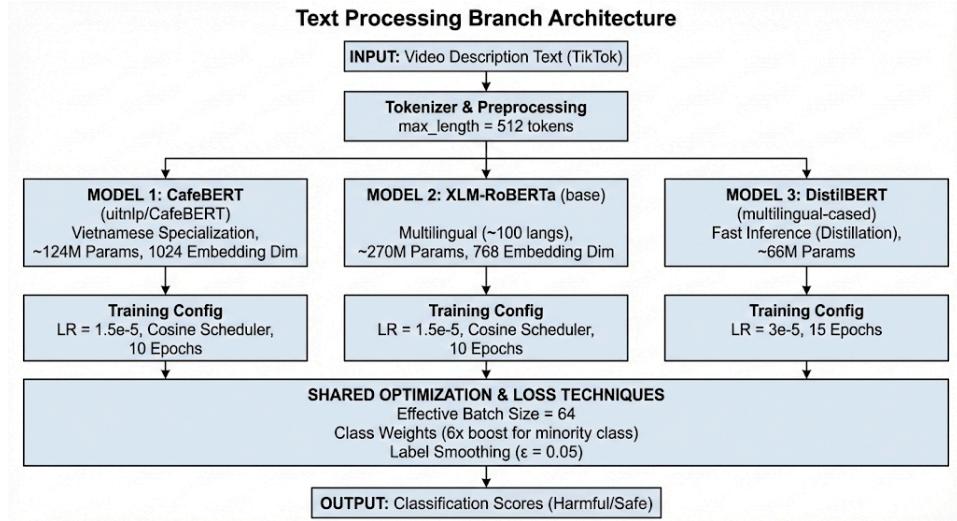
3.3.1. Nhánh xử lý văn bản

Để phân tích nội dung văn bản mô tả video, chúng tôi đã thực nghiệm với ba mô hình ngôn ngữ tiền huấn luyện tiên tiến, mỗi mô hình có đặc điểm riêng phù hợp với các tình huống khác nhau.

Mô hình thứ nhất là CafeBERT (uitnlp/CafeBERT), một kiến trúc Transformer được huấn luyện đặc biệt cho ngôn ngữ tiếng Việt với khoảng 124 triệu tham số. Mô hình này có ưu điểm nổi bật trong việc xử lý các đặc thù ngôn ngữ Việt Như từ vựng địa phương, tiếng lóng mạng xã hội và các cấu trúc ngữ pháp đặc trưng. Embedding dimension của mô hình là 1024, cho phép mã hóa các đặc trưng ngữ nghĩa phong phú.

Mô hình thứ hai là XLM-RoBERTa (xlm-roberta-base) với khoảng 270 triệu tham số, được huấn luyện trên hơn 100 ngôn ngữ. Mô hình này phù hợp với các nội dung có sự pha trộn ngôn ngữ, điều rất phổ biến trên TikTok khi người dùng thường xen kẽ tiếng Việt, tiếng Anh và các ngôn ngữ khác. Embedding dimension là 768.

Mô hình thứ ba là DistilBERT (distilbert-base-multilingual-cased) với khoảng 66 triệu tham số, được tạo ra bằng phương pháp knowledge distillation. Mô hình này có tốc độ inference nhanh hơn đáng kể, phù hợp với các ứng dụng yêu cầu low latency.

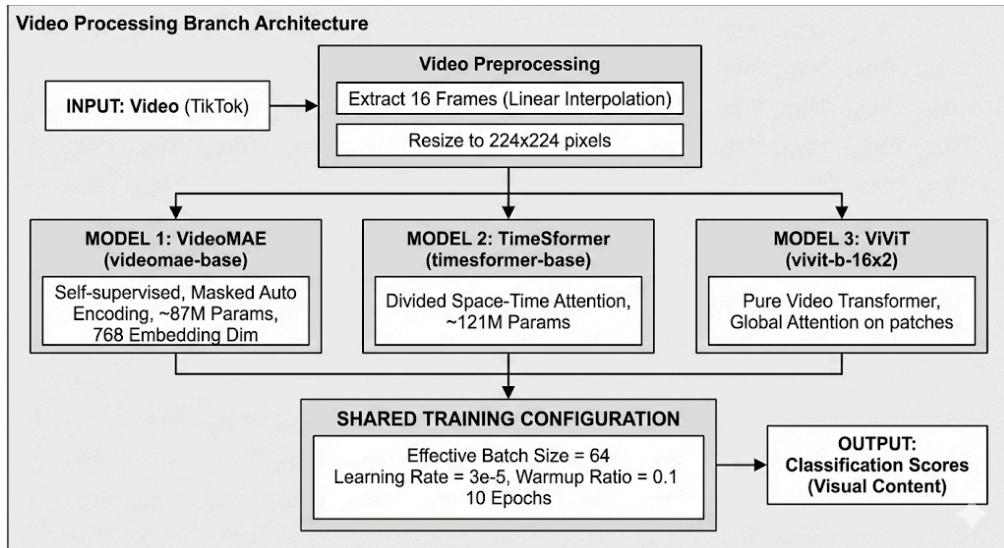


Hình 3: Kiến trúc và cấu hình thực nghiệm các mô hình cho nhánh xử lý văn bản.

Quá trình huấn luyện các mô hình văn bản được thực hiện với các cấu hình tối ưu: max_length = 512 tokens, effective_batch_size = 64, learning rate = 1.5e-5 với cosine scheduler, và 10 epochs. Riêng mô hình DistilBERT được huấn luyện và learning rate = 3e-5 và 15 epochs do kiến trúc nhẹ hơn đòi hỏi nhiều bước huấn luyện hơn. Để giải quyết vấn đề class imbalance nghiêm trọng (lớp độc hại chiếm tỷ lệ nhỏ), chúng tôi áp dụng class weights với hệ số boost 6x cho lớp thiểu số, kết hợp với label smoothing ($\epsilon = 0.05$) để cải thiện khả năng generalization của mô hình.

3.3.2. Nhánh xử lý video

Để phân tích nội dung thị giác của video, chúng tôi thực nghiệm với ba kiến trúc Video Transformer tiên tiến, mỗi kiến trúc áp dụng cách tiếp cận khác nhau để encode thông tin spatial và temporal.



Hình 4: Kiến trúc và cấu hình thực nghiệm các mô hình cho nhánh xử lý video.

Mô hình thứ nhất là VideoMAE (MCG-NJU/videomae-base-finetuned-kinetics), sử dụng phương pháp self-supervised learning thông qua kỹ thuật Masked Auto Encoding, có khoảng 87 triệu

tham số và embedding dimension 768. Phương pháp này buộc mô hình học các biểu diễn ngữ nghĩa mạnh mẽ bằng cách yêu cầu tái tạo các phần video bị mask trong quá trình pre-training.

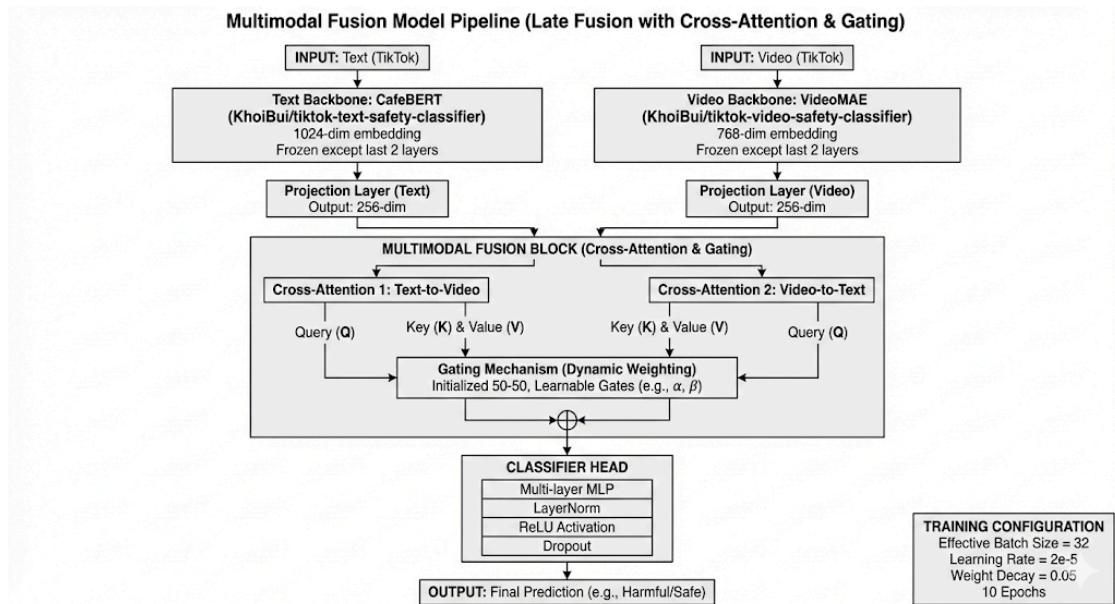
Mô hình thứ hai là TimeSformer (facebook/timesformer-base-finetuned-k400), áp dụng cơ chế Divided Space-Time Attention với khoảng 121 triệu tham số. Thay vì tính toán attention đồng thời trên cả hai chiều, mô hình này lần lượt tính spatial attention (giữa các vùng trong một frame) rồi temporal attention (giữa các frames), giúp giảm độ phức tạp tính toán đáng kể.

Mô hình thứ ba là ViViT (google/vivit-b-16x2-kinetics400), kiến trúc Video Vision Transformer thuần túy, áp dụng trực tiếp global attention trên toàn bộ các patches từ tất cả các frames.

Quy trình preprocessing video bao gồm việc trích xuất 16 frames phân bố đều từ mỗi video bằng linear interpolation, sau đó resize về 224×224 pixels. Số lượng 16 frames được chọn để cân bằng giữa việc bao quát đủ nội dung video và chi phí tính toán. Quá trình training sử dụng effective batch size = 64, learning rate = $3e-5$ với warmup ratio = 0.1 và 10 epochs.

3.3.3. Mô hình kết hợp đa phương thức (Multimodal Fusion)

Để tận dụng tối đa thông tin từ cả hai nguồn dữ liệu text và video, chúng tôi thiết kế một multimodal fusion model sử dụng chiến lược late fusion kết hợp cross-attention mechanism. Đây là một trong những phương pháp tiên tiến hiện nay để tích hợp thông tin đa nguồn vì nó cho phép mô hình học được các tương quan phức tạp giữa các modalities.



Hình 5: Kiến trúc và cấu hình thực nghiệm các mô hình kết hợp đa phương thức.

Mô hình fusion sử dụng hai backbone đã được fine-tune: CafeBERT cho text (1024-dim embedding) và VideoMAE cho video (768-dim embedding). Cả hai backbone được đăng ký trên Hugging Face Hub tại KhoiBui/tiktok-text-safety-classifier và KhoiBui/tiktok-video-safety-classifier. Để tiết kiệm tài nguyên, phần lớn tham số của backbone được freeze, chỉ 2 layers cuối được unfreeze để fine-tune cùng fusion head. Chúng tôi sử dụng các projection layers để đưa các representations từ hai nguồn về cùng một common space có fusion hidden dim = 256.

Cross-attention hai chiều là thành phần cốt lõi của fusion model. Theo chiều thứ nhất (text-to-video attention), text representation được sử dụng làm query để tìm kiếm các thông tin liên quan trong video representation. Theo chiều ngược lại (video-to-text attention), video trở thành query

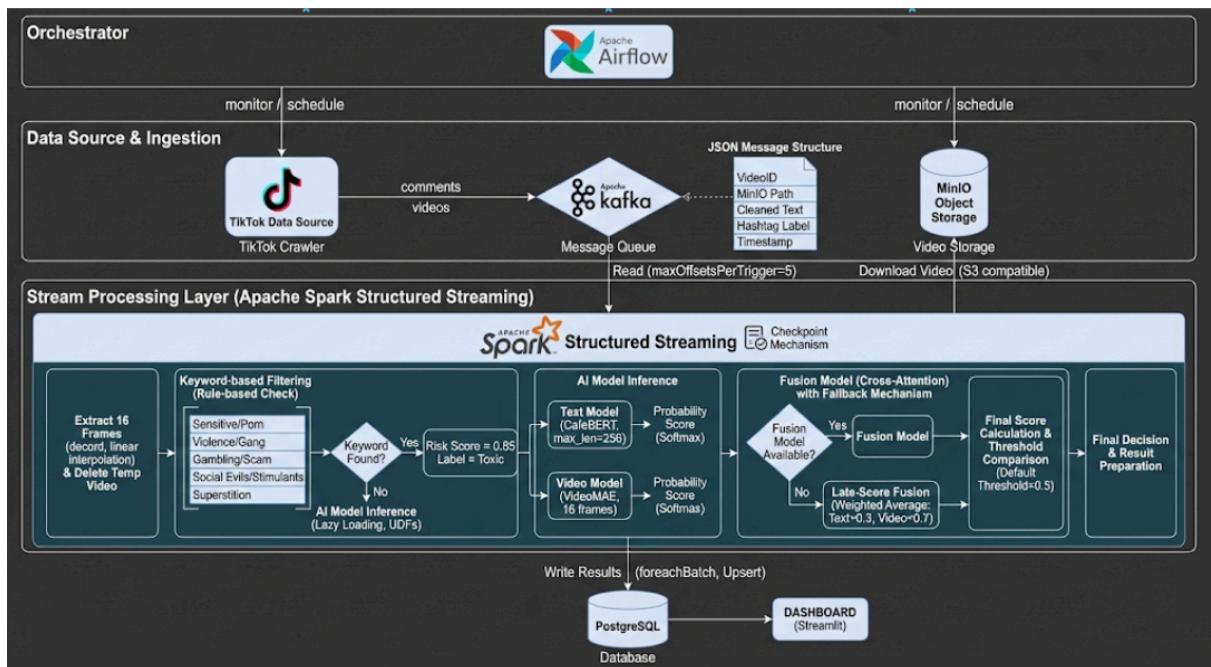
để tìm ngữ cảnh trong text. Cơ chế này cho phép mô hình phát hiện các trường hợp không nhất quán giữa text và video content.

Ngoài ra, chúng tôi thiết kế một gating mechanism để tự động điều chỉnh mức độ đóng góp của từng modality. Cơ Chế gates học được chiến lược dynamic weighting: khi video bị mờ hoặc tối, hệ thống tự động tăng weight cho text branch và ngược lại.

Kết quả được đưa qua classifier head gồm nhiều layers với LayerNorm, ReLU activation và Dropout để đưa ra prediction cuối cùng. Training config: effective batch size = 32, learning rate = 2e-5, weight_decay = 0.05, và 10 epochs. Modality weights được khởi tạo 50-50, sau đó gate sẽ học điều chỉnh động.

3.4. Quy trình xử lý dòng thời gian thực

Một trong những yêu cầu quan trọng của hệ thống là khả năng phân tích nội dung ngay khi video được thu thập, không cần chờ tích lũy thành lô lớn như trong các hệ thống xử lý theo mẻ truyền thống. Để đáp ứng yêu cầu này, chúng tôi xây dựng một pipeline xử lý dòng dữ liệu dựa trên Apache Spark Structured Streaming.



Hình 6: Sơ đồ pipeline Spark Structured Streaming

Quy trình xử lý bắt đầu khi Spark Structured Streaming đóng vai trò consumer, liên tục đọc các thông điệp từ hàng đợi Kafka thông qua topic được cấu hình sẵn. Mỗi thông điệp có cấu trúc JSON bao gồm năm trường dữ liệu: định danh duy nhất của video, đường dẫn đến tệp video trên kho lưu trữ MinIO, nội dung văn bản đã được làm sạch, nhãn gợi ý từ quá trình thu thập dựa trên hashtag, và dấu thời gian ghi nhận. Để kiểm soát tốc độ xử lý và tránh quá tải hệ thống, Spark được cấu hình với giới hạn tối đa năm offset mỗi lần kích hoạt, đồng thời sử dụng cơ chế checkpoint để đảm bảo khả năng phục hồi khi xảy ra lỗi.

Sau khi nhận được thông điệp, hệ thống tiến hành tải tệp video từ kho lưu trữ MinIO sử dụng giao thức tương thích S3. Từ mỗi video, hệ thống sử dụng thư viện decord để đọc và trích xuất các khung hình với phương pháp nội suy tuyến tính, lấy 16 khung hình phân bố đều từ đầu đến cuối video. Số lượng 16 khung hình này được chọn dựa trên cấu hình tối ưu của mô hình VideoMAE và

yêu cầu cân bằng giữa độ bao phủ nội dung và chi phí tính toán. Sau khi xử lý xong, tệp video tạm được xóa ngày để giải phóng dung lượng đĩa.

Trước khi gọi các mô hình học sâu tồn kén tài nguyên, hệ thống thực hiện một bước kiểm tra nhanh dựa trên luật. Một danh sách từ khóa cấm được xây dựng bao gồm hơn 60 thuật ngữ tiếng Việt phổ biến, được tổ chức thành năm nhóm chuyên đề. Nhóm thứ nhất bao gồm các từ khóa liên quan đến nội dung nhạy cảm và khiêu dâm. Nhóm thứ hai tập trung vào bạo lực và các nội dung giang hồ. Nhóm thứ ba bao gồm các thuật ngữ về cờ bạc và lừa đảo tài chính. Nhóm thứ tư liên quan đến các tệ nạn xã hội và chất kích thích. Nhóm thứ năm bao gồm các nội dung tâm linh và mê tín dị đoan. Nếu nội dung văn bản chứa bất kỳ từ khóa nào trong danh sách này, hệ thống lập tức gán điểm rủi ro là 0.85 và phân loại nội dung là độc hại mà không cần thực hiện suy luận bằng mô hình trí tuệ nhân tạo. Cách tiếp cận kết hợp này giúp tăng tốc đáng kể quá trình xử lý các trường hợp rõ ràng đồng thời giảm tải cho GPU.

Đối với các nội dung không khớp với danh sách từ khóa, hệ thống tiến hành suy luận bằng hai nhánh mô hình trí tuệ nhân tạo được đăng ký dưới dạng User-Defined Functions trong Spark. Nhánh thứ nhất sử dụng mô hình CafeBERT đã được tinh chỉnh để phân tích nội dung văn bản từ caption và bình luận, với độ dài tối đa 256 token, trả về điểm xác suất thuộc lớp độc hại thông qua hàm softmax. Nhánh thứ hai sử dụng mô hình VideoMAE đã được tinh chỉnh để phân tích 16 khung hình đã trích xuất, trả về điểm xác suất tương tự. Để tối ưu hiệu năng, hệ thống áp dụng chiến lược tải lùi, tức là các mô hình chỉ được tải vào bộ nhớ lần đầu tiên khi cần thiết và được giữ trong biến toàn cục để tái sử dụng cho các batch tiếp theo.

Điểm số cuối cùng được tính bằng phương pháp trung bình có trọng số giữa điểm từ nhánh văn bản và nhánh video. Theo mặc định, trọng số dành cho văn bản là 0.3 và trọng số dành cho video là 0.7, phản ánh tầm quan trọng lớn hơn của thông tin thị giác trên nền tảng video ngắn. Cả hai trọng số này đều được cho phép cấu hình thông qua biến môi trường để thuận tiện cho việc thử nghiệm. Quyết định phân loại cuối cùng được đưa ra bằng cách so sánh điểm số trung bình với ngưỡng quyết định mặc định là 0.5, ngưỡng này cũng có thể được điều chỉnh linh hoạt.

Lưu ý rằng trong pipeline streaming hiện tại, hệ thống ưu tiên sử dụng mô hình Fusion. Với cơ chế cross-attention để đạt độ chính xác cao nhất. Khi mô hình Fusion không thể tải được (do lỗi hoặc chưa được đăng ký), hệ thống tự động fallback chuyển sang phương pháp late-score (tổng có trọng số điểm từ hai mô hình riêng biệt với hệ số 0.3 cho văn bản và 0.7 cho video). Cơ chế fallback này đảm bảo pipeline luôn hoạt động ổn định trong mọi tình huống.

Kết quả phân tích được ghi vào cơ sở dữ liệu PostgreSQL thông qua cơ chế foreach Batch của Spark. Trước khi ghi, batch dữ liệu được lưu tạm vào bộ nhớ để tránh việc chạy lại các UDF tồn kén nhiều lần. Hệ thống cũng thực hiện loại bỏ trùng lặp trong cùng một batch dựa trên định danh video, giữ lại bản ghi cuối cùng cho mỗi video. Mỗi bản ghi chứa đầy đủ thông tin bao gồm định danh video, nội dung văn bản gốc, nhãn con người từ quá trình thu thập, điểm số và phán quyết từ từng nhánh, điểm trung bình có trọng số, ngưỡng quyết định, phán quyết cuối cùng, và dấu thời gian xử lý. Hệ thống sử dụng cơ chế upsert để xử lý các trường hợp video được phân tích lại, đảm bảo dữ liệu luôn được cập nhật mà không tạo bản ghi trùng lặp và đồng thời cập nhật trường thời gian xử lý mới nhất.

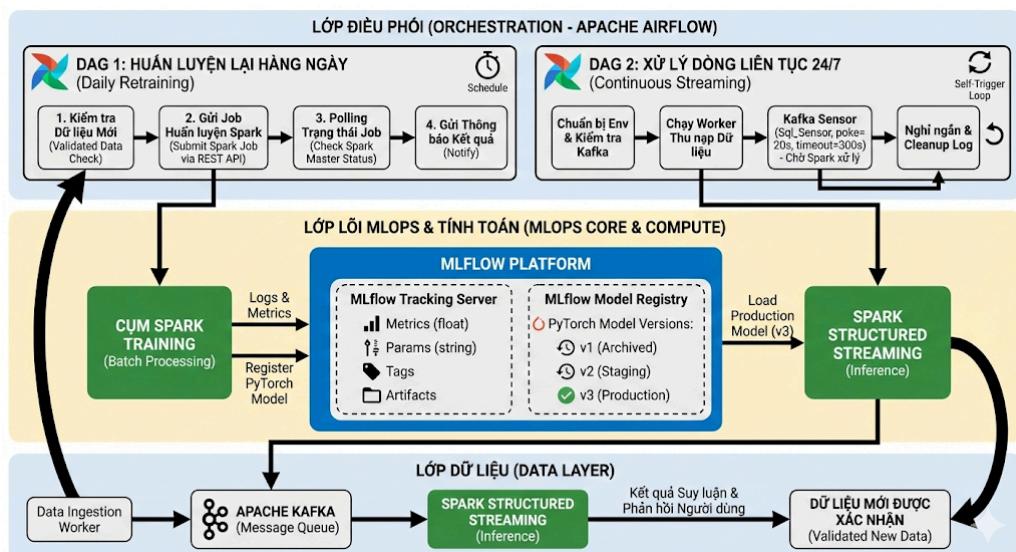
Cuối cùng, một Dashboard được xây dựng trên nền tảng Streamlit để trực quan hóa kết quả phân tích theo thời gian thực. Dashboard truy vấn dữ liệu từ PostgreSQL và hiển thị danh sách các video mới nhất đã được xử lý, thống kê tổng hợp về tỷ lệ nội dung độc hại phát hiện được theo từng phiên, và cho phép người quản trị xem chi tiết từng video bao gồm cả điểm số từ từng nhánh để đánh giá chất lượng phân loại.

3.5. Tích hợp vận hành máy học (MLOps)

Một hệ thống trí tuệ nhân tạo ứng dụng trong thực tế cần có khả năng tự cải thiện theo thời gian để thích ứng với các dạng nội dung độc hại mới. Để đáp ứng yêu cầu này, chúng tôi thiết kế một quy trình vận hành máy học toàn diện dựa trên nền tảng MLflow và Apache Airflow.

Thành phần đầu tiên của quy trình MLOps là hệ thống theo dõi thực nghiệm. Mỗi lần huấn luyện mô hình, hệ thống tự động ghi lại đầy đủ thông tin vào MLflow Tracking Server. Các thí nghiệm được tổ chức theo loại mô hình với định danh riêng biệt cho mô hình văn bản, mô hình video, và mô hình kết hợp. Mỗi lần chạy huấn luyện được đặt tên theo quy ước bao gồm loại mô hình, tên mô hình cụ thể và dấu thời gian. Hệ thống ghi lại đầy đủ các chỉ số đánh giá dưới dạng số thực, các siêu tham số huấn luyện dưới dạng chuỗi, và các thẻ đánh dấu để phân loại và tìm kiếm sau này.

Thành phần thứ hai là hệ thống đăng ký mô hình. Khi quá trình huấn luyện hoàn tất, mô hình PyTorch được đăng ký trực tiếp vào MLflow Model Registry với định dạng chuẩn của PyTorch. Điều này cho phép lưu trữ nhiều phiên bản của cùng một mô hình với các trạng thái khác nhau như đang thử nghiệm, sẵn sàng triển khai, và lưu trữ lịch sử. Nếu không có đối tượng mô hình PyTorch, hệ thống vẫn có thể lưu các tệp artifact của checkpoint để tham khảo sau này.



Hình 7: Quy trình vận hành máy học (MLOps) tự động tích hợp Apache Airflow và MLflow.

Thành phần thứ ba là quy trình tự động huấn luyện lại. Một DAG Airflow được lên lịch chạy hàng ngày để kiểm tra và thực hiện huấn luyện lại mô hình. Quy trình này bao gồm bốn bước chính. Bước đầu tiên kiểm tra xem có dữ liệu mới được thu thập và xác nhận bởi người dùng hay không. Bước thứ hai gửi công việc huấn luyện đến cụm Spark thông qua REST API, với các biến môi trường được cấu hình sẵn bao gồm địa chỉ MLflow Tracking Server để mô hình được đăng ký tự động sau khi huấn luyện xong. Bước thứ ba liên tục kiểm tra trạng thái công việc bằng cách truy vấn Spark Master cho đến khi công việc hoàn thành, thất bại hoặc bị hủy. Bước cuối cùng gửi thông báo về kết quả huấn luyện.

Thành phần thứ tư là cơ chế xử lý dòng liên tục không gián đoạn. Một DAG Airflow khác điều phối toàn bộ pipeline streaming, bao gồm các bước chuẩn bị môi trường, kiểm tra hạ tầng Kafka, chạy worker thu nạp dữ liệu, xác nhận Spark đang xử lý thông qua Sql_Sensor được cấu hình với poke_interval là 20 giây và timeout là 300 giây (5 phút) để đảm bảo không block quá lâu nếu Spark không phản hồi, và quan trọng nhất là tự động kích hoạt lại chính nó sau khi hoàn thành một chu trình. Cơ chế tự kích hoạt này đảm bảo hệ thống luôn trong trạng thái sẵn sàng thu thập và xử lý dữ

liệu mới 24/7 mà không cần can thiệp thủ công. Giữa các chu trình, hệ thống nghỉ một khoảng thời gian ngắn để giải phóng tài nguyên và đóng các log cũ.

Cơ chế này tạo nên một vòng lặp khép kín từ thu thập dữ liệu, phân tích bằng mô hình hiện tại, thu thập phản hồi từ người dùng, huấn luyện lại với dữ liệu mới, và triển khai mô hình cập nhật. Toàn bộ quy trình được tự động hóa và có khả năng tự phục hồi khi gặp lỗi nhờ cơ chế retry được cấu hình trong Airflow.

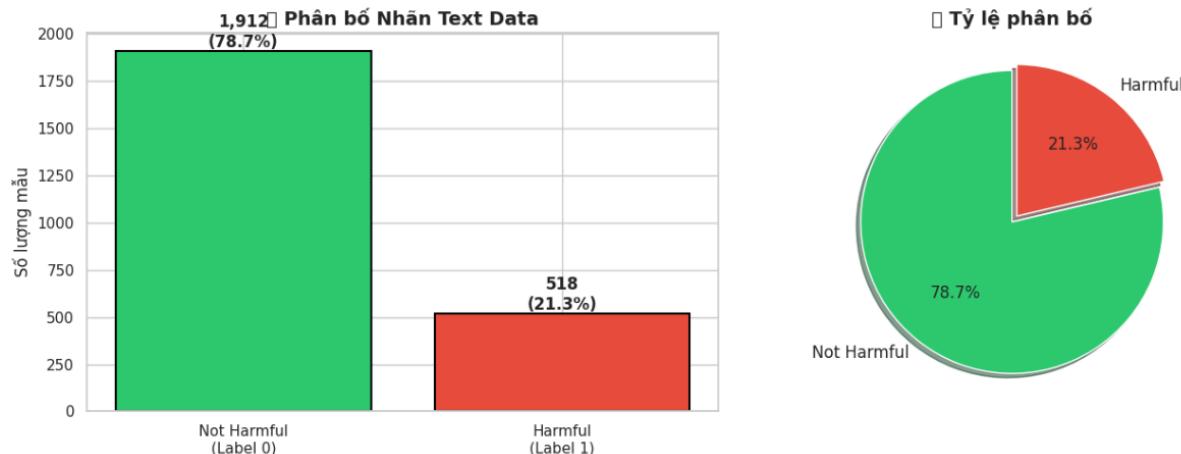
Chương 4: THỰC NGHIỆM VÀ KẾT QUẢ

4.1. Tập dữ liệu

Tập dữ liệu được xây dựng thông qua hệ thống thu thập tự động đã mô tả ở phần trước, tổng cộng thu thập được 2,615 video TikTok từ nhiều nguồn khác nhau. Sau quá trình lọc và tiền xử lý (loại bỏ video bị lỗi, quá ngắn, hoặc không có metadata), số mẫu sử dụng được cho huấn luyện và đánh giá là 1.820 video. Dữ liệu được thu thập trong nhiều đợt, bao gồm các chủ đề đa dạng từ giải trí thông thường đến các nội dung có nguy cơ cao. Cụ thể, dữ liệu được tổng hợp từ ba nguồn chính: tập dữ liệu gốc với 959 video đa ngôn ngữ, tập dữ liệu bổ sung với 838 video, và tập dữ liệu tiếng Việt thuần túy với 818 video. Sau quá trình làm sạch và loại bỏ các mẫu không hợp lệ (video corrupt, thời lượng dưới 1 giây, thiếu text mô tả), 1.820 video được đưa vào sử dụng. Sự đa dạng này giúp mô hình học được các đặc trưng phân biệt giữa nội dung an toàn và độc hại trong nhiều ngữ cảnh khác nhau.

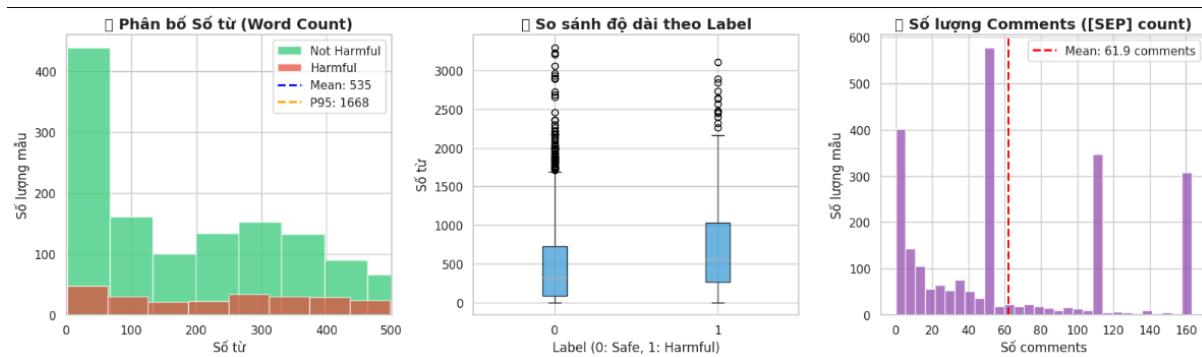
Quá trình gán nhãn được thực hiện thủ công bởi nhóm cộng tác viên theo hướng dẫn nghiêm ngặt, chia thành hai lớp: an toàn và độc hại. Tiêu chí gán nhãn độc hại bao gồm các nội dung bạo lực, nội dung người lớn, quảng cáo cờ bạc, và các hành vi vi phạm pháp luật khác. Tập dữ liệu được chia theo tỷ lệ 80:10:10 cho các tập huấn luyện, kiểm định, và kiểm thử. Tập huấn luyện gồm 1.456 mẫu, tập kiểm định gồm 182 mẫu, và tập kiểm thử gồm 182 mẫu. Tập kiểm thử có phân bố: 137 video an toàn (75,3%) và 45 video độc hại (24,7%), phản ánh tính mất cân bằng thực tế của dữ liệu kiểm duyệt nội dung. Lưu ý rằng một số video không thể xử lý được trong quá trình đánh giá mô hình video do lỗi định dạng hoặc thời lượng quá ngắn, dẫn đến tập kiểm thử cho mô hình video chỉ còn 180 mẫu (ít hơn 2 mẫu so với tập văn bản).

4.1.1. EDA tỉ lệ phân bố nhãn



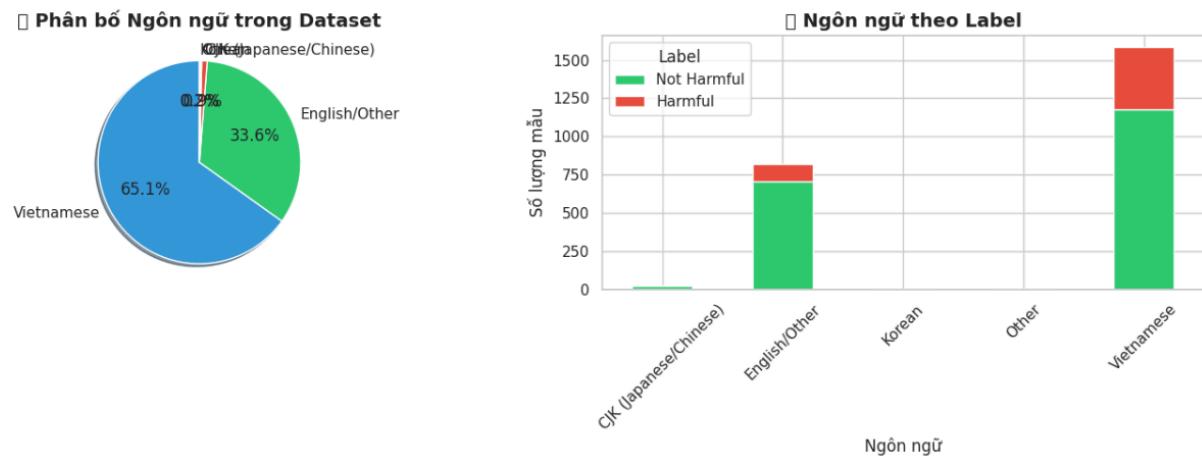
Nhận xét: Dữ liệu bị mất cân bằng đáng kể

4.1.2. EDA phân tích độ dài văn bản



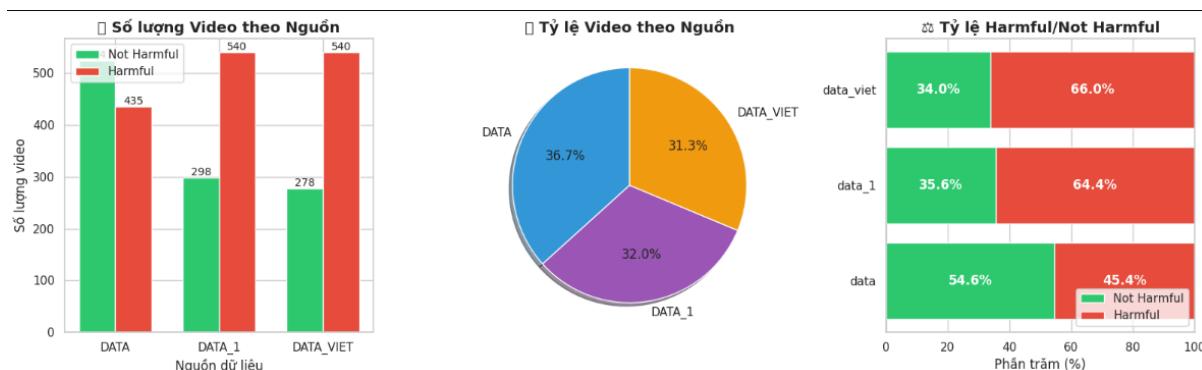
Nhận xét: Phần lớn các câu caption và comment trên TikTok đều ngắn gọn. Phân tích cho thấy độ dài 512 tokens bao phủ được 95% dữ liệu.

4.1.3. EDA thống kê các ngôn ngữ trong bộ dữ liệu



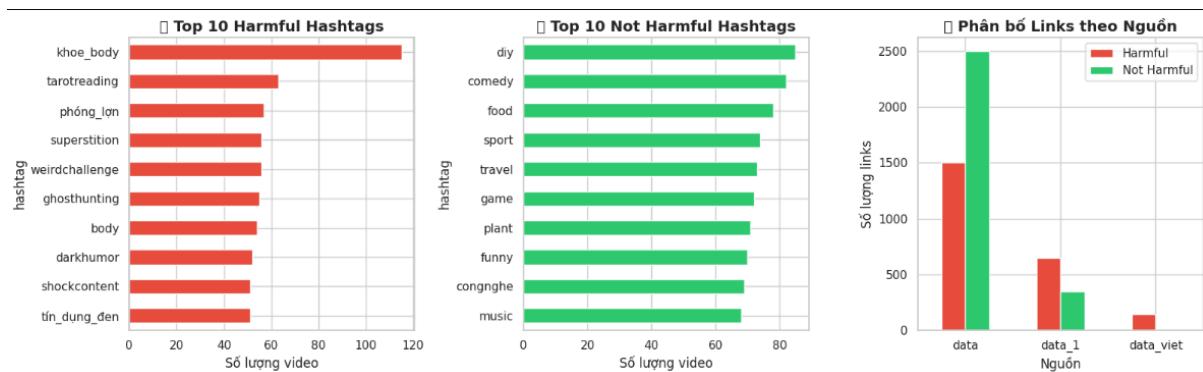
Nhận xét: Bộ dữ liệu phản ánh đúng thực tế của TikTok tại Việt Nam — nơi nội dung chủ yếu là tiếng Việt nhưng có sự giao thoa lớn với các xu hướng quốc tế.

4.1.4. EDA phân tích dữ liệu video



Nhận xét: Phân tích dữ liệu video cho thấy sự đa dạng về nguồn gốc.

4.1.5. EDA phân tích các hashtag trong dữ liệu video



Nhận xét: Phân tích Hashtag chỉ ra rằng ngoài các thẻ phổ biến dùng để tăng tương tác, các video độc hại thường có xu hướng sử dụng các thẻ liên quan đến 'drama' hoặc các từ khóa gây tranh cãi.

4.2. Môi trường thực nghiệm

Hệ thống được triển khai trên máy chủ với hệ điều hành Linux 64-bit, cấu hình phần cứng gồm bộ xử lý đa nhân, 32GB bộ nhớ trong, và card đồ họa có hỗ trợ tính toán song song. Môi trường phần mềm bao gồm các thư viện học sâu hiện đại với khả năng tận dụng đơn vị xử lý đồ họa, nền tảng tính toán phân tán cho xử lý dòng dữ liệu, hệ thống hàng đợi thông điệp phân tán, hệ thống điều phối công việc tự động, nền tảng quản lý vòng đời mô hình máy học, cơ sở dữ liệu quan hệ, và kho lưu trữ đối tượng phân tán. Toàn bộ hệ thống được container hóa để đảm bảo tính nhất quán khi triển khai.

4.3. Các chỉ số đánh giá

Việc đánh giá được thực hiện dựa trên bộ chỉ số chuẩn cho bài toán phân loại nhị phân mất cân bằng, bao gồm: Accuracy (độ chính xác tổng thể), Precision (độ chuẩn - tỷ lệ dự đoán đúng trong các mẫu được dự đoán là dương), Recall (độ phủ - tỷ lệ phát hiện đúng trong các mẫu thực sự dương), và F1-Score (trung bình điều hòa của Precision Và Recall). Do tính chất mất cân bằng dữ liệu, chúng tôi đặc biệt chú trọng F1-Score cho lớp Harmful (F1-Harmful) vì đây là lớp thiểu số cần phát hiện.

4.4. Kết quả so sánh mô hình văn bản

Nhóm tiến hành so sánh ba mô hình ngôn ngữ đã được mô tả ở phần phương pháp trên tập kiểm thử 182 mẫu.

Mô hình	Accuracy	Precision	Recall	F1-Score
CafeBERT (vietnamese)	79.67%	79.24%	79.67%	79.43%
DistilBERT (multilingual)	70.33%	72.57%	70.33%	71.25%
XLM-RoBERT (base)	75.82%	73.04%	75.82%	73.55%

Bảng 1: Kết quả test các mô hình văn bản

Trong số các mô hình văn bản, CafeBERT đạt hiệu quả cao nhất (F1 79,4%), chứng minh sự ưu việt của việc sử dụng mô hình ngôn ngữ tiên huấn luyện chuyên biệt cho tiếng Việt. Việc được học trên ngữ liệu tiếng Việt lớn giúp CafeBERT xử lý tốt hơn các từ lóng và cấu trúc câu đặc thù trên mạng xã hội so với các mô hình đa ngôn ngữ.

XLM-RoBERTa đứng thứ hai (73,6%), cho thấy khả năng chuyển giao tri thức đa ngôn ngữ ở mức khá, nhưng vẫn kém hơn CafeBERT do thiếu sự tinh chỉnh sâu sắc về văn hóa và ngôn ngữ đích. DistilBERT có hiệu suất thấp nhất (71,3%), điều này dễ hiểu do kiến trúc đã được rút gọn (distilled) để tối ưu tốc độ, dẫn đến việc giảm khả năng nắm bắt các đặc trưng ngữ nghĩa phức tạp trong các văn bản ngắn và nhiều như mô tả TikTok.

4.5. Kết quả so sánh mô hình video

Nhóm tiền hành so sánh ba mô hình video Transformer trên tập kiểm thử 180 mẫu.

Mô hình	Accuracy	Precision	Recall	F1-Score
VideoMAE	87.22%	87.39%	87.22%	87.24%
TimeSformer	85.56%	85.58%	85.56%	85.52%
ViViT	78.33%	78.70%	78.33%	78.09%

Bảng 2: Kết quả test các mô hình video

VideoMAE thể hiện sự vượt trội nhất (F1 87,2%) nhờ chiến lược huấn luyện Masked Auto Encoding. Cơ chế này buộc mô hình phải tái tạo lại các phần video bị che, qua đó học được các biểu diễn ngữ cảnh và chuyển động rất mạnh mẽ, đặc biệt hiệu quả trong việc nhận diện các hành vi bắt thường.

TimeSformer bám sát với kết quả 85,5%, đồng thời có ưu thế vượt trội về tốc độ xử lý (2.38 samples/s) nhờ cơ chế Divided Space-Time Attention. Đây là sự lựa chọn cân bằng tốt giữa độ chính xác và hiệu năng thực tế.

Ngược lại, ViViT có kết quả thấp nhất (78,1%). Kiến trúc Pure Transformer của ViViT thường đòi hỏi lượng dữ liệu huấn luyện khổng lồ để hội tụ tốt. Với tập dữ liệu hiện tại, ViViT có thể chưa phát huy hết khả năng so với VideoMAE (vốn tận dụng tốt hơn dữ liệu ít nhờ pre-training task mạnh).

4.6. Kết quả mô hình kết hợp đa phương thức

Mô hình	Accuracy	Precision	Recall	F1-Score
CafeBERT & VideoMAE	79.12%	81%	79%	79.66%

Bảng 3: Kết quả test mô hình kết hợp đa phương thức

Mô hình Fusion (kết hợp CafeBERT và VideoMAE) đạt độ chính xác tổng thể 79,1% và chỉ số F1 trung bình có trọng số (Weighted F1) là 79,7%. Ma trận nhầm lẫn cho thấy 114 mẫu an toàn được nhận diện đúng, 23 mẫu an toàn bị nhầm thành độc hại, 15 mẫu độc hại bị bỏ sót, và 30 mẫu độc hại được phát hiện đúng.

Kết quả này cho thấy một sự đánh đổi quan trọng: so với mô hình văn bản đơn lẻ CafeBERT, mô hình Fusion cải thiện đáng kể độ phủ (Recall) đối với lớp độc hại (tăng từ 55,6% lên 66,7%). Điều này chứng tỏ việc bổ sung thông tin hình ảnh giúp hệ thống "bắt" được nhiều nội dung độc hại hơn mà văn bản bỎ sót. Tuy nhiên, tỷ lệ báo động giả cũng tăng nhẹ, dẫn đến độ chính xác tổng thể giảm nhẹ (79,1% so với 79,7% của Text). Hiệu suất chung vẫn thấp hơn mô hình VideoMAE độc lập (87,2%), cho thấy cần cải thiện kỹ thuật kết hợp để tận dụng tốt hơn sức mạnh của cả hai nhánh.

4.7. Phân tích hiệu năng hệ thống

Về thông lượng xử lý, hệ thống đạt trung bình 1,66 video mỗi giây với mô hình video che mặt nạ, tương đương khả năng xử lý khoảng 6.000 video mỗi giờ. Con số này đáp ứng được nhu cầu giám sát một kênh Tik Tok có quy mô trung bình. Cụ thể, thời gian trích xuất 16 khung hình từ mỗi video mất trung bình 0,2 giây, thời gian suy luận mô hình VideoMAE mất khoảng 0,6 giây (108.78s / 180 samples), và thời gian ghi kết quả vào cơ sở dữ liệu mất khoảng 0,05 giây trên mỗi batch.

Về độ trễ, thời gian từ lúc dữ liệu vào hàng đợi đến khi kết quả được ghi vào cơ sở dữ liệu trung bình dưới 2 giây cho mỗi video. Độ trễ thấp này cho phép hệ thống phản ứng nhanh chóng với các nội dung độc hại.

Về tài nguyên tính toán, hệ thống sử dụng khoảng 4 GB bộ nhớ đồ họa cho quá trình suy luận với kích thước lô là 1. Con số này cho phép triển khai trên các card đồ họa phổ thông, giảm chi phí triển khai thực tế.

Chương 5: Thảo luận

5.1. Phân tích ý nghĩa kết quả

Các kết quả thực nghiệm đã làm sáng tỏ vai trò khác biệt của từng phương thức dữ liệu trong bài toán kiểm duyệt nội dung TikTok.

Thứ nhất, sự vượt trội rõ rệt của mô hình video VideoMAE (F1 87,2%) so với mô hình văn bản tốt nhất CafeBERT Weighted F1 79,4%) khẳng định rằng trên nền tảng video ngắn, thông tin thị giác mang tín hiệu quyết định. Văn bản mô tả thường ngắn, chứa nhiều hashtag, biểu tượng cảm xúc và tiếng lóng, gây nhiễu cho việc phân loại. Trong khi đó, hình ảnh và chuyển động trong video trực tiếp thể hiện hành vi (bạo lực, nhạy cảm) một cách rõ ràng và ít mơ hồ hơn.

Thứ hai, về mô hình Fusion, kết quả cho thấy việc kết hợp đa phương thức giúp tăng cường "độ nhạy" của hệ thống. Độ phủ (Recall) đối với lớp độc hại tăng lên 66,7% (so với 55,6% của văn bản đơn lẻ), giúp hệ thống bỏ sót ít nội dung độc hại hơn. Tuy nhiên, chỉ số F1 tổng thể (79,7%) của Fusion vẫn thấp hơn VideoMAE đơn lẻ (87,2%). Điều này phản ánh thách thức của chiến lược Late Fusion: khi tín hiệu văn bản (chất lượng thấp) mâu thuẫn với tín hiệu video (chất lượng cao), việc gộp trung bình dự đoán có thể khiến kết quả cuối cùng bị "kéo lùi" so với nhánh tốt nhất. Mặc dù vậy, sự cải thiện về Recall vẫn mang lại giá trị thực tiễn lớn cho các bài toán yêu cầu độ an toàn cao.

5.2. Các trường hợp nghiên cứu điển hình

Phân tích định tính trên các mẫu dự đoán sai cho thấy một số pattern thú vị. Các trường hợp False Positive thường xuất hiện ở các video diễn xuất kịch bản với hành động mạnh nhưng mục đích giải trí hoặc các video giáo dục về an toàn có hình ảnh cảnh báo. Các trường hợp False Negative thường là các nội dung độc hại tinh vi sử dụng ẩn dụ, mã hóa, hoặc ngữ cảnh văn hóa đặc thù mà mô hình chưa học được. Ví dụ điển hình của False Positive là một video hướng dẫn tự phòng vệ với các động tác đấm đá được hệ thống nhận diện nhầm là bạo lực. Ví dụ điển hình của False Negative là một video quảng cáo cờ bạc nút bóng dưới dạng trò chơi giải trí với giao diện đầy màu sắc và nhạc vui nhộn.

5.3. Hạn chế và thách thức

Thứ nhất, kênh âm thanh (audio) chưa được khai thác trong hệ thống hiện tại. Điều này đặc biệt quan trọng vì nhiều video Tik Tok chứa thông tin nhạy cảm trong lời nói, nhạc nền có ca từ không phù hợp, hoặc hiệu ứng âm thanh gợi cảm. Các nghiên cứu gần đây cho thấy việc tích hợp mô hình Whisper hoặc Wav2Vec có thể cải thiện đáng kể khả năng phát hiện, đặc biệt với các video sử dụng tiếng lóng hoặc mã hóa ngôn ngữ mà văn bản mô tả không thể hiện.

Thứ hai, phương pháp lấy mẫu 16 frames đều đặn tuy hiệu quả về mặt tính toán nhưng có thể bỏ sót các hành vi độc hại diễn ra trong khoảng thời gian ngắn giữa các frames. Ví dụ, một video 15 giây với 16 frames được trích xuất sẽ có khoảng cách gần 1 giây giữa mỗi frame, đủ để bỏ sót một hành động bạo lực chớp nhoáng hoặc hình ảnh nhạy cảm xuất hiện thoáng qua. Các giải pháp tiềm năng bao gồm tăng số lượng frames hoặc sử dụng phương pháp lấy mẫu thích ứng dựa trên phát hiện chuyển động.

Thứ ba, chi phí tính toán cao của các mô hình Transformer như VideoMAE đòi hỏi GPU mạnh với bộ nhớ đồ họa tối thiểu 4GB. Điều này hạn chế khả năng triển khai trên các thiết bị edge hoặc trong môi trường có ngân sách hạn chế. Các kỹ thuật nén mô hình như quantization (INT8/INT4)

và knowledge distillation có thể giám yếu cầu phản ứng nhưng cần nghiên cứu thêm về mức độ ảnh hưởng đến độ chính xác.

Cuối cùng, các vấn đề về đạo đức AI cần được quan tâm sâu hơn. Thiên kiến trong dữ liệu gán nhãn có thể dẫn đến việc một nhóm văn hóa hoặc phong cách nội dung bị đánh giá sai lệch. Quyền riêng tư của người dùng Tik Tok cũng là mối quan ngại khi video của họ được thu thập và phân tích mà không có sự đồng ý rõ ràng. Ngoài ra, nguy cơ lạm dụng công nghệ này để kiểm duyệt quá mức hoặc nhắm mục tiêu vào các nhóm thiểu số cần được cân nhắc trong quá trình triển khai thực tế.

Chương 6: Kết luận và hướng phát triển

6.1. Tổng kết

Đội tài đã hoàn thành mục tiêu xây dựng một hệ thống kiểm duyệt nội dung độc hại trên TikTok theo thời gian thực hoạt động từ đầu đến cuối. Hệ thống tích hợp thành công kiến trúc xử lý đa tầng 8 lớp với các mô hình AI đa phương thức tiên tiến, đạt được hiệu suất phát hiện cao với chỉ số F1 đạt 87,8% sử dụng mô hình VideoMAE và thông lượng đáp ứng yêu cầu thời gian thực với 1,66 video mỗi giây. Quy trình MLOps khép kín đảm bảo khả năng cập nhật và duy trì mô hình tự động.

6.2. Hướng phát triển trong tương lai

Trong bối cảnh các hệ thống AI đa phương thức ngày càng được ứng dụng rộng rãi trong thực tế, việc đánh giá và định hướng phát triển trong tương lai là cần thiết nhằm khắc phục những hạn chế hiện tại và nâng cao tính hoàn thiện của hệ thống. Dựa trên kết quả thực nghiệm cũng như quá trình triển khai ban đầu, nhóm nhận thấy hệ thống vẫn còn tiềm năng mở rộng cả về mặt năng lực xử lý dữ liệu, hiệu quả tính toán và khả năng vận hành ở quy mô lớn.

Vì vậy, trong giai đoạn tiếp theo, nhóm đề xuất một số hướng phát triển trọng tâm nhằm cải thiện hiệu năng tổng thể, tăng tính linh hoạt khi triển khai, đồng thời hướng đến một kiến trúc hoàn chỉnh và phù hợp hơn với các kịch bản ứng dụng thực với các hướng triển khai như sau:

1. Tích hợp nhánh Audio sử dụng các mô hình như Whisper (ASR) và Wav2Vec để khai thác thông tin âm thanh, tạo thành hệ thống Tri-Modal hoàn chỉnh
2. Áp dụng các kỹ thuật tối ưu mô hình như quantization (INT8) và knowledge distillation để giảm yêu cầu phần cứng và tăng thông lượng.
3. Triển khai hệ thống lên Kubernetes với auto-scaling để xử lý lưu lượng không đồng đều trong thực tế.

Chương 7: TÀI LIỆU THAM KHẢO

- [1] Phong Nguyen-Thuan Do, undefined., et al, "VLUE: A New Benchmark and Multi-task Knowledge Transfer Learning for Vietnamese Natural Language Understanding," 2024.
- [2]. Zhan Tong, undefined., et al, "VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training," 2022.
- [3]. Gedas Bertasius, undefined. Heng Wang, undefined. Lorenzo Torresani, "Is Space-Time Attention All You Need for Video Understanding?," 2021.
- [4]. Alexis Conneau, undefined., et al, "Unsupervised Cross-lingual Representation Learning at Scale," 2020.
- [5]. Victor Sanh, undefined., et al, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," 2020.
- [6]. Kasper Cools, undefined. Gideon Mailette de Buy Wenniger, undefined. Clara Maathuis, "Modeling offensive content detection for TikTok," 2024.