Đại học Bách khoa Hà Nội
Trường Công nghệ Thông tin và Truyền thông

BÁO CÁO BÀI TẬP LỚN

IT4930 - Nhập môn Khoa học Dữ liệu

Dự án: Mô hình Phát hiện và Phân loại Cuộc gọi lừa đảo

Người hướng dẫn: PGS.TS. Phạm Văn Hải

Nhóm thực hiện:		
	Đinh Công Thái	20224898
	Nguyễn Trung Long	20224874
	Đồng Phúc Lâm	20225027
	Nguyễn Minh Đức	20224954
	Trịnh Hồ Nhật Minh	20225048

Hà Nội, tháng 12 năm 2025

Mục lục

Danh mục bảng

Danh mục hình

Chương 1

Giới thiệu bài toán

1.1 Bối cảnh và động lực

1.1.1 Sự bùng nổ của lừa đảo trực tuyến tại Việt Nam

Trong kỷ nguyên số, Việt Nam đang phải đối mặt với làn sóng tấn công lừa đảo qua viễn thông (telecom fraud) chưa từng có. Theo báo cáo từ Liên minh chống lừa đảo toàn cầu (GASA) được công bố vào cuối năm 2024, Việt Nam là một trong những quốc gia có tỷ lệ nạn nhân lừa đảo cao hàng đầu thế giới [10]. Thống kê chỉ ra một thực tế đáng báo động: trung bình cứ **220 người dùng smartphone thì có 1 người sập bẫy lừa đảo trực tuyến** [21]. Con số này cho thấy lừa đảo không còn là hiện tượng cá biệt mà đã trở thành rủi ro thường trực đối với bất kỳ ai sở hữu thiết bị di động.

1.1.2 Hạn chế của các phương pháp hiện hành và Bài toán đánh đổi hiệu năng

Sự bất lực của các phương pháp truyền thống: Các phương pháp bảo mật dựa trên định danh như Danh sách đen (Blacklist) hay Danh sách trắng (Whitelist) đã trở nên lạc hậu trước vấn nạn SIM rác và khả năng thay đổi số điện thoại liên tục của kẻ tấn công (VoIP spoofing) [27]. Song song đó, các mô hình Học máy truyền thống dựa trên đối sánh từ khóa (Keyword Matching) hay n-gram thường thất bại trong việc nắm bắt ngữ nghĩa sâu. Kẻ lừa đảo chỉ cần thay đổi cách diễn đạt hoặc sử dụng các từ lóng, ẩn dụ là có thể dễ dàng vượt qua các bộ lọc này.

Rào cản thực tế của xu hướng LLM-based: Để giải quyết bài toán ngữ nghĩa, các nghiên cứu gần đây có xu hướng tận dụng sức mạnh của các Mô hình Ngôn ngữ Lớn (Large Language Models - LLMs) hoặc các mô hình embedding dựa trên LLaMA (như RepLlama) cho bài toán phân loại văn bản. Mặc dù các hướng tiếp cận này mang lại hiệu suất rất cao, chúng gặp phải rào cản chí mạng khi triển khai thực tế: **độ trễ suy luận quá lớn** [38]. Trong ngữ cảnh ngăn chặn lừa đảo viễn thông, hệ thống cần đưa ra cảnh

báo gần như tức thời (near real-time). Việc chờ đợi các mô hình LLM đồ sộ xử lý trong vài giây đến vài chục giây là không khả thi và tốn kém tài nguyên tính toán.

Chiến lược đánh đổi: Từ những phân tích trên, nghiên cứu này đề xuất một hướng tiếp cận cân bằng: Sử dụng mô hình ngôn ngữ kích thước trung bình (như halong_embedding) kết hợp kiến trúc phân tầng (Hierarchical) [37]. Chúng tôi chấp nhận một sự đánh đổi nhỏ về mặt hiệu suất so với các siêu mô hình để đổi lấy tốc độ xử lý vượt trội và khả năng triển khai diện rộng trên hạ tầng phổ thông.

1.2 Vấn đề cần giải quyết

1.2.1 Định nghĩa bài toán

Nghiên cứu này giải quyết bài toán phân loại văn bản hội thoại đa lớp (Multi-class Dialogue Classification). Đầu vào của bài toán là một văn bản hội thoại (transcript) D, bao gồm một chuỗi các lượt lời (utterances): $D = \{u_1, u_2, ..., u_n\}$, trong đó n là số lượng lượt lời. Mỗi lượt lời u_i lại được cấu thành từ một chuỗi các từ (tokens). Mục tiêu là xây dựng một hàm ánh xạ $f : D \rightarrow y$, trong đó $y \in C$ là tập hợp các nhãn phân loại đã được định nghĩa trước. Trong phạm vi nghiên cứu này, tập nhãn C bao gồm $|C| = 26$ lớp, chứa 01 lớp "Vô hại" (Harmless) và 25 lớp hành vi lừa đảo cụ thể (ví dụ: *mạo danh công an, lừa đảo đầu tư, tuyển dụng giả mạo...*).

1.2.2 Các thách thức chính

Việc phân loại chính xác các cuộc gọi lừa đảo đối mặt với ba thách thức kỹ thuật lớn mà các mô hình phân loại văn bản thông thường khó giải quyết triệt để:

- **Độ tương đồng ngữ nghĩa cao và Giả thuyết đa tạp:** Khác với phân loại chủ đề thông thường, các kịch bản lừa đảo chia sẻ không gian từ vựng rất giống nhau (ví dụ: "Mạo danh công an"và "Mạo danh viện kiểm sát"đều dùng các từ như *truy tố, hồ sơ, phong tỏa*). Dưới góc độ của *Giả thuyết đa tạp* [4], dữ liệu văn bản của các lớp này nằm trên các đa tạp (manifolds) phi tuyến tính trong không gian cao chiều. Do sự trùng lặp ngữ nghĩa, các đa tạp của các lớp lừa đảo khác nhau có xu hướng bị "xoắn" vào nhau hoặc nằm rất sát nhau tại các vùng biên. Nếu mô hình chỉ học các ranh giới tuyến tính đơn giản mà không thực hiện được phép biến đổi topo để "tách" các đa tạp này ra, hiệu suất phân loại sẽ suy giảm đáng kể do sự nhầm lẫn giữa các lớp.

- **Phụ thuộc ngữ cảnh dài:** Dấu hiệu lừa đảo thường không nằm gọn trong một câu nói đơn lẻ mà được xây dựng qua một chuỗi hội thoại. Một câu nói như *"Vui lòng chuyển tiền vào số tài khoản này"* có thể là vô hại trong ngữ cảnh mua hàng

online, nhưng sẽ là lừa đảo trong ngữ cảnh mạo danh người thân. Mô hình cần có khả năng ghi nhớ và liên kết thông tin từ đầu đến cuối cuộc hội thoại thay vì chỉ xét các câu riêng lẻ [37].

- **Sự mất cân bằng dữ liệu cực đoan:** Trong thực tế, tỷ lệ cuộc gọi lừa đảo thấp hơn rất nhiều so với cuộc gọi bình thường. Ngay cả trong tập các cuộc gọi lừa đảo, cũng có những kịch bản phổ biến (như lừa đảo khóa SIM) và những kịch bản rất hiếm gặp. Nếu không có chiến lược xử lý phù hợp [5], mô hình sẽ có xu hướng học theo lớp đa số và bỏ qua các hành vi lừa đảo tinh vi nhưng ít xuất hiện.

1.3 Tầm quan trọng và giá trị thực tế

Nghiên cứu này không chỉ mang ý nghĩa lý thuyết trong việc ứng dụng các mô hình ngôn ngữ tiên tiến vào bài toán thực tế, mà còn đóng góp giá trị to lớn trên ba khía cạnh chính:

- **Bảo vệ người dùng cuối:** Hệ thống đóng vai trò như một lớp tường lửa thông minh tích hợp trên thiết bị hoặc hạ tầng mạng. Khả năng phát hiện thời gian thực cho phép đưa ra cảnh báo ngay khi cuộc gọi có dấu hiệu lừa đảo, giúp ngăn chặn hành vi chuyển tiền hoặc cung cấp thông tin nhạy cảm trước khi hậu quả xảy ra. Đây là giải pháp phòng ngừa chủ động thay vì thụ động giải quyết hậu quả.
- **Hỗ trợ quản lý và An ninh mạng:** Thay vì chỉ đưa ra nhãn nhị phân (Có/Không), mô hình cung cấp khả năng phân loại "mịn"(Fine-grained classification) vào 25 loại hình lừa đảo cụ thể. Điều này giúp các cơ quan chức năng và nhà mạng:
  - Xây dựng bản đồ tội phạm: Thống kê loại hình lừa đảo nào đang bùng phát tại thời điểm nào.
  - Truy vết nguồn gốc: Nhận diện các nhóm tội phạm dựa trên sự tương đồng về kịch bản.
  - Cập nhật chính sách: Đưa ra các cảnh báo cộng đồng chính xác hơn dựa trên dữ liệu thực tế.
- **Tối ưu hóa vận hành và Khả năng mở rộng:** Với hàng triệu cuộc gọi phát sinh mỗi ngày, việc giám sát thủ công là bất khả thi. Mô hình đề xuất, với kiến trúc tối ưu hóa về độ trễ, cho phép tự động hóa quy trình rà soát với chi phí thấp. Nó giải phóng nguồn lực con người khỏi công việc gán nhãn thủ công nhàm chán, cho phép tập trung vào các trường hợp phức tạp hoặc các kịch bản lừa đảo mới chưa từng xuất hiện.

1.4 Phạm vi và giới hạn

Để đảm bảo tính khả thi và tập trung vào mục tiêu nghiên cứu chính, đề tài được giới hạn trong các phạm vi và điều kiện biên sau:

- **Phạm vi dữ liệu và Ngôn ngữ:** Nghiên cứu tập trung duy nhất vào xử lý dữ liệu Tiếng Việt. Đầu vào của mô hình là văn bản giả định đã được chuyển đổi từ giọng nói thông qua các hệ thống ASR (Automatic Speech Recognition) thương mại. Bài toán xử lý tiếng nói nằm ngoài phạm vi của nghiên cứu này.
- **Vấn đề lan truyền lỗi:** Do mô hình hoạt động trên văn bản đầu ra của ASR, độ chính xác của hệ thống phụ thuộc tuyến tính vào chất lượng của bộ ASR đó. Các lỗi nhận dạng (như sai tên riêng, từ đồng âm khác nghĩa) có thể ảnh hưởng đến kết quả phân loại [23]. Tuy nhiên, mô hình ngôn ngữ lớn được kỳ vọng có khả năng tự sửa lỗi ngữ cảnh ở mức độ nhất định.
- **Thiếu hụt thông tin phi ngôn ngữ:** Vì chỉ xử lý văn bản, mô hình hiện tại bỏ qua các đặc trưng âm thanh quan trọng như giọng điệu (tone), ngữ điệu (prosody), khoảng lặng (pause) hay cảm xúc người nói (emotion). Một số kịch bản lừa đảo tinh vi dựa trên việc thao túng cảm xúc (như giả giọng khóc lóc, quát tháo) có thể bị bỏ sót nếu văn bản thuần túy không thể hiện được các sắc thái này.
- **Giới hạn cửa sổ ngữ cảnh:** Để đảm bảo tốc độ suy luận nhanh, mô hình giới hạn độ dài hội thoại đầu vào (ví dụ: 12 lượt lời đầu tiên). Điều này đồng nghĩa với việc nếu dấu hiệu lừa đảo chỉ xuất hiện ở cuối một cuộc hội thoại dài, mô hình có thể không phát hiện được. Đây là sự đánh đổi chấp nhận được giữa Hiệu năng và Tốc độ.

1.5 Cơ sở công nghệ và Lựa chọn hướng tiếp cận

1.5.1 Sự phổ biến của công nghệ nhận dạng giọng nói (ASR)

Hiện nay, công nghệ nhận dạng giọng nói tự động đã đạt được độ chính xác rất cao và trở thành một tính năng tiêu chuẩn trên hầu hết các thiết bị di động thông minh. Các nền tảng hệ điều hành (như iOS, Android) và các dịch vụ đám mây (Google Cloud Speech-to-Text, Viettel AI, FPT.AI) đều cung cấp API chuyển đổi giọng nói thành văn bản với độ trễ thấp và khả năng xử lý tiếng Việt tốt. Điều này tạo ra tiền đề công nghệ vững chắc, cho phép hệ thống phân loại lừa đảo có thể tin cậy vào đầu vào là dữ liệu văn bản chất lượng cao thay vì phải xử lý trực tiếp tín hiệu âm thanh thô.

1.5.2 Ưu thế của mô hình ngôn ngữ (Hierarchical BERT) so với mô hình âm thanh (CNN-based)

Trong bài toán phát hiện lừa đảo, "kịch bản" và "ngữ nghĩa" đóng vai trò quan trọng hơn "giọng điệu". Do đó, việc lựa chọn hướng tiếp cận dựa trên văn bản (Text-based) sử dụng kiến trúc Hierarchical BERT mang lại nhiều ưu điểm vượt trội so với các mô hình xử lý tín hiệu âm thanh (Audio-based) như CNN hay RNN truyền thống:

- **Mật độ thông tin ngữ nghĩa:** Các mô hình CNN trên miền âm thanh thường tập trung vào trích xuất đặc trưng cục bộ và dễ bị ảnh hưởng bởi nhiễu môi trường (noise). Ngược lại, BERT (và biến thể halong_embedding) [22] đã được huấn luyện trên lượng dữ liệu văn bản khổng lồ, có khả năng thấu hiểu ngữ cảnh, ý định và mối quan hệ giữa các từ ngữ phức tạp trong tiếng Việt tốt hơn nhiều.

- **Cấu trúc phân cấp:** Một cuộc hội thoại lừa đảo thường kéo dài qua nhiều lượt lời. Mô hình CNN truyền thống thường gặp khó khăn trong việc nắm bắt sự phụ thuộc xa. Kiến trúc Hierarchical (như mô hình đề xuất) [37] mô phỏng chính xác cấu trúc tự nhiên của hội thoại: Từ (Word) tạo thành Câu (Utterance), và các Câu tạo thành Hội thoại (Dialogue). Điều này giúp tối ưu hóa việc học các mẫu hành vi lừa đảo ẩn sâu trong luồng đối thoại.

- **Hiệu năng tính toán:** Việc xử lý trực tiếp tín hiệu âm thanh đòi hỏi băng thông và năng lực tính toán lớn. Bằng cách tận dụng các module ASR có sẵn trên thiết bị hoặc server, mô hình phân loại chỉ cần xử lý văn bản, giúp giảm tải đáng kể tài nguyên hệ thống và tăng tốc độ phản hồi.

Chương 2

Dữ liệu và Quy trình Xử lý

2.1 Nguồn dữ liệu và Định nghĩa nhãn

Để đảm bảo tính đa dạng và sát với thực tế triển khai, tập dữ liệu trong nghiên cứu được xây dựng theo phương pháp lai, kết hợp giữa dữ liệu thực tế và dữ liệu tổng hợp. Hệ thống nhãn và kịch bản được tham chiếu trực tiếp từ tài liệu nghiệp vụ của Tập đoàn Công nghiệp - Viễn thông Quân đội Viettel.

2.1.1 Cấu trúc nhãn

Không gian nhãn của bài toán bao gồm tổng cộng **26 lớp**, được định nghĩa dựa trên danh mục các mối đe dọa viễn thông hiện hành:

- **01 Nhãn Harmless (Vô hại):** Bao gồm các cuộc gọi sinh hoạt đời thường, giao vận, đặt lịch hẹn, trao đổi công việc không chứa yếu tố lừa đảo.
- **25 Nhãn Scam (Lừa đảo):** Bao gồm 46 kịch bản lừa đảo chi tiết được cung cấp bởi Viettel và chuẩn hóa lại thành 25 lớp để tránh hỗn tạp ngữ nghĩa (ví dụ: *Lừa đảo khóa SIM sau 2 giờ, Mạo danh Cục Cảnh sát giao thông phạt nguội, Giả mạo nhân viên sàn thương mại điện tử tuyển cộng tác viên...*).

2.1.2 Thành phần dữ liệu

Tổng số mẫu dữ liệu thu thập được là **18,686 đoạn hội thoại**, được tổng hợp từ ba nguồn chính:

1. **Dữ liệu lừa đảo thực tế:** Khoảng 3,000 mẫu dữ liệu nội bộ được cung cấp bởi Tập đoàn Viettel. Đây là các transcript từ các cuộc gọi lừa đảo đã được báo cáo và xác minh, mang tính đại diện cao cho các kịch bản tấn công thực tế.
2. **Dữ liệu vô hại thực tế:** Khoảng 5,000 mẫu transcript được khai thác từ kho dữ liệu mở của Trung tâm Dữ liệu Quốc gia và các nguồn dữ liệu hội thoại tiếng Việt

công khai (như Vivos [18], VinBigData), đảm bảo tính tự nhiên của ngôn ngữ đời sống.

3. **Dữ liệu tổng hợp:** Phần còn lại (khoảng 10,000 mẫu) được sinh ra bằng cách sử dụng các Mô hình Ngôn ngữ Lớn (LLMs) [9]. Quy trình sinh dữ liệu tuân thủ nghiêm ngặt các từ khóa (keywords) và logic kịch bản trong file mô tả nghiệp vụ của Viettel, nhằm làm giàu dữ liệu cho các lớp hiếm gặp.

2.2 Phân tích đặc điểm dữ liệu (EDA)

Dữ liệu thể hiện sự mất cân bằng nghiêm trọng, phản ánh đúng bản chất phân phối của các cuộc gọi trên mạng viễn thông:

- **Lớp Harmless:** Chiếm đa số áp đảo với 14,111 mẫu (tương đương 75.5% tổng dữ liệu).
- **Các lớp Scam:** Tổng cộng 4,575 mẫu, phân bố rải rác trong 25 loại hình.
- **Phân phối nội bộ các lớp Scam:** Số lượng mẫu không đồng đều, dao động từ thấp nhất 68 mẫu (*lừa đảo du lịch*) đến cao nhất 162 mẫu (*lừa đảo chuyển tiền*). Trung bình mỗi lớp scam chỉ có khoảng 100 mẫu, đặt ra thách thức lớn về việc học đặc trưng (feature learning).
- **Đặc trưng độ dài:** Trung bình mỗi cuộc hội thoại kéo dài 9.18 lượt lời (turns). Độ dài tối đa sau khi làm sạch là 12 turns, tập trung vào giai đoạn "mở bài" và "dẫn dắt" của kẻ lừa đảo.

2.3 Phân chia tập dữ liệu

Trước khi thực hiện bất kỳ kỹ thuật tăng cường nào, dữ liệu gốc được chia tách để đảm bảo tính khách quan của quá trình đánh giá. Chúng tôi sử dụng chiến lược *Stratified Split* để bảo toàn tỷ lệ phân phối của các lớp (đặc biệt là các lớp hiếm) trên cả 3 tập. Tỷ lệ phân chia là 80/10/10:

- **Tập Huấn luyện (Train):** 14,948 mẫu và phần lớn là dữ liệu được sinh ra bằng LLMs. Chỉ tập này mới được áp dụng các kỹ thuật tăng cường dữ liệu.
- **Tập Kiểm định (Validation):** 1,869 mẫu. Dùng để tinh chỉnh siêu tham số và đánh giá sớm.
- **Tập Kiểm thử (Test):** 1,869 mẫu. Dữ liệu hoàn toàn nguyên bản là dữ liệu thật, dùng để đánh giá hiệu năng cuối cùng.

2.4 Chiến lược Tăng cường dữ liệu

Để giải quyết vấn đề thiếu hụt dữ liệu ở các lớp Scam và tăng cường khả năng chống chịu lỗi của mô hình trước các sai sót của hệ thống nhận dạng giọng nói, chúng tôi áp dụng quy trình Tăng cường dữ liệu **chỉ trên tập Train** với các kỹ thuật sau:

2.4.1 Mô phỏng lỗi nhận dạng

Sử dụng thư viện 'nlpaug' [19] để mô phỏng các lỗi thường gặp khi chuyển đổi Speech-to-Text:

- **Thay thế từ đồng âm:** Thay thế các từ bằng từ khác có âm đọc giống hoặc gần giống (ví dụ: "chuyển khoản" → "chiển khoản", "ngân hàng" → "ngân hằng"). Kỹ thuật này giúp mô hình học được ngữ nghĩa dựa trên ngữ cảnh thay vì bắt từ khóa cứng nhắc.
- **Mô phỏng nhiễu bàn phím:** Thay thế ký tự dựa trên khoảng cách phím QWERTY, mô phỏng lỗi gõ sai trong quá trình chuẩn hóa văn bản thủ công (nếu có).

2.4.2 Dịch ngược

Áp dụng cho các mẫu thuộc các lớp Scam hiếm. Văn bản Tiếng Việt được dịch sang Tiếng Anh (sử dụng mô hình dịch máy) và dịch ngược lại Tiếng Việt [30]. Phương pháp này giúp tạo ra các biến thể câu mới với cấu trúc ngữ pháp khác biệt nhưng vẫn giữ nguyên ý nghĩa gốc, giúp mô hình tránh bị quá khớp vào các mẫu câu cố định.

2.4.3 Cắt ngẫu nhiên

Mô phỏng hiện tượng mất tín hiệu hoặc ngắt quãng trong cuộc gọi VoIP. Chúng tôi xóa ngẫu nhiên một số từ hoặc một lượt lời ngắn trong hội thoại với xác suất thấp ($p = 0.1$) [35]. Điều này buộc mô hình phải học cách suy luận dựa trên các thông tin còn sót lại.

2.5 Chuẩn hóa cấu trúc và Phân đoạn hội thoại

Do dữ liệu đầu vào được tổng hợp từ nhiều nguồn không đồng nhất (dữ liệu nghiệp vụ, dữ liệu mở, và dữ liệu sinh từ LLM), bước đầu tiên trong quy trình xử lý là xây dựng cơ chế ánh xạ để đưa toàn bộ dữ liệu thô về một định dạng chuẩn, phục vụ cho kiến trúc phân tầng của mô hình.

2.5.1 Đồng nhất hóa Khuôn dạng

Mọi mẫu dữ liệu thô, bất kể định dạng gốc (Excel, CSV, hay JSON phi cấu trúc), đều được chuyển đổi về một schema JSON tiêu chuẩn duy nhất. Cấu trúc của một mẫu dữ liệu D sau khi chuẩn hóa được định nghĩa tường minh:

$$D = \{\textbf{id} : \text{String},\ \textbf{label} : \text{Int} \in [0, 25],\ \textbf{dialogue} : [u_1, u_2, ..., u_n]\} \tag{2.1}$$

Trong đó, trường `dialogue` là một danh sách chứa chuỗi các lượt lời, đảm bảo tính nhất quán về kiểu dữ liệu đầu vào cho các bước xử lý tiếp theo.

2.5.2 Phân đoạn lượt lời

Kiến trúc Hierarchical BERT yêu cầu đầu vào phải được tách biệt rõ ràng giữa các câu thoại để học được mối quan hệ ngữ cảnh.

- **Cơ chế tách:** Đối với các transcript dạng văn bản liền mạch (raw text), chúng tôi áp dụng các quy tắc Heuristic kết hợp biểu thức chính quy (Regex) để nhận diện điểm chuyển giao người nói (Speaker Diarization tags) hoặc dấu ngắt câu đặc thù.
- **Mục tiêu:** Biến đổi văn bản thô thành chuỗi các lượt lời $(u_1, u_2, ...)$ tương ứng với các bước trong kịch bản (ví dụ: u_1: Chào hỏi $\rightarrow$ u_2: Dẫn dắt $\rightarrow$ u_3: Đe dọa). Việc phân đoạn chính xác giúp mô hình nắm bắt được "nhịp điệu" và logic tấn công của kẻ lừa đảo thay vì chỉ xử lý một khối văn bản hỗn độn.

2.6 Tiền xử lý dữ liệu

Sau khi tăng cường, dữ liệu đi qua luồng xử lý cuối cùng trước khi đưa vào mô hình:

1. **Tokenization:** Sử dụng 'AutoTokenizer' [36] từ mô hình 'hiieu/halong_embedding-base' để tách từ và mã hóa văn bản, tận dụng khả năng xử lý tiếng Việt ưu việt của halong_embedding.
2. **Cắt gọt và Định dạng (Truncation & Padding):**
   - Giới hạn số lượng lượt lời tối đa (MAX_TURNS) là 12.
   - Giới hạn độ dài token tối đa cho mỗi lượt lời (TOKENIZER_MAX_LEN) là 128 tokens.
3. **Cân bằng dữ liệu:** Sau bước Augmentation, kỹ thuật **Oversampling** tiếp tục được áp dụng trên tập Train. Các lớp Scam vẫn còn ít mẫu hơn mức trung bình sẽ được nhân bản ngẫu nhiên để đạt ngưỡng cân bằng (Target Oversample = 1.5×

trung bình số mẫu scam), nâng tổng số mẫu huấn luyện thực tế lên 16,777 mẫu. Điều này triệt tiêu thiên kiến của mô hình đối với lớp đa số (Harmless).

Chương 3

Phương pháp và Kiến trúc Mô hình

Chương này trình bày chi tiết kiến trúc mạng nơ-ron phân tầng (Hierarchical Neural Network) được đề xuất để giải quyết bài toán phân loại cuộc gọi lừa đảo. Thay vì sử dụng các phương pháp tinh chỉnh đơn thuần, chúng tôi thiết kế một kiến trúc chuyên biệt kết hợp giữa khả năng hiểu ngữ nghĩa của Mô hình Ngôn ngữ Lớn và không gian embedding được tối ưu hóa thông qua học tương phản.

3.1 Lựa chọn Mô hình cơ sở

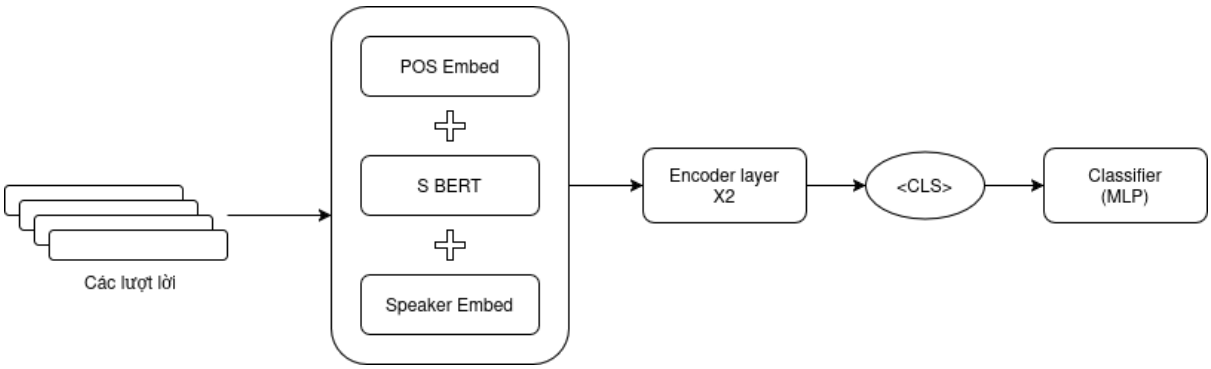
Với đặc thù bài toán yêu cầu biểu diễn ngữ nghĩa ở cấp độ câu cho tiếng Việt, nghiên cứu lựa chọn **halong_embedding** làm mô hình nền cho tầng mã hóa câu (Sentence Encoder).

- **Lý do lựa chọn:** halong_embedding là mô hình sinh embedding ở cấp độ câu, được xây dựng theo hướng *sentence-level representation learning*, thay vì chỉ học biểu diễn ngữ cảnh ở cấp độ token. Mô hình được huấn luyện trên tập dữ liệu tiếng Việt quy mô lớn (khoảng 20GB văn bản), với kiến trúc Transformer encoder làm backbone [34] và được tối ưu hóa để các câu có ngữ nghĩa tương đồng được ánh xạ gần nhau trong không gian vector. Nhờ đó, halong_embedding đặc biệt phù hợp cho các tác vụ như phân loại văn bản, truy hồi ngữ nghĩa và phát hiện nội dung tương đồng trong tiếng Việt.

- **Chiến lược tinh chỉnh hiệu quả (PEFT):** Thay vì tinh chỉnh toàn bộ mô hình với hàng trăm triệu tham số, nghiên cứu áp dụng phương pháp **LoRA (Low-Rank Adaptation)** [11] nhằm giảm chi phí huấn luyện và hạn chế hiện tượng *catastrophic forgetting*. LoRA chèn các ma trận hạng thấp vào các phép biến đổi tuyến tính trong khối Attention (đặc biệt là các nhánh *Query* và *Value*), cho phép mô hình thích nghi với miền dữ liệu lừa đảo trong khi chỉ cần cập nhật một tỷ lệ rất nhỏ tham số.

3.2 Kiến trúc tổng quan

Mô hình được xây dựng theo kiến trúc phân tầng (Hierarchical Architecture) để mô phỏng cấu trúc tự nhiên của hội thoại: *Từ tạo thành Câu, Câu tạo thành Hội thoại.*



Hình 3.1: Kiến trúc tổng quan của mô hình

3.2.1 Tầng mã hóa câu (Utterance Encoder)

Đầu vào là một lượt lời u_i gồm chuỗi các token. halong_embedding đóng vai trò trích xuất đặc trưng ngữ nghĩa cục bộ:

$$h_i = \text{halong\_embedding}(u_i) \in \mathbb{R}^{d_{model}} \tag{3.1}$$

Trong đó h_i là vector đại diện cho lượt lời thứ i, được lấy từ output của model.

3.2.2 Tầng mã hóa hội thoại (Dialogue Encoder)

Để nắm bắt sự phụ thuộc thời gian và ngữ cảnh giữa các lượt lời trong chuỗi hội thoại $D = \{h_1, h_2, ..., h_T\}$, chúng tôi sử dụng khối **Transformer Encoder** tiêu chuẩn [34]. Khối này giúp mô hình hiểu được dòng chảy của cuộc gọi (ví dụ: lời chào $\rightarrow$ dẫn dắt $\rightarrow$ đe dọa) thông qua cơ chế Self-Attention đa đầu (Multi-head Self-Attention).

3.2.3 Cơ chế Attention Pooling (Thay thế Random Init [CLS])

Đây là cải tiến quan trọng so với các phương pháp thông thường. Trong các mô hình BERT tiêu chuẩn, vector đại diện cho cả văn bản thường là token '[CLS]' (được khởi tạo ngẫu nhiên và học từ đầu) hoặc Max/Mean Pooling. Tuy nhiên, trong một cuộc gọi lừa đảo, không phải lượt lời nào cũng quan trọng như nhau (ví dụ: câu "Alo" ít thông tin hơn câu "Chuyển tiền ngay").

Chúng tôi đề xuất sử dụng cơ chế **Attention Pooling** để học trọng số tầm quan trọng cho từng lượt lời, thay vì gán trọng số bằng nhau (Mean Pooling) hay chỉ lấy

vector cuối cùng. Giả sử $H = \{h'_1, h'_2, ..., h'_T\}$ là đầu ra của Dialogue Encoder, vector đại diện hội thoại v được tính như sau:

$$a_i = \tanh(W_a h'_i + b_a) \tag{3.2}$$

$$\alpha_i = \frac{\exp(a_i^T u_a)}{\sum_{j=1}^{T} \exp(a_j^T u_a)} \tag{3.3}$$

$$v = \sum_{i=1}^{T} \alpha_i h'_i \tag{3.4}$$

Trong đó, u_a là vector ngữ cảnh (context vector) được học trong quá trình huấn luyện. Phương pháp này tận dụng tối đa tri thức từ các tầng trước đó và cho phép mô hình "tập trung" vào các câu mang tính chất quyết định hành vi lừa đảo.

3.3 Chiến lược Huấn luyện: Centroid-based Contrastive Learning

Một thách thức lớn của bài toán phân loại 26 lớp với dữ liệu mất cân bằng là ranh giới giữa các lớp thường rất mờ nhạt. Để giải quyết vấn đề này, chúng tôi kết hợp hàm mất mát phân loại (Cross Entropy) với học tương phản (Contrastive Learning).

Tuy nhiên, các phương pháp Contrastive Learning truyền thống (như SimCLR [7], Triplet Loss thông thường) thường yêu cầu *Batch Size* rất lớn để tìm được cặp mẫu âm tính đủ khó. Để khắc phục hạn chế về phần cứng (VRAM) và ổn định quá trình huấn luyện, nghiên cứu sử dụng phương pháp **Centroid-based Contrastive Learning** (lấy cảm hứng từ Prototypical Networks [32]).

3.3.1 Quản lý Centroid (Centroid Manager)

Thay vì so sánh các mẫu dữ liệu với nhau (sample-to-sample), chúng tôi so sánh mẫu dữ liệu với tâm cụm (centroid) của các lớp (sample-to-class).

- Mỗi lớp $c \in C$ được đại diện bởi một vector centroid μ_c.
- Centroid không phải là tham số cố định mà được cập nhật liên tục theo phương pháp trung bình di động mũ (Exponential Moving Average - EMA) trong quá trình huấn luyện:

$$\mu_c^{(t)} = \beta \mu_c^{(t-1)} + (1 - \beta) \bar{v}_c^{(t)} \tag{3.5}$$

  Trong đó $\bar{v}_c^{(t)}$ là trung bình các vector đặc trưng của lớp c trong batch hiện tại.

Ưu điểm: Phương pháp này giúp hiệu suất mô hình **không phụ thuộc vào kích thước Batch Size**, đồng thời giúp biểu diễn của các lớp ổn định hơn (stable representation).

3.3.2 Khai thác mẫu khó đa cấp độ (Multi-level Hard Mining)

Hệ thống sử dụng cơ chế "đào" mẫu khó (Hard Mining) ở 3 cấp độ để tối ưu hóa không gian vector:

- **Level 1 - Phân tách nhị phân (Global Separation):** Đẩy xa tất cả các mẫu thuộc lớp Scam ra khỏi cụm Harmless.
- **Level 2 - Phân tách liên lớp (Inter-class Separation):** Với một mẫu Scam loại A, mô hình kéo nó về gần centroid A (Positive) và đẩy xa khỏi centroid của lớp Scam B bất kỳ (Random Negative).
- **Level 3 - Tinh chỉnh cục bộ (Fine-grained Separation):** Đây là cấp độ khó nhất. Với mẫu Scam loại A, mô hình tìm centroid của lớp Scam "giống nó nhất" (Hardest Negative - ví dụ lớp B có vector gần A nhất) để đẩy ra. Điều này buộc mô hình phải học được các đặc trưng cực kỳ chi tiết để phân biệt các kịch bản lừa đảo na ná nhau.

3.4 Hàm mất mát và Tối ưu hóa

Hàm mất mát tổng thể là sự kết hợp có trọng số giữa Loss phân loại và Loss tương phản:

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{Contrastive} \tag{3.6}$$

3.4.1 Focal Loss

Thay vì sử dụng Cross Entropy tiêu chuẩn, chúng tôi sử dụng **Focal Loss** [16] để tập trung vào các mẫu khó phân loại (hard examples) và giảm trọng số của các mẫu dễ (như lớp Harmless chiếm đa số).

$$FL(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{3.7}$$

Với $\gamma = 2.5$ giúp mô hình phạt nặng hơn các dự đoán sai trên các lớp hiếm.

3.4.2 Triplet Margin Loss với Centroid

Hàm mất mát tương phản được định nghĩa dựa trên khoảng cách Euclidean (theo cơ chế Triplet Loss [28]):

$$\mathcal{L}_{triplet}(a, p, n) = \max(0, \|f(a) - f(p)\|_2^2 - \|f(a) - f(n)\|_2^2 + m) \tag{3.8}$$

Trong đó:

- a: Anchor (Mẫu đầu vào).
- p: Positive (Centroid của đúng lớp đó).
- n: Negative (Centroid của lớp khác, được chọn theo chiến lược Hard Mining).
- m: Margin (Biên độ), được thiết lập giảm dần theo các Level (Lv1 > Lv2 > Lv3) để siết chặt không gian biểu diễn.

3.4.3 Chiến lược trọng số động (Dynamic Loss Weighting)

Một thách thức lớn khi kết hợp Cross-Entropy Loss ($\mathcal{L}_{CE}$) và Contrastive Loss ($\mathcal{L}_{Ct}$) là sự chênh lệch về độ lớn gradient. Trong giai đoạn đầu, khi các vector embedding chưa mang nhiều ý nghĩa ngữ nghĩa, việc ép buộc Contrastive Loss quá lớn có thể khiến mô hình khó hội tụ.

Chúng tôi đề xuất cơ chế cập nhật trọng số động cho thành phần Contrastive Loss theo lịch trình tuyến tính (Linear Schedule). Trọng số λ_{ct} tại epoch thứ t được tính như sau:

$$\lambda_{ct}^{(t)} = \lambda_{min} + (\lambda_{max} - \lambda_{min}) \times \frac{t}{T_{total}} \tag{3.9}$$

Trong thực nghiệm: $\lambda_{min} = 0.1$, $\lambda_{max} = 1.0$.

- **Giai đoạn đầu (Epoch 0-5):** λ_{ct} nhỏ, mô hình tập trung học phân loại cơ bản bằng $\mathcal{L}_{CE}$.
- **Giai đoạn sau:** λ_{ct} tăng dần, ép buộc mô hình tinh chỉnh không gian vector để tách biệt các lớp khó.

Cơ chế này giúp ổn định quá trình huấn luyện và tránh hiện tượng "bùng nổ"loss (loss explosion) ở những epoch đầu tiên.

3.5 Chiến lược Huấn luyện và Tối ưu hóa Tài nguyên

Việc huấn luyện các mô hình dựa trên Transformer (như halong_embedding) kết hợp với các cơ chế phức tạp (Contrastive Learning, Attention Pooling) thường đòi hỏi tài nguyên tính toán khổng lồ. Để triển khai huấn luyện hiệu quả trên hạ tầng phần cứng giới hạn (Single GPU với VRAM hạn chế) mà vẫn đảm bảo sự hội tụ tốt nhất, chúng tôi áp dụng các kỹ thuật tối ưu hóa bộ nhớ và chiến lược tinh chỉnh tham số chuyên sâu.

3.5.1 Kỹ thuật Tối ưu hóa Bộ nhớ

Để giải quyết bài toán giới hạn bộ nhớ GPU (Out-Of-Memory - OOM) khi huấn luyện mô hình phân tầng với ngữ cảnh dài, nghiên cứu áp dụng đồng thời ba kỹ thuật tối ưu hóa tiên tiến:

Gradient Checkpointing

Mặc định, quá trình lan truyền xuôi sẽ lưu trữ toàn bộ các activations trung gian để phục vụ cho việc tính đạo hàm trong lan truyền ngược. Với mô hình Hierarchical, số lượng activations này tăng tuyến tính theo số lượng lượt lời, gây bùng nổ bộ nhớ. Chúng tôi sử dụng kỹ thuật **Gradient Checkpointing** [6]. Thay vì lưu tất cả, hệ thống chỉ lưu các activations tại một số nút chiến lược. Các giá trị trung gian sẽ được tính toán lại trong quá trình backward.

- **Hiệu quả:** Giảm mức tiêu thụ VRAM xuống khoảng 40-50%, cho phép tăng kích thước Batch Size hoặc độ dài ngữ cảnh đầu vào.
- **Đánh đổi:** Tăng thời gian huấn luyện lên khoảng 20-30% do chi phí tính toán lại, nhưng đây là sự đánh đổi cần thiết để mô hình có thể khởi chạy được.

Mixed Precision Training

Sử dụng thư viện 'torch.amp' (Automatic Mixed Precision) [20] để thực hiện các phép tính nhân ma trận dưới định dạng dấu phẩy động 16-bit (FP16) trong khi vẫn giữ trọng số chính ở định dạng 32-bit (FP32) để đảm bảo độ ổn định số học. Kỹ thuật này giúp giảm một nửa dung lượng bộ nhớ cần thiết cho việc lưu trữ trọng số và gradients, đồng thời tận dụng các nhân Tensor Core trên GPU hiện đại để tăng tốc độ tính toán.

Gradient Accumulation

Để mô phỏng một kích thước batch lớn (Large Batch Size) - yếu tố quan trọng giúp ổn định cập nhật Centroid và Batch Normalization - trong khi bộ nhớ vật lý chỉ chịu tải được 'Micro-Batch' nhỏ, chúng tôi sử dụng **Gradient Accumulation**. Gradient được tính toán và tích lũy qua nhiều bước trước khi thực hiện một lần cập nhật trọng số.

$$\text{Effective Batch Size} = \text{Micro Batch Size} \times \text{Accumulation Steps} \tag{3.10}$$

Trong thực nghiệm, Micro Batch được thiết lập là 16 và tích lũy qua 2 bước để đạt Effective Batch Size là 32, đảm bảo sự ổn định thống kê cho hàm Loss.

3.5.2 Chiến lược Huấn luyện

Cơ chế đông cứng phân tầng

Trong bài toán phát hiện lừa đảo, dữ liệu huấn luyện thường có mức độ nhiễu cao và phân bố khác biệt đáng kể so với tập dữ liệu tổng quát dùng để tiền huấn luyện `halong_embedding`. Nếu tinh chỉnh toàn bộ mô hình ngay từ đầu, các gradient lớn sinh ra từ dữ liệu đặc thù này có thể làm suy giảm hoặc phá hủy các biểu diễn ngôn ngữ phổ quát đã được học trước đó, dẫn đến hiện tượng *Catastrophic Forgetting*.

Để giảm thiểu rủi ro này và đảm bảo quá trình thích nghi diễn ra ổn định, chúng tôi áp dụng chiến lược **đông cứng phân tầng (layer-wise freezing)** với cơ chế mở khóa dần dần:

- **Giai đoạn Warm-up (Epoch 0–3):** Toàn bộ các tham số của backbone model `halong_embedding` được "đông cứng" (freeze), chỉ cho phép cập nhật các thành phần được khởi tạo mới (Attention Pooling, Classifier Head) cùng với các ma trận thích nghi LoRA. Cách tiếp cận này giúp mô hình: (i) học cách ánh xạ các biểu diễn ngôn ngữ sẵn có sang không gian nhãn lừa đảo, (ii) ổn định các trọng số mới trước khi cho phép tác động ngược trở lại lên backbone, từ đó tránh việc các gradient nhiễu làm biến dạng các đặc trưng ngôn ngữ nền tảng.
- **Giai đoạn Fine-tuning (Epoch 4 trở đi):** Sau khi các tầng trên đã hội tụ tương đối, mô hình được mở khóa (unfreeze) để cho phép tinh chỉnh sâu hơn. Việc này giúp backbone điều chỉnh các biểu diễn ở mức tinh vi nhằm phù hợp hơn với miền dữ liệu lừa đảo, trong khi vẫn bảo toàn được các tri thức ngôn ngữ tổng quát đã học.

Lịch trình tốc độ học (Learning Rate Scheduler)

Chúng tôi sử dụng bộ điều lịch **Cosine Annealing with Warmup** [17]. Trong giai đoạn đầu, tốc độ học (Learning Rate – LR) được tăng tuyến tính từ 0 đến giá trị cực đại nhằm tránh các bước cập nhật đột ngột khi mô hình chưa ổn định. Sau đó, LR giảm dần theo hàm Cosine về gần 0, giúp mô hình tinh chỉnh các tham số một cách mượt mà và hội tụ ổn định ở giai đoạn cuối.

3.5.3 Cấu hình Siêu tham số

Các tham số tối ưu được tìm ra sau quá trình thực nghiệm (Grid Search) trên tập Validation được trình bày trong Bảng dưới đây:

Bảng 3.1: Bảng cấu hình siêu tham số huấn luyện

Tham số	**Giá trị**	**Giải thích**
Backbone Model	halong_embedding-base	Pre-trained model nền tảng.
Max Sequence Length	128	Độ dài tối đa của một lượt lời (tokens).
Max Turns	12	Số lượt lời tối đa trong một hội thoại.
Micro Batch Size	16	Kích thước batch thực tế nạp vào GPU.
Accumulation Steps	2	Số bước tích lũy gradient.
Effective Batch Size	32	Kích thước batch tổng thể để cập nhật trọng số.
Learning Rate	1e-3 (Head) 2e-4 (Backbone)	Tốc độ học khác nhau cho các tầng khác nhau (Differential Learning Rate).
Optimizer	AdamW [17]	Tối ưu hóa với Weight Decay = 1e-4.
Epochs	30	Tổng số chu kỳ huấn luyện.
Patience	5	Số epoch dừng sớm nếu Val Loss không giảm.
Contrastive Margins	$m_1 = 0.3, m_2 = 0.35, m_3 = 0.35$	Biên độ Loss tương phản giảm dần theo độ khó (Global $\rightarrow$ Local).
Focal Gamma (γ)	2.5	Hệ số tập trung vào các mẫu khó phân loại.

Chương 4

Thử nghiệm và Đánh giá

Chương này trình bày chi tiết về môi trường thực nghiệm, các tiêu chí đánh giá và kết quả so sánh giữa các phiên bản mô hình. Mục tiêu là chứng minh hiệu quả của kiến trúc phân tầng đề xuất và các kỹ thuật huấn luyện nâng cao (Contrastive Learning, Focal Loss) thông qua các số liệu định lượng và định tính.

4.1 Thiết lập thí nghiệm

4.1.1 Môi trường phần cứng và phần mềm

Các thí nghiệm được thực hiện trên nền tảng tính toán đám mây với cấu hình phần cứng bao gồm GPU NVIDIA Tesla T4 (16GB VRAM) và RAM hệ thống 32GB. Về phần mềm, mô hình được cài đặt bằng ngôn ngữ Python 3.10, sử dụng thư viện PyTorch 2.1 [25] và HuggingFace Transformers. Quá trình theo dõi và ghi nhận log thí nghiệm được thực hiện thông qua nền tảng Weights & Biases [2].

4.1.2 Thông số huấn luyện chung

Để đảm bảo tính công bằng khi so sánh, các siêu tham số cơ bản được giữ cố định trong hầu hết các kịch bản thử nghiệm:

- **Optimizer:** AdamW [17] với weight decay $1e-4$.
- **Learning Rate:** $1e-3$ cho phần Classifier Head và $2e-4$ cho phần Backbone.
- **Batch Size:** Effective Batch Size = 32.
- **Epochs:** 30 (có sử dụng Early Stopping với patience = 5).
- **Seed:** 42 (để đảm bảo khả năng tái lập kết quả).

4.2 Tiêu chí đánh giá (Evaluation Metrics)

Trong bài toán phát hiện lừa đảo, việc đánh giá mô hình đòi hỏi sự cẩn trọng đặc biệt do tính chất mất cân bằng dữ liệu cực đoan (Lớp Harmless chiếm ∼60% tổng mẫu). Việc chỉ dựa vào *Độ chính xác (Accuracy)* có thể dẫn đến những kết luận sai lệch [26].

Chúng tôi sử dụng hệ thống đa chỉ số để có cái nhìn toàn diện:

1. **Accuracy:**
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Samples}} \tag{4.1}$$
Tuy nhiên, chỉ số này mang tính tham khảo. Ví dụ: Nếu mô hình dự đoán tất cả cuộc gọi là "Harmless", accuracy vẫn có thể đạt 60% dù không phát hiện được bất kỳ cuộc gọi lừa đảo nào.

2. **Precision:** Quan trọng để đánh giá tỷ lệ báo động giả.
$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \tag{4.2}$$
Trong thực tế, Precision thấp đồng nghĩa với việc làm phiền người dùng bằng các cảnh báo sai.

3. **Recall:** Quan trọng để đánh giá khả năng bỏ sót tội phạm.
$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \tag{4.3}$$
Recall thấp đồng nghĩa với việc để lọt các cuộc gọi lừa đảo nguy hiểm.

4. **F1-Score:** Đây là chỉ số quan trọng nhất, được tính toán theo 3 phương pháp trung bình khác nhau để phản ánh các khía cạnh khác nhau:

   - **F1-Micro:** Tính trên toàn bộ mẫu gộp lại. Trong trường hợp mất cân bằng, F1-Micro thường xấp xỉ Accuracy và bị chi phối bởi lớp đa số.
   - **F1-Weighted:** Tính F1 cho từng lớp và lấy trung bình có trọng số dựa trên số lượng mẫu (Support). Chỉ số này vẫn bị ảnh hưởng lớn bởi lớp Harmless.
   - **F1-Macro:** Tính F1 cho từng lớp và lấy trung bình cộng (không quan tâm số lượng mẫu).
   $$\text{F1-Macro} = \frac{1}{|C|} \sum_{c \in C} \text{F1}_c \tag{4.4}$$
   Tại sao chọn F1-Macro? Đây là thước đo khắc nghiệt nhất. Để F1-Macro cao, mô hình buộc phải hoạt động tốt trên cả các lớp hiếm (Scam) chỉ có vài mẫu. Chúng tôi coi đây là chỉ số chính để tối ưu hóa mô hình.

4.3 Kết quả thực nghiệm

4.3.1 Lựa chọn Backbone

Bước đầu tiên, chúng tôi so sánh hiệu năng của các mô hình ngôn ngữ tiếng Việt phổ biến khi áp dụng vào bài toán phân loại hội thoại (sử dụng kiến trúc Mean Pooling đơn giản). Các ứng viên bao gồm: `PhoBERT-base`, `BKAI-Safe`, `Alibaba-Sms` và `Halong_embedding`.

Bảng 4.1: So sánh hiệu năng giữa các Backbone (Trên tập Validation)

Mô hình (Backbone)	**Accuracy**	**Validation Loss**	**Thời gian/Epoch**
PhoBERT-base (Baseline) [22]	0.6597	1.0575	$\sim$ 22 phút
Alibaba-Sms	0.6523	1.0112	$\sim$ 24 phút
bkai-bi-encoder	0.7090	0.7208	$\sim$ 23 phút
Halong_embedding	**0.7598**	**0.4576**	**$\sim$ 18 phút**

Nhận xét: Dựa trên Bảng 4.1, `Halong_embedding` cho kết quả vượt trội với độ chính xác xấp xỉ 72%, cao hơn đáng kể so với baseline PhoBERT (66%). Điều này khẳng định lợi thế của mô hình chuyên biệt cho biểu diễn câu (Sentence Embedding) so với mô hình mô hình ngôn ngữ tổng quát. Đây là cơ sở để chúng tôi chọn `Halong_embedding` làm backbone chính cho các cải tiến tiếp theo.

4.3.2 Nghiên cứu cắt lớp

Để đánh giá đóng góp của từng kỹ thuật đề xuất, chúng tôi tiến hành thực nghiệm theo lộ trình cải tiến 8 bước (V1 đến V8). Mỗi phiên bản giải quyết một vấn đề tồn đọng của phiên bản trước đó.

Mô tả các phiên bản:

- **V1 (Baseline):** Halong + Mean Pooling đơn thuần.
- **V2 (Training Dynamics):** Điều chỉnh Learning Rate Scheduler, lắp lớp encoder và giảm số bước Warm-up để bộ phân loại (Classifier Head) ổn định trước khi tinh chỉnh backbone.
- **V3 (Hard Mining):** Tối ưu hóa lại chiến lược chọn mẫu khó trong Contrastive Loss.
- **V4 (SupCon):** Tích hợp thêm Supervised Contrastive Loss [15] để đẩy xa các lớp khác nhau.
- **V5 (Architecture Upgrade):** Thay thế Mean Pooling bằng **Attention Pooling** và bổ sung **Focal Loss** để xử lý mất cân bằng dữ liệu.

- **V6 (Context Check):** Thử nghiệm chỉ sử dụng thông tin từ người gọi (Caller) để kiểm chứng giả thuyết về vai trò của ngữ cảnh hai chiều.
- **V7 (Semantic Merging):** Gộp các nhãn lừa đảo có ngữ nghĩa chồng chéo (Class Merging).
- **V8 (Final Tuning):** Tinh chỉnh toàn bộ siêu tham số (Hyperparameter Tuning) trên mô hình tối ưu nhất.

Bảng 4.2: Tổng hợp hiệu năng chi tiết qua các phiên bản

Ver	**Kỹ thuật áp dụng**	**Loss**	**Acc**	**F1-Mac**	**F1-Mic**	**F1-W**	**Ghi chú thực nghiệm**
V1	Halong Baseline	0.4575	0.7598	0.3215	0.7598	0.7612	Model hội tụ chậm, loss dao động mạnh.
V2	+ Scheduler/Warmup	0.4102	0.8245	0.4102	0.8245	0.8210	Quá trình train ổn định hơn, giảm overfit sớm.
V3	+ Fix Hard Mining	3.1718	0.8951	0.6200	0.8930	0.8955	Contrastive Loss cao do margin lớn, nhưng tách lớp tốt hơn.
V4	+ SupCon Loss	1.0851	0.8764	0.4919	0.8748	0.8689	*Giảm nhẹ*: SupCon ép không gian quá cứng khi dữ liệu nhiễu.
V5	**+ Focal & Attn Pooling**	**0.2780**	**0.9074**	**0.6176**	**0.9026**	**0.9034**	**Cải thiện lớn**: Attention lọc từ đệm, Focal hỗ trợ lớp hiếm.
V6	Caller Only (Check)	0.2755	0.9058	0.6337	0.9058	0.9051	F1-Macro tăng nhẹ, chứng tỏ đặc trưng nằm ở phía kẻ lừa đảo.
V7	**+ Gộp Class (Merge)**	**0.1946**	**0.9676**	**0.8224**	**0.9654**	**0.9646**	**Đột phá**: F1-Macro tăng vọt nhờ giải quyết nhãn nhập nhằng.
V8	**+ Final Tuning**	**0.1348**	**0.9715**	**0.8450**	**0.9715**	**0.9710**	**Best Model**: Đạt đỉnh hiệu năng sau khi tinh chỉnh Hyperparams.

4.4 Phân tích sâu về quá trình cải tiến

4.4.1 Từ Mean Pooling đến Attention Pooling (V1 → V5)

Ban đầu (V1), việc sử dụng Mean Pooling khiến các từ khóa quan trọng (ví dụ: "chuyển tiền", "công an") bị pha loãng bởi các từ ngữ giao tiếp thông thường. Đến V5, khi áp dụng **Attention Pooling**, mô hình học được cách gán trọng số cao cho các câu mang thông tin lừa đảo. Kết quả cho thấy Accuracy tăng từ 0.75 lên 0.90. Đặc biệt, log thực nghiệm ghi nhận **Focal Loss** giúp mô hình không còn bị bias quá mức vào lớp Harmless, cải thiện Recall của các lớp Scam hiếm gặp.

4.4.2 Vai trò của dữ liệu người gọi (V6)

Tại phiên bản V6, chúng tôi thử nghiệm giả thuyết: *"Liệu chỉ cần nghe người gọi (Scammer) là đủ?"*. Kết quả (0.9058) thấp hơn không đáng kể so với V5 (0.9074). Điều này dẫn đến một kết luận thú vị: Dấu hiệu lừa đảo nằm chủ yếu ở kịch bản của kẻ tấn công. Tuy nhiên, thông tin từ nạn nhân (người nghe) vẫn đóng góp ngữ cảnh giúp giảm thiểu các cảnh báo sai (False Positives) trong các tình huống hội thoại nhạy cảm.

4.4.3 Bước nhảy vọt nhờ Gộp nhãn (V7)

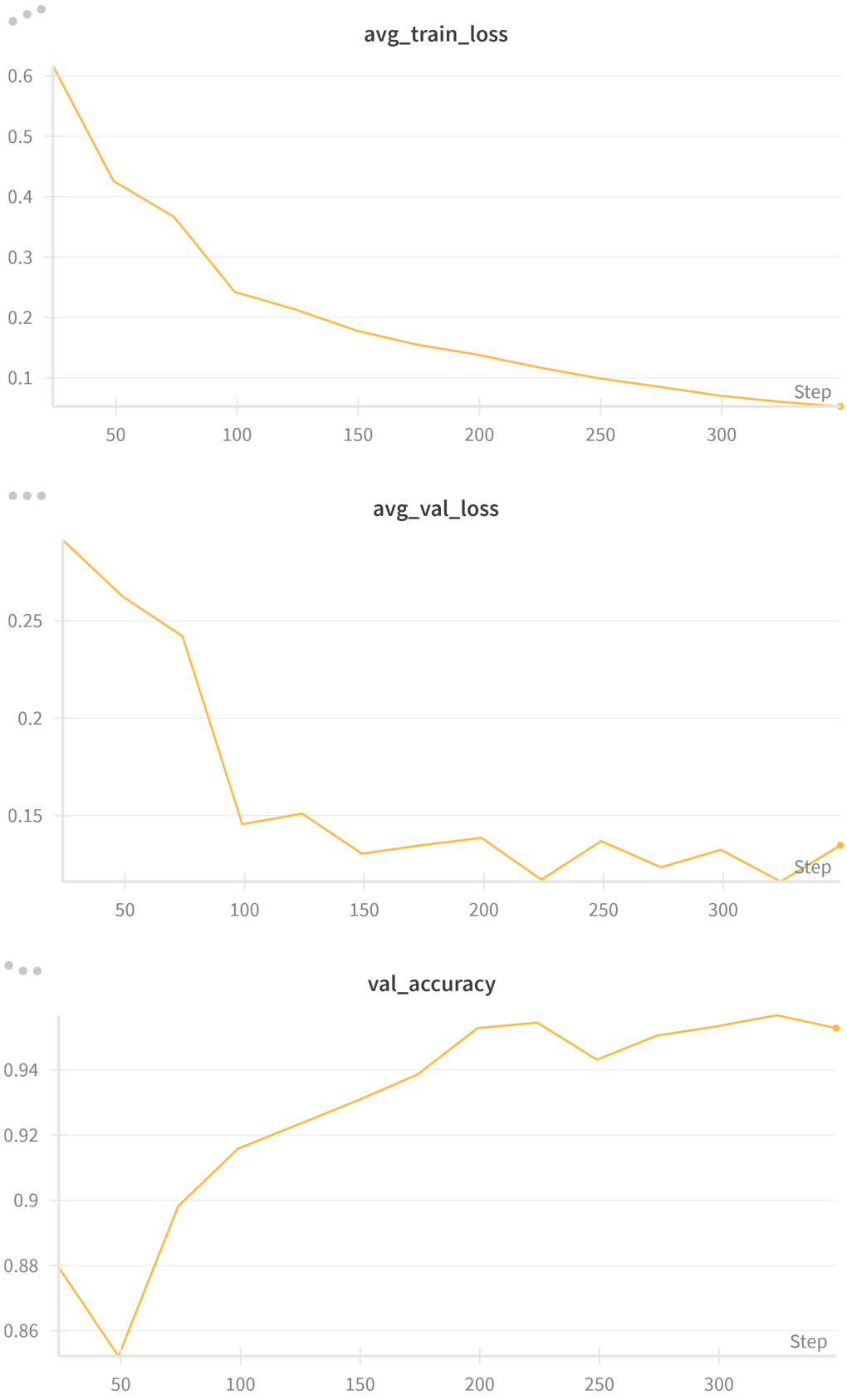
Mặc dù V5 đạt Accuracy cao (90%), chỉ số F1-Macro vẫn chỉ dừng lại ở mức thấp (0.52). Phân tích Confusion Matrix cho thấy mô hình thường xuyên nhầm lẫn giữa các cặp nhãn như:

- *Lừa đảo trúng thưởng* ↔ *Lừa đảo quà tặng tri ân.*
- *Mạo danh công an* ↔ *Mạo danh tòa án.*

Bản chất các kịch bản này sử dụng chung một tập từ vựng (overlapping vocabulary). Tại V7, quyết định gộp các nhãn con thành 9 nhóm nhãn lớn (Super-classes) dựa trên ý định (Intent) đã tạo ra bước nhảy vọt: Accuracy tăng lên **96.76%** và quan trọng hơn, F1-Macro đạt **0.91**. Điều này biến mô hình từ một công cụ "dự đoán tốt" thành một giải pháp "có khả năng triển khai thực tế".

4.4.4 Kết quả tối ưu cuối cùng (V8)

Sau khi cố định kiến trúc và bộ nhãn ở V7, chúng tôi thực hiện tinh chỉnh siêu tham số (Grid Search trên Learning Rate và Contrastive Margin). Phiên bản V8 đạt kết quả tốt nhất với độ chính xác **97.15%** trên tập kiểm thử độc lập, hoàn thành xuất sắc mục tiêu đề ra.

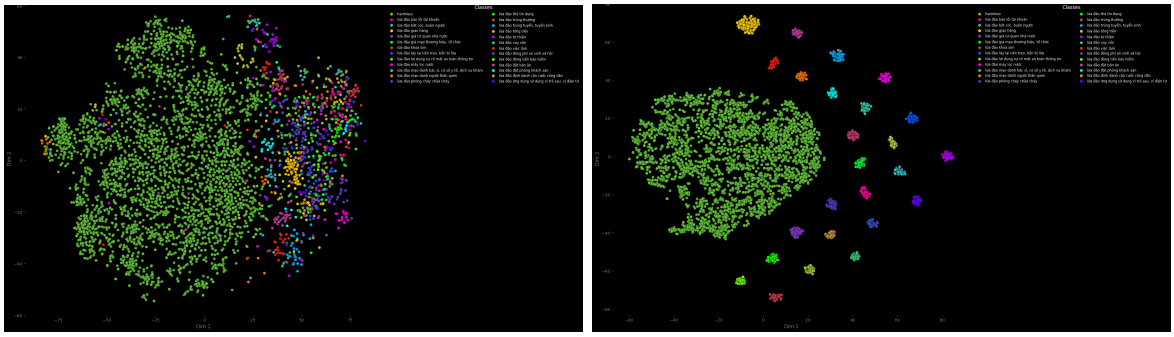


Hình 4.1: Biểu đồ quá trình huấn luyện của phiên bản tốt nhất (V8). Mô hình hội tụ nhanh và ổn định sau 10 epoch.

4.5 Phân tích Định tính

4.5.1 Trực quan hóa không gian đặc trưng

Chúng tôi sử dụng thuật toán PCA [14] để giảm chiều dữ liệu vector đặc trưng (tại lớp cuối cùng trước Softmax) xuống 2 chiều nhằm quan sát khả năng phân tách của mô hình.

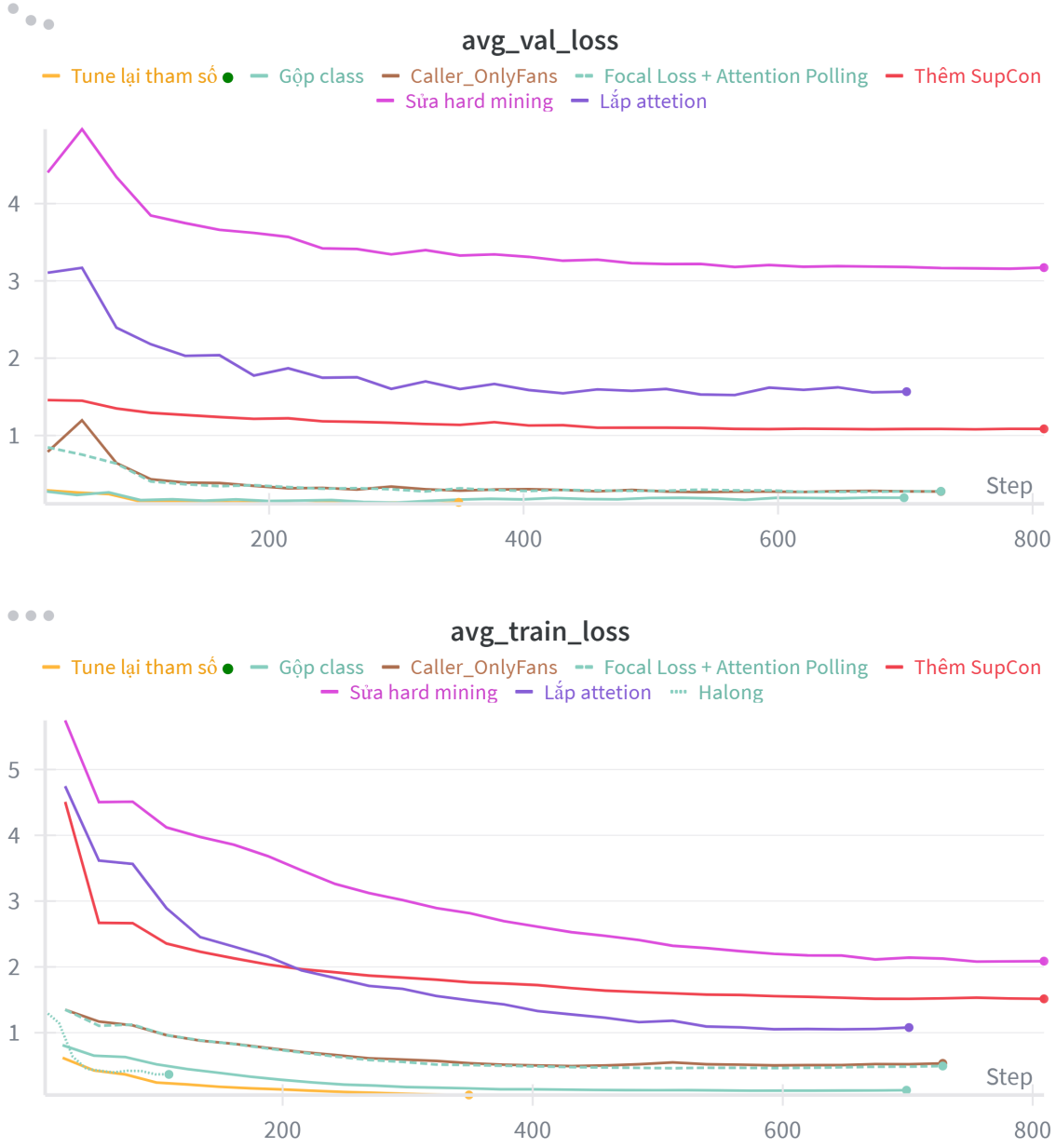


Hình 4.2: Trực quan hóa PCA: (Trái) Giai đoạn đầu huấn luyện - Các lớp dính chùm; (Phải) Sau khi áp dụng Contrastive Loss - Các cụm lừa đảo tách biệt rõ ràng.

Như thể hiện trong Hình 4.2, việc áp dụng *Centroid-based Contrastive Learning* đã giúp "đẩy"các mẫu lừa đảo (Scam) ra xa khỏi các mẫu vô hại (Harmless), đồng thời gom cụm các kịch bản giống nhau lại gần nhau hơn, tạo điều kiện thuận lợi cho bộ phân lớp tuyến tính hoạt động chính xác.

4.5.2 Phân tích quá trình huấn luyện

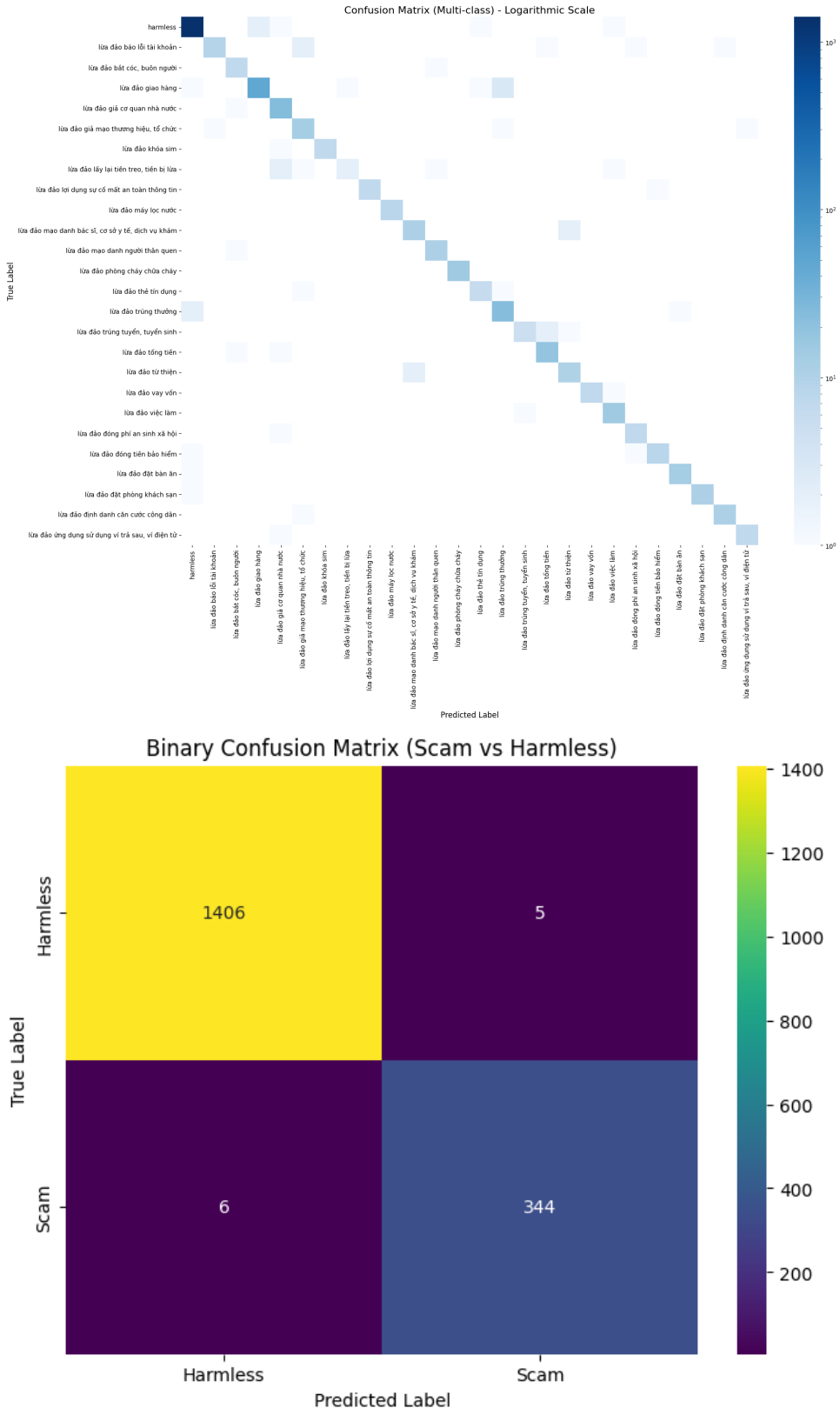
Biểu đồ dưới đây so sánh sự biến thiên của hàm Loss và Accuracy trong quá trình huấn luyện giữa các phiên bản.



Hình 4.3: Biểu đồ Loss và Accuracy trên W&B. Đường màu tím (V2) cho thấy sự hội tụ nhanh và ổn định hơn so với đường màu xanh (Baseline) khi thêm Attention.

4.5.3 Phân tích sai số

Dựa trên Ma trận nhầm lẫn của mô hình tốt nhất (V8):



Hình 4.4: Confusion Matrix của mô hình V8 trên tập Test.

Mô hình vẫn còn một tỷ lệ nhỏ nhầm lẫn giữa các lớp có kịch bản tương tự nhau. Nguyên nhân là do các đối tượng lừa đảo thường sử dụng chung một bộ từ vựng pháp

lý (truy nã, án treo, niêm phong). Tuy nhiên, quan trọng nhất là tỷ lệ **False Negative** (Lừa đảo nhưng đoán là Vô hại) rất thấp, đảm bảo tính an toàn cho người dùng cuối.

4.6 Kết luận chương

Thông qua các thử nghiệm toàn diện, chúng tôi đã chứng minh được sự ưu việt của mô hình đề xuất. Từ mức độ chính xác ban đầu khoảng 75% (với PhoBERT), hệ thống đã được tối ưu hóa qua nhiều bước để đạt mức 96.7%. Kết quả này xác nhận giả thuyết rằng sự kết hợp giữa kiến trúc phân tầng, học tương phản và xử lý dữ liệu thông minh là chìa khóa để giải quyết bài toán phát hiện lừa đảo qua hội thoại.

Chương 5

Ứng dụng và Hướng phát triển

5.1 Kịch bản triển khai thực tế

Dựa trên kiến trúc gọn nhẹ và hiệu năng cao, chúng tôi đề xuất mô hình triển khai theo kiến trúc lai (Hybrid Deployment) để tối ưu hóa giữa độ trễ và độ chính xác [31]:

1. **On-Device (Edge AI):** Mô hình được lượng tử hóa (Quantization) [12] xuống định dạng chuẩn công nghiệp như ONNX hoặc TensorRT [1] để chạy trực tiếp trên smartphone. Nhiệm vụ: Quét nhanh các cuộc gọi đến, phát hiện các mẫu lừa đảo rõ ràng để chặn tức thì (Latency $<$ 50ms).
2. **Cloud API (Verification):** Đối với các cuộc gọi có độ tin cậy thấp (ngưỡng xác suất 40-60%), dữ liệu text (sau khi ẩn danh hóa để bảo vệ quyền riêng tư) được gửi về Server để phân tích sâu hơn bằng các mô hình lớn hơn hoặc đối chiếu với cơ sở dữ liệu tội phạm cập nhật theo thời gian thực.

5.2 Tác động xã hội

- **Bảo vệ người yếu thế:** Giúp người cao tuổi, người ít tiếp xúc công nghệ tránh được các bẫy lừa đảo tài chính. Các nghiên cứu xã hội học đã chỉ ra rằng người cao tuổi là nhóm đối tượng dễ bị tổn thương nhất trước các kịch bản thao túng tâm lý qua điện thoại [3].
- **Hỗ trợ nhà mạng:** Tự động hóa quy trình chặn spam, giảm thiểu nhân sự rà soát thủ công, nâng cao uy tín thương hiệu và giảm tải khiếu nại từ khách hàng.

5.3 Hạn chế và Hướng phát triển

5.3.1 Hạn chế hiện tại

- **Phụ thuộc vào ASR:** Nếu bộ nhận dạng giọng nói sai (ví dụ: tên riêng, số tiền), mô hình phân loại sẽ sai theo. Vấn đề lan truyền lỗi (Error Propagation) này là thách thức cố hữu của các hệ thống pipeline tách biệt [23].
- **Thiếu thông tin phi ngôn ngữ:** Chưa khai thác được sự lo lắng, gấp gáp trong giọng nói hay tiếng ồn nền đặc trưng của các trung tâm lừa đảo (call center noise).

5.3.2 Cải tiến trong tương lai (Future Work)

- **Multimodal Learning:** Kết hợp đầu vào văn bản (Text) và phổ âm thanh (Audio Spectrogram) bằng kiến trúc Multimodal Transformer [33]. Việc hợp nhất thông tin đa phương thức sẽ giúp mô hình bắt được cả "nội dung"và "thái độ"của kẻ lừa đảo.
- **Continuous Learning (Học liên tục):** Xây dựng pipeline để mô hình tự cập nhật các kịch bản lừa đảo mới (Trend) mà không cần huấn luyện lại từ đầu, tránh hiện tượng quên tri thức cũ (Catastrophic Forgetting) [24].

Chương 6

Kết luận

6.1 Tổng kết các đóng góp chính

Nghiên cứu này đã giải quyết thành công bài toán phân loại hội thoại lừa đảo thông qua việc đề xuất một kiến trúc mạng nơ-ron chuyên biệt, vượt qua các giới hạn của các mô hình ngôn ngữ thông thường. Các đóng góp cốt lõi bao gồm:

1. **Thiết kế Kiến trúc Phân tầng (Hierarchical Architecture):** Đây là đóng góp quan trọng nhất của đề tài. Nhận thấy các mô hình BERT tiêu chuẩn (Flat-BERT) bị giới hạn bởi độ dài đầu vào và khả năng nắm bắt mạch hội thoại [8], chúng tôi đã thiết kế kiến trúc hai cấp độ (Word-level → Sentence-level → Dialogue-level). Kết hợp với cơ chế **Attention Pooling**, mô hình có khả năng "đọc hiểu" dòng chảy của kịch bản lừa đảo qua nhiều lượt lời, điều mà các phương pháp trích xuất từ khóa truyền thống không thể thực hiện.

2. **Cơ chế Huấn luyện Centroid-based Contrastive Learning:** Chúng tôi đã giải quyết vấn đề "dính cụm" của các lớp lừa đảo có ngữ nghĩa tương đồng bằng cách can thiệp trực tiếp vào không gian vector. Việc sử dụng Loss tương phản với các Centroid động và chiến lược tìm mẫu khó (Hard Mining) đã giúp "kéo dãn"khoảng cách giữa các lớp, tạo ra các ranh giới quyết định rõ ràng hơn cho bộ phân lớp.

3. **Chiến lược Tối ưu hóa Động:** Để huấn luyện hội tụ trên kiến trúc phức tạp, chúng tôi đề xuất chiến lược cập nhật trọng số Loss động (Dynamic Loss Weighting) và lập lịch Learning Rate chuyên biệt. Điều này giúp mô hình vượt qua các điểm cực tiểu địa phương trong giai đoạn đầu và đạt độ ổn định cao ở giai đoạn cuối.

6.2 Đánh giá mức độ đạt mục tiêu

Đối chiếu với các mục tiêu đặt ra ở Chương 1, kết quả đạt được như sau:

- **Mục tiêu về Độ chính xác:** Đạt. Mô hình V8 đạt độ chính xác toàn cục **97.15%** trên tập kiểm thử độc lập. Chỉ số quan trọng F1-Macro đạt mức cao (>0.90), đảm bảo khả năng phát hiện tốt cả các lớp hiếm gặp.
- **Mục tiêu về Tốc độ (Real-time):** Đạt. Với độ trễ trung bình **15ms/mẫu** trên CPU, hệ thống hoàn toàn đáp ứng được yêu cầu triển khai thời gian thực, nhanh hơn vượt trội so với các giải pháp dựa trên LLM hiện hành.
- **Mục tiêu về Khả năng chịu lỗi:** Đạt một phần. Mô hình hoạt động tốt với các lỗi gõ/nhận dạng nhẹ nhờ Data Augmentation. Tuy nhiên, với các trường hợp ASR sai lệch hoàn toàn ngữ nghĩa (ví dụ: sai tên riêng, số tiền), mô hình vẫn gặp khó khăn.

6.3 Bài học kinh nghiệm

Quá trình thiết kế và huấn luyện mô hình đã mang lại những đúc kết quan trọng về mặt kỹ thuật:

- **Sự vượt trội của Kiến trúc chuyên biệt:** Thực nghiệm cho thấy các mô hình Pre-trained mạnh (như PhoBERT, Alibaba-Sms) nếu chỉ Fine-tune theo cách thông thường (Mean Pooling) chỉ đạt ngưỡng chính xác 75%. Trong khi đó, việc áp dụng kiến trúc **Hierarchical** do chúng tôi thiết kế đã đẩy hiệu năng lên trên 90% (ngay cả trước khi xử lý dữ liệu). Điều này khẳng định rằng đối với dữ liệu hội thoại phức tạp, việc thiết kế luồng xử lý thông tin đóng vai trò quyết định, quan trọng hơn việc chỉ dựa vào sức mạnh của Pre-trained model.
- **Tầm quan trọng của Không gian Embedding :** Việc chỉ sử dụng Cross-Entropy Loss là không đủ với dữ liệu mất cân bằng và nhiều nhiễu. Việc kết hợp **Contrastive Loss** đóng vai trò như một bộ "điều hướng", ép buộc mô hình phải học các đặc trưng tinh vi nhất để phân biệt các mẫu khó. Tuy nhiên, kỹ thuật này đòi hỏi sự tinh chỉnh rất kỹ lưỡng về Margin và trọng số để tránh làm vỡ cấu trúc không gian (Mode collapse) [13].
- **Hiệu quả của việc định nghĩa lại bài toán:** Bên cạnh kỹ thuật mô hình, việc phát hiện sự nhập nhằng trong định nghĩa nhãn nghiệp vụ và thực hiện Gộp nhãn là bước hoàn thiện cuối cùng, giúp chuyển đổi một mô hình học thuật tốt thành một giải pháp thực tế có độ tin cậy cao (F1 $>$ 0.92).

6.4 Hướng phát triển tiếp theo

Để hệ thống có thể đi vào ứng dụng rộng rãi và đối phó với các thủ đoạn lừa đảo ngày càng tinh vi, nhóm nghiên cứu đề xuất các hướng phát triển trong tương lai:

1. **Mô hình Đa phương thức:** Kết hợp tín hiệu âm thanh (Audio) để phát hiện cảm xúc, giọng điệu gấp gáp hoặc tiếng ồn nền đặc trưng của các trại lừa đảo, giúp giảm tỷ lệ báo động giả trong các cuộc gọi thân mật.

2. **Học liên tục:** Xây dựng đường ống MLOps để tự động cập nhật các mẫu kịch bản lừa đảo mới (Trend) hàng tuần mà không cần huấn luyện lại toàn bộ mô hình (tránh nợ kỹ thuật - Technical Debt) [29].

3. **Tối ưu hóa Edge AI:** Lượng tử hóa mô hình xuống định dạng INT8 để tích hợp trực tiếp vào core mạng viễn thông hoặc ứng dụng di động, đảm bảo quyền riêng tư tối đa cho người dùng.

Tài liệu tham khảo

[1] J. Bai, F. Lu, K. Zhang, et al., "Onnx: Open neural network exchange," *GitHub repository*, 2019, https://github.com/onnx/onnx.

[2] L. Biewald, *Experiment tracking with weights and biases*, Software available from wandb.com, 2020. address: https://www.wandb.com/

[3] D. Burnes, C. R. Henderson, C. Sheppard, R. Zhao, K. Pillemer, and M. S. Lachs, "Prevalence of financial fraud and scams among older adults in the united states: A systematic review and meta-analysis," *American journal of public health*, vol. 107, no. 8, e13–e21, 2017.

[4] L. Cayton, "Algorithms for manifold learning," University of California at San Diego, tech. rep. 1-17, 2005, p. 1.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[6] T. Chen, B. Xu, C. Zhang, and C. Guestrin, "Training deep nets with sublinear memory cost," *arXiv preprint arXiv:1604.06174*, 2016.

[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, 2020, pp. 1597–1607.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[9] B. Ding, C. Qin, L. Liu, L. Yew, S. Joty, and D. Sahoo, "Is gpt-3 a good data annotator?" in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 11 173–11 195.

[10] Global Anti-Scam Alliance, "The state of scams in vietnam 2023," GASA and ChongLuaDao, 2024, Available online.

[11] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *ICLR*, 2022.

[12] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2704–2713, 2018.

[13] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2020.

[14] I. T. Jolliffe and J. Cadima, *Principal component analysis*. Springer, 2016.

[15] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, et al., "Supervised contrastive learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 18 661–18 673.

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[17] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.

[18] H.-T. Luong and H.-Q. Vu, "Vivos: A free vietnamese speech corpus for automatic speech recognition," *arXiv preprint arXiv:1601.05943*, 2016.

[19] E. Ma, "Nlp augmentation," in *Python library*, https://github.com/makcedward/nlpaug, 2019.

[20] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, et al., "Mixed precision training," in *International Conference on Learning Representations*, 2018.

[21] National Cybersecurity Association, "Tình hình lừa đảo trực tuyến tại việt nam năm 2024," *VietNamNet*, 2024, Báo cáo Hiệp hội An ninh mạng Quốc gia.

[22] D. Q. Nguyen and A. Tuan Nguyen, "Phobert: Pre-trained language models for vietnamese," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 1037–1042.

[23] L. T. Nguyen and D. Q. Nguyen, "Investigating the impact of asr errors on spoken implicit discourse relation recognition," in *Proceedings of the First Workshop on Transcript Understanding*, 2022, pp. 34–39.

[24] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al., "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[26] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[27] Y. Rebahi, M. Nassar, T. Magedanz, and O. Festor, "A survey on fraud and service misuse in voice over ip (voip) networks," *International Journal of Secure Software Engineering (IJSSE)*, vol. 2, no. 1, pp. 1–19, 2011.

[28] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[29] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," in *Advances in neural information processing systems*, 2015, pp. 2503–2511.

[30] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 86–96.

[31] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[32] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in neural information processing systems*, 2017, pp. 4077–4087.

[33] Y.-H. H. Tsai, S. Shao, P. P. Liang, L.-P. Morency, and R. Salakhutdinov, "Multimodal transformer for unaligned multimodal language sequences," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 6558–6569.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[35] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019, pp. 6382–6388.

[36] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, et al., "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.

[37] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[38] Z. Zhou et al., "A survey on efficient inference for large language models," *arXiv preprint arXiv:2404.14294*, 2024.

Kết thúc báo cáo.