

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN**

HOÀNG MINH NHẬT - 24550031

KHÓA LUẬN TỐT NGHIỆP

**HỆ THỐNG GIỚI THIỆU KẾT BẠN DỰA TRÊN
TÍNH CÁCH, XỬ LÝ TRỰC TIẾP TRÊN THIẾT
BỊ ĐỂ BẢO VỆ QUYỀN RIÊNG TƯ**

**On-device personality-based friend recommendation for privacy-
preserving social networking**

CỬ NHÂN NGÀNH CÔNG NGHỆ THÔNG TIN

TP. HỒ CHÍ MINH, 2025

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN

HOÀNG MINH NHẬT - 24550031

KHÓA LUẬN TỐT NGHIỆP

**HỆ THỐNG GIỚI THIỆU KẾT BẠN DỰA TRÊN
TÍNH CÁCH, XỬ LÝ TRỰC TIẾP TRÊN THIẾT
BỊ ĐỂ BẢO VỆ QUYỀN RIÊNG TƯ**

**On-device personality-based friend recommendation for privacy-
preserving social networking**

CỦ NHÂN NGÀNH CÔNG NGHỆ THÔNG TIN

CÁN BỘ HƯỚNG DẪN: ThS Ngô Khánh Khoa

TP. HỒ CHÍ MINH, 2025

THÔNG TIN HỘI ĐỒNG CHẤM KHÓA LUẬN TỐT NGHIỆP

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số
ngày của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến Ban giám hiệu Trường Đại học Công nghệ Thông tin – ĐHQG-HCM cùng các Thầy Cô trong Khoa đã tạo điều kiện và truyền đạt những kiến thức quý báu trong suốt quá trình em học tập tại trường. Đặc biệt, em xin bày tỏ lòng tri ân sâu sắc nhất đến thầy Ngô Khánh Khoa. Trong suốt thời gian thực hiện đề tài, thầy không chỉ hỗ trợ tận tình về mặt kiến thức chuyên môn mà còn là một người thầy tâm huyết, luôn dành thời gian sát cánh và đưa ra những lời khuyên sâu sắc giúp em hoàn thành tốt khóa luận này.

TP. Hồ Chí Minh, ngày 15 tháng 1 năm 2026

Sinh viên thực hiện

Hoàng Minh Nhật

MỤC LỤC

Thông tin hội đồng.....	3
Lời cảm ơn.....	0
Mục lục.....	1
Danh mục hình ảnh.....	4
Danh mục bảng.....	5
Danh mục giải thuật.....	6
Danh mục từ viết tắt.....	7
Tóm tắt.....	1
Chương 1. Giới thiệu.....	2
1.1. Bối cảnh và vấn đề	2
1.2. Mục tiêu, phạm vi và cách tiếp cận	4
1.3. Đóng góp chính	7
1.4. Cấu trúc của báo cáo	7
Chương 2. Tổng quan hệ thống.....	8
2.1. Các nguồn dữ liệu đầu vào	9
2.2. Tổng quan quy trình và tác nhân	10
2.3. Mô hình giới thiệu và trọng số trong giới thiệu	11
2.4. Luồng dữ liệu chi tiết theo tác nhân	13
Chương 3. Đề xuất và Triển khai.....	15
3.1. Chuyển đổi dữ liệu tính cách (PCA-4)	15
3.1.1. Chuẩn hóa điểm Big Five	17
3.1.2. Đề xuất PCA-4	18

3.1.3.	Huấn luyện và Trích xuất tham số PCA	19
3.1.4.	Triển khai PCA trên thiết bị	21
3.2.	Bảo mật và mã hoá dữ liệu	22
3.2.1.	Tổng quan về cơ chế AES-GCM	22
3.2.2.	Dữ liệu đầu vào từ góc nhìn người dùng	24
3.2.3.	Mã hóa dữ liệu bằng AES-256-GCM	25
3.2.4.	Dữ liệu sở thích và mã hóa	30
3.3.	Hệ giới thiệu và cơ chế xếp hạng	30
3.3.1.	Đề xuất thuật toán ELO	31
3.3.2.	Nguồn sử dụng sở thích	33
3.3.3.	Đề xuất mô hình ngữ nghĩa (semantic model)	33
3.3.4.	Công thức xếp hạng tổng hợp	36
3.3.5.	Bảo vệ dữ liệu sở thích và quyền riêng tư	38
Chương 4.	Hiện thực và Đánh giá.....	39
4.1.	Câu hỏi nghiên cứu (Research Questions)	39
4.2.	Thiết lập thực nghiệm	40
4.3.	Kết quả và phân tích	41
4.3.1.	RQ1: Hiệu quả của mô hình PCA-4 và độ tương đồng cosine ..	41
4.3.2.	RQ2: Đánh giá hệ thống giới thiệu lai	42
4.3.3.	RQ3: Phân tích hiệu năng	42
4.3.4.	RQ4-5: Ngữ nghĩa sở thích và quyền riêng tư	43
4.3.5.	RQ6: Kiểm thử suy luận gián tiếp qua Edge Functions	45
4.4.	Thảo luận và Hạn chế	45

4.5. Kết quả thực nghiệm chi tiết	47
Chương 5. Kết luận và hướng phát triển.....	50
5.1. Tổng kết kết quả đạt được	50
5.2. Hạn chế và hướng phát triển tương lai	50
TÀI LIỆU THAM KHẢO.....	52
PHỤ LỤC.....	54
A.. Thông tin về mã nguồn và cơ sở dữ liệu	54
B.. Quy trình phân tích và tính toán trọng số PCA	55
C.. Bộ câu hỏi Big Five (IPIP-50) Anh - Việt	55
D.. Mã nguồn thực nghiệm và các hàm quan trọng	59
D.1.. Kịch bản kiểm chứng độ tương đồng PCA (pca_worst_best_case.ts)	59
D.2.. Mã hóa và Giải mã dữ liệu (score-crypto)	60
D.3.. Thuật toán Giới thiệu người dùng lai (recommend-users)	61
D.4.. Kịch bản đo hiệu năng hệ thống (Benchmarks)	61
D.5.. Kịch bản kiểm thử suy luận gián tiếp qua Edge Functions	62
D.6.. Kỹ thuật tối ưu hóa cơ sở dữ liệu	63
D.7.. Minh họa kịch bản kiểm thử logic tính điểm	64
D.8.. Kịch bản kiểm chứng độ tương đồng ngữ nghĩa	65

DANH MỤC HÌNH ẢNH

Hình 1.1	Bối cảnh ứng dụng mạng xã hội và nhu cầu kết nối theo tính cách	3
Hình 1.2	Rủi ro khi xử lý dữ liệu tính cách theo mô hình tập trung	4
Hình 1.3	Quy trình giới thiệu Big Five sang vector PCA-4	6
Hình 1.4	Luồng mã hoá AES-GCM và lưu trữ dữ liệu tính cách	6
Hình 2.1	Quy trình tổng thể của hệ thống Twins	10
Hình 2.2	Sơ đồ trọng số tính giới thiệu xếp hạng	13
Hình 2.3	Đoạn log tại edge function thể hiện quá trình mã hoá dữ liệu được gửi từ người dùng.	14
Hình 3.1	Minh họa mô hình MBTI và cách phân nhóm tính cách	16
Hình 3.2	Minh họa cấu trúc 6 yếu tố của HEXACO	17
Hình 3.3	Minh họa tiêu chí lựa chọn PCA-4	19
Hình 3.4	Định dạng đầu vào/đầu ra của AES-GCM	24
Hình 3.5	Luồng giao diện và vị trí tổng hợp điểm Big Five	25
Hình 3.6	Ví dụ chi phí tính toán khi dùng RSA cho payload nhỏ	26
Hình 3.7	So sánh dữ liệu băm và dữ liệu có thể giải mã	27
Hình 3.8	Minh họa độ phức tạp của mã hoá đồng hình	28
Hình 3.9	So sánh sự riêng tư biệt lập và mã hoá dữ liệu cá nhân	28
Hình 3.10	Nhật ký Edge Function khi mã hoá và giải mã dữ liệu	29
Hình 3.11	Ví dụ dữ liệu đã mã hoá của Big Five trong bảng profiles	29
Hình 3.12	Luồng mã hoá dữ liệu sở thích và lưu trữ vector nhúng	30
Hình 3.13	Ví dụ ELO phản ánh hành vi xã giao qua chuỗi tương tác	32
Hình 3.14	Mức độ tương đồng ngữ nghĩa của hai từ được so sánh bằng cosine similarity.	34
Hình 3.15	TF-IDF vượt trội ở khả năng tìm kiếm từ khoá quan trọng.	35
Hình 3.16	Word2Vec vượt trội trong việc tìm quan hệ ngữ nghĩa giữa các từ.	36

DANH MỤC BẢNG

Bảng 4.1 Độ trễ phản hồi của các thành phần hệ thống	43
Bảng 4.2 Hiệu năng quy trình tạo tài khoản	47
Bảng 4.3 So sánh hiệu năng giới thiệu trước và sau tối ưu hoá	47
Bảng 4.4 Kết quả xếp hạng thực tế sau khi tối ưu hoá	48
Bảng 1.1 Danh sách 50 câu hỏi Big Five (IPIP-50) Anh - Việt	55

DANH MỤC GIẢI THUẬT

Thuật toán 2.1	Thuật toán tính điểm giới thiệu tổng hợp	12
Thuật toán 3.1	Phép chiếu PCA giảm chiều dữ liệu tính cách	21
Thuật toán 3.2	Quy trình mã hoá và xác thực dữ liệu bằng AES-256-GCM	23
Thuật toán 3.3	Quy tắc cập nhật điểm ELO hợp tác	31
Thuật toán 3.4	Tính toán kỳ vọng thắng trong mô hình Elo	31
Thuật toán 3.5	Tính toán độ gần (proximity) ELO	31
Thuật toán 3.6	Công thức tính độ tương đồng cosine (Cosine Similarity)	33
Thuật toán 1.1	Minh họa các kịch bản kiểm thử trong pca_worst_best_case.ts	59
Thuật toán 1.2	Hiện thực hàm mã hoá và giải mã điểm tính cách	60
Thuật toán 1.3	Hiện thực hàm giới thiệu người dùng lai	61
Thuật toán 1.4	Kịch bản đo hiệu năng chi tiết	61
Thuật toán 1.5	Kịch bản kiểm thử suy luận gián tiếp qua Edge Functions	62
Thuật toán 1.6	Mã nguồn SQL tối ưu hoá cơ sở dữ liệu	63
Thuật toán 1.7	Minh họa các kịch bản kiểm thử trong score_verifier.ts	64
Thuật toán 1.8	Minh họa các kịch bản trong demo_semantic_comparison.ts ..	65

DANH MỤC TỪ VIẾT TẮT

Viết tắt	Tên đầy đủ / Thuật ngữ	Giải thích
PCA	Principal Component Analysis	Phân tích thành phần chính. Kỹ thuật giảm chiều dữ liệu được sử dụng để nén vector tính cách 5 chiều xuống 4 chiều nhằm tối ưu hóa lưu trữ và so khớp.
Big Five / OCEAN	Big Five Personality Traits	Mô hình 5 yếu tố tính cách lớn bao gồm: Cởi mở (Openness), Tận tâm (Conscientiousness), Hướng ngoại (Extraversion), Hòa đồng (Agreeableness), và Bất ổn cảm xúc (Neuroticism).
Cosine Similarity	Độ tương đồng Cosine	Độ đo góc giữa hai vector khác 0 trong không gian tích vô hướng, được dùng để tính toán mức độ phù hợp giữa tính cách của hai người dùng.
ELO	ELO Rating System	Hệ thống xếp hạng ban đầu dành cho cờ vua, được đề tài cải tiến để mô hình hóa hành vi tương tác xã hội (thích/bỏ qua) của người dùng.
AES-256-GCM	Advanced Encryption	Chuẩn mã hóa đối xứng với độ dài khóa 256-bit kết hợp chế độ xác

Viết tắt	Tên đầy đủ / Thuật ngữ	Giải thích
	Standard - Galois/ Counter Mode	thực GCM, đảm bảo cả tính bí mật và tính toàn vẹn của dữ liệu tính cách.
HE	Homomorphic Encryption	Mã hóa đồng hình. Một dạng mã hóa cho phép thực hiện tính toán trực tiếp trên dữ liệu mã hóa mà không cần giải mã (đề cập trong hướng phát triển).
RLS	Row Level Security	Chính sách bảo mật mức hàng trong PostgreSQL, cho phép kiểm soát quyền truy cập dữ liệu chi tiết đến từng bản ghi dựa trên định danh người dùng.
Edge Functions	Edge Computing Functions	Các hàm thực thi tại biên (serverless) giúp xử lý các tác vụ tính toán nhẹ và mã hóa dữ liệu gần người dùng nhất để giảm độ trễ.
Cold Start	Ván đề khởi động lạnh	Tình trạng hệ thống gợi ý hoạt động kém hiệu quả khi người dùng mới tham gia chưa có lịch sử tương tác. Đề tài giải quyết bằng cách dùng tính cách làm dữ liệu khởi tạo.
UI / UX	User Interface	Giao diện người dùng và Trải nghiệm người dùng.

Viết tắt	Tên đầy đủ / Thuật ngữ	Giải thích
	User Experience	
JSON	JavaScript Object Notation	Định dạng trao đổi dữ liệu văn bản nhẹ, ngôn ngữ độc lập, được sử dụng để truyền tải dữ liệu giữa Client và Server.

TÓM TẮT

Khóa luận tập trung giải quyết bài toán xây dựng hệ thống giới thiệu kết bạn dựa trên sự tương đồng về tính cách, đồng thời đảm bảo quyền riêng tư cho dữ liệu người dùng thông qua mô hình xử lý trực tiếp trên thiết bị (on-device processing). Trong bối cảnh các mạng xã hội hiện đại đang đổi mới với nhiều thách thức về bảo mật thông tin nhạy cảm, đề tài đề xuất một quy trình hoàn chỉnh từ thu thập dữ liệu trả lời câu hỏi của mô hình tính cách Big-5, chuyển đổi sang không gian vector thu gọn bằng Phân tích Thành phần chính (PCA-4), cho đến việc lưu trữ an toàn bằng chuẩn mã hóa AES-256-GCM thông qua các hàm thực thi biên (Edge Functions). Hệ thống sử dụng thuật toán giới thiệu lai kết hợp giữa độ tương đồng tính cách, hành vi xã giao qua điểm số ELO và nhúng ngữ nghĩa (semantic embedding) cho sở thích cá nhân. Kết quả thực nghiệm trên hệ thống thật cho thấy mô hình PCA-4 giữ được hơn 90% phương sai dữ liệu gốc, cho phép phân biệt hiệu quả giữa các hồ sơ tương đồng và khác biệt. Hiệu năng hệ thống được tối ưu hóa đạt mức trễ chấp nhận được với trung bình 1.8 giây cho quy trình đăng ký và 2.3 giây cho việc tạo danh sách giới thiệu, đồng thời khẳng định tính khả thi của kiến trúc “Quyền riêng tư theo thiết kế” trong việc bảo vệ dữ liệu nhạy cảm mà vẫn duy trì hiệu quả kết nối cộng đồng.

Từ khóa: Mạng xã hội, Tâm lý học, Big Five, PCA, Quyền riêng tư, Hệ thống giới thiệu, Mã hóa dữ liệu

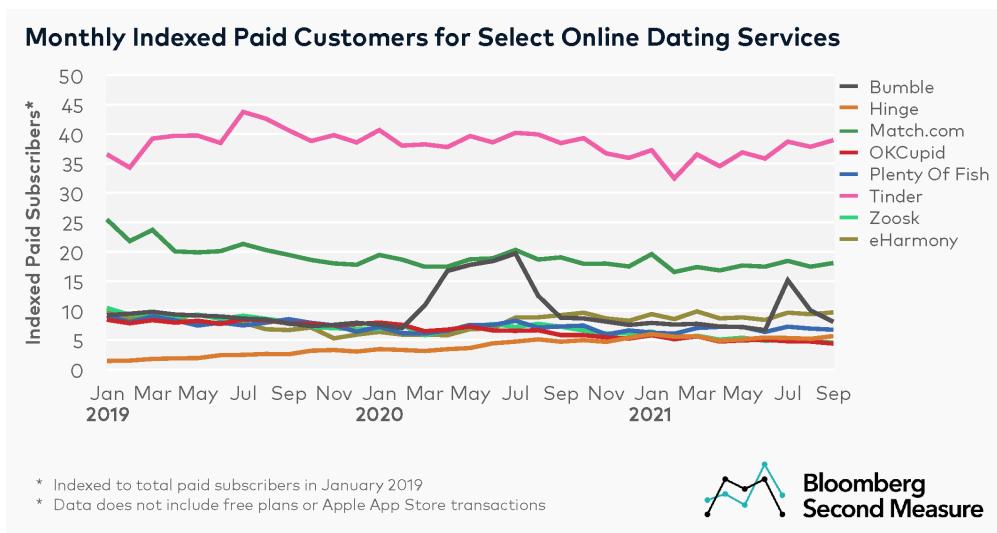
Chương 1 Giới thiệu

1.1 Bối cảnh và vấn đề

Twins là một ứng dụng mạng xã hội theo hướng bán khép kín, tập trung vào các cộng đồng nhỏ và chất lượng. Ứng dụng hướng tới việc tìm bạn có tính cách và sở thích tương đồng, lấy cảm hứng từ cơ chế lướt của Tinder và yếu tố kết nối thân mật của Locket. Khác với những nền tảng đại trà, Twins ưu tiên kết nối có chiều sâu thay vì số lượng tương tác. Mục tiêu này dẫn tới việc giảm bớt các tín hiệu bè mặt và tăng trọng số cho các yếu tố phản ánh đặc trưng cá nhân ổn định hơn. Về mặt trải nghiệm, người dùng được dẫn qua một chuỗi câu hỏi ngắn gọn để trích xuất tính cách, sau đó dùng kết quả này như một “dấu vân tính cách” phục vụ giới thiệu và phân nhóm.

Các nền tảng mạng xã hội và ứng dụng kết nối hiện nay thường tối ưu cho tốc độ ghép cặp và số lượt tương tác, dựa trên yếu tố vị trí, sở thích bè mặt hoặc mạng bạn bè sẵn có. Cách tiếp cận này tạo ra nhiều kết quả, nhưng chưa chắc dẫn tới sự tương hợp lâu dài. Trong khi đó, các mô hình tính cách như Big-5 được xem là khung tham chiếu ổn định, có khả năng giải thích xu hướng hành vi và mức độ phù hợp giữa các cá nhân [1], [2].

Ở góc nhìn của đề tài, nhu cầu kết nối theo tính cách có ý nghĩa vì nó gắn với các đặc trưng ít thay đổi theo thời gian, nên phù hợp cho bài toán giới thiệu dài hạn. Lựa chọn này cũng tránh việc phụ thuộc quá nhiều vào dữ liệu tương tác ngắn hạn, vốn dễ bị ảnh hưởng bởi bối cảnh, tâm trạng hoặc hiệu ứng thuật toán. [Hình 1.1](#) minh họa bối cảnh ứng dụng và mục tiêu kết nối theo tính cách.



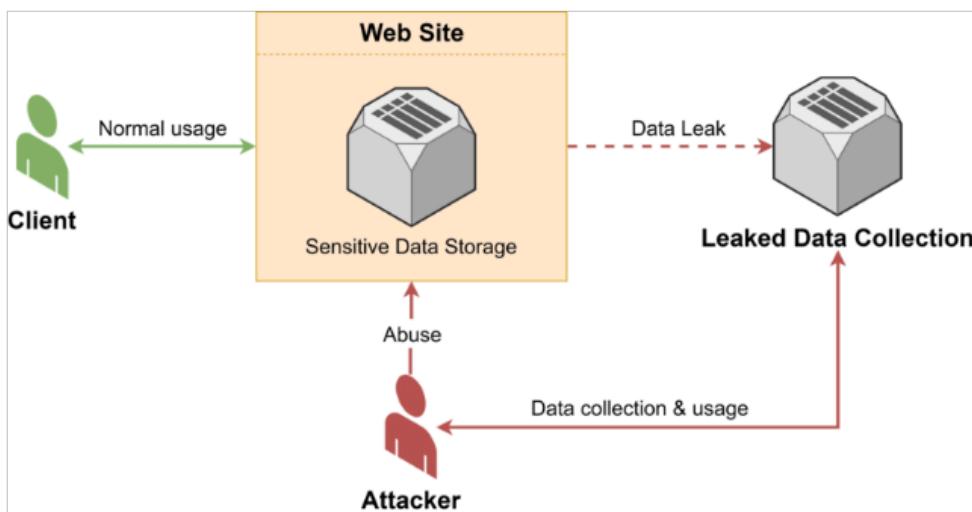
Hình 1.1 — Bối cảnh ứng dụng mạng xã hội và nhu cầu kết nối theo tính cách

Dữ liệu tính cách có thể được suy diễn từ hành vi số hoặc từ bài trắc nghiệm, và thường được xem là dữ liệu nhạy cảm vì nó liên quan trực tiếp đến xu hướng tâm lý và hành vi của người dùng. Nhiều nghiên cứu chỉ ra rằng đặc điểm tính cách có thể dự đoán từ dữ liệu số và có mức độ ổn định cao [3]. Đồng thời, các đặc điểm này có thể bị khai thác để tác động đến hành vi, ví dụ trong các kịch bản thao túng nội dung hoặc quảng cáo cá nhân hóa quá mức [4]. Việc thu thập và lưu trữ tập trung vì thế cần được xem xét cẩn trọng về quyền riêng tư.

Trong những năm gần đây, nhiều nền tảng lớn liên tục bị cơ quan quản lý chỉ trích và xử phạt vì vi phạm quyền riêng tư. Ví dụ, FTC đã áp mức phạt 5 tỉ USD với Facebook vì các vi phạm về dữ liệu cá nhân [5]. Ở châu Âu, CNIL áp phạt Google vì thiếu minh bạch và không có cơ sở pháp lý đầy đủ cho việc xử lý dữ liệu [6]. Các vụ việc này cho thấy áp lực pháp lý ngày càng tăng đối với những hệ thống thu thập dữ liệu người dùng quy mô lớn. Trong bối cảnh đó, việc thiết kế một quy trình (pipeline) có cơ chế bảo vệ dữ liệu ngay từ đầu là nhu cầu thực tế, không chỉ là lựa chọn kỹ thuật.

Trong bối cảnh đó, đề tài đặt ra yêu cầu bảo vệ dữ liệu tính cách ở mức tương tự như các loại dữ liệu nhạy cảm khác (tin nhắn, mật khẩu). Thay vì để dữ liệu gốc tồn tại dạng văn bản thuần (plaintext) trên máy chủ, hệ thống cần có cơ chế chuyển đổi

và mã hoá để giảm thiểu rủi ro rò rỉ. **Hình 1.2** mô tả các rủi ro chính khi xử lý dữ liệu tính cách theo mô hình tập trung.



Hình 1.2 — Rủi ro khi xử lý dữ liệu tính cách theo mô hình tập trung

1.2 Mục tiêu, phạm vi và cách tiếp cận

Mục tiêu của đề tài là xây dựng một quy trình giới thiệu và bảo vệ dữ liệu tính cách, trong đó dữ liệu gốc được xử lý trên thiết bị, chuyển sang biểu diễn gọn hơn, và chỉ lưu trữ trên máy chủ dưới dạng mã hoá. Bên cạnh đó, hệ thống vẫn phải giữ khả năng so khớp và giới thiệu người dùng một cách hiệu quả.

Các mục tiêu chính bao gồm ba hướng then chốt: (1) xây dựng cơ chế chuyển đổi điểm Big Five sang không gian đặc trưng nhỏ gọn bằng Phân tích Thành phần chính (Principal Component Analysis - PCA) với 4 chiều (PCA-4); (2) thiết kế cơ chế mã hoá theo Chuẩn mã hóa tiên tiến ở chế độ Galois/Counter (AES-GCM) để bảo vệ dữ liệu tính cách khi lưu trữ; và (3) duy trì khả năng so khớp dựa trên độ tương đồng cosine (cosine similarity) để phục vụ quy trình giới thiệu.

Đề tài tập trung vào khía cạnh chuyển đổi dữ liệu và bảo mật, không đi sâu vào triển khai giao diện hay tối ưu hóa trải nghiệm người dùng. Phạm vi hệ thống vì vậy được giới hạn ở việc xử lý trên thiết bị người dùng (chấm điểm Big Five và chuyển đổi PCA-4), một hàm thực thi biên (Edge Function) chịu trách nhiệm mã hoá và giải

mã bằng AES-GCM, và cơ sở dữ liệu lưu trữ vector PCA cùng dữ liệu đã mã hoá (ciphertext) thay vì dữ liệu thô.

Bài toán đặt ra là chuyển đổi vector Big Five chiều thành biểu diễn nhỏ gọn nhưng vẫn giữ được tính phân biệt đủ cao cho việc so khớp. Có nhiều hướng thay thế như dùng mô hình nhúng ngữ nghĩa hoặc học sâu, nhưng các hướng này thường yêu cầu dữ liệu huấn luyện lớn hơn và khó giải thích.

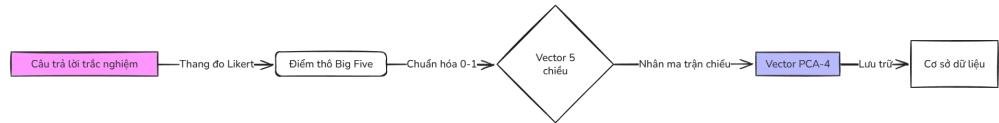
Trong đề tài, PCA được chọn vì Big Five là mô hình tâm lý chuẩn hóa, đã có dữ liệu công khai quy mô lớn và ổn định theo quốc gia [1], [2]. PCA cho phép giảm chiều mà vẫn giữ được phần lớn phương sai. Kết quả từ notebook thực nghiệm cho thấy PCA-4 giữ khoảng hơn 90% phương sai của dữ liệu gốc, trong khi PCA-2 hoặc PCA-3 mất đáng kể thông tin [7]. [Hình 1.3](#) mô tả quy trình giới thiệu Big Five sang PCA-4.

Một điểm quan trọng là tính cách khác với ngôn ngữ tự nhiên. Đối với ngôn ngữ, việc nhúng văn bản thường dựa trên các mô hình ngữ nghĩa (semantic model) lớn vì nội dung có tính mơ hồ, đa nghĩa và phụ thuộc ngữ cảnh. Trong khi đó, Big Five đã là một mô hình tâm lý chuẩn hóa, có cấu trúc dữ liệu rõ ràng và nguồn dữ liệu đủ lớn. Vì vậy PCA và độ tương đồng cosine phù hợp hơn cho phần tính cách, giúp giữ tính diễn giải và ổn định. Các mô hình ngữ nghĩa vẫn được sử dụng cho phần sở thích (hobbies), nơi dữ liệu là văn bản tự do và cần ánh xạ ngữ nghĩa.

Nói cách khác, đề tài không tìm cách “học lại” tính cách bằng mô hình ngôn ngữ, mà tận dụng một hệ đo đã có sẵn trong tâm lý học. PCA chỉ là bước nén và sắp xếp lại thông tin, không thay đổi ý nghĩa gốc của Big Five. Điều này giúp tránh lệch chuẩn khi dùng mô hình học sâu khó giải thích, đồng thời giảm phụ thuộc vào dữ liệu huấn luyện nội bộ. Tính cách vì thế được xử lý như một tín hiệu có cấu trúc, còn ngôn ngữ được xử lý như tín hiệu mở.

Ở cấp độ thu thập, hệ thống sử dụng bộ câu hỏi tính cách lớn hơn, sau đó chọn ngẫu nhiên 25 câu cho mỗi lượt làm bài. Mỗi 5 câu đại diện cho một nhóm đặc điểm (trait), và điểm số được cộng hoặc trừ tùy theo hướng câu hỏi. Mô hình không phụ thuộc nội dung câu hỏi mà chỉ quan tâm đến hướng (key) và trait tương ứng. Cách

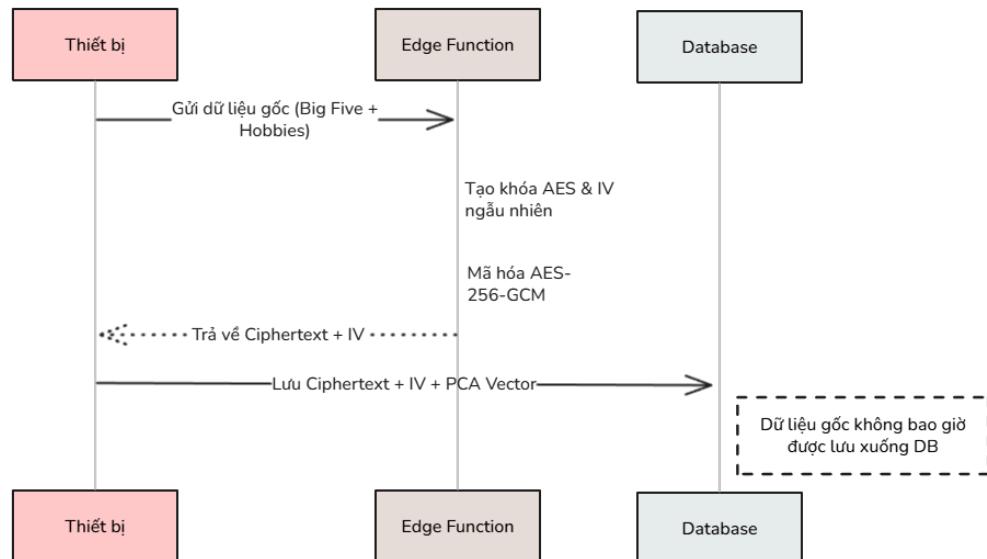
tiếp cận này giúp duy trì tính nhất quán của thang đo trong khi giảm tài thời gian trả lời cho người dùng.



Hình 1.3 — Quy trình giới thiệu Big Five sang vector PCA-4

PCA không phải cơ chế bảo mật. Các thành phần PCA có thể bị suy ngược gần đúng nếu biết tham số mô hình. Vì vậy, dữ liệu gốc vẫn cần được mã hoá. Trong số các phương án, AES-256-GCM được chọn vì phù hợp với khối lượng dữ liệu (payload) nhỏ, tốc độ cao và có tính toàn vẹn dữ liệu (integrity) nhờ GCM [8]. So với RSA hoặc Bcrypt, AES-GCM ít tốn tài nguyên hơn cho dữ liệu dạng JSON, và phù hợp với mô hình hàm thực thi bên.

Trong hệ thống, khóa AES chỉ nằm ở phía máy chủ (Edge Function). Thiết bị người dùng không giữ khóa, nhằm tránh nguy cơ bị trích xuất từ ứng dụng và vẫn cho phép khôi phục dữ liệu khi đăng nhập lại trên thiết bị khác. [Hình 1.4](#) mô tả luồng mã hoá và lưu trữ dữ liệu tính cách.



Hình 1.4 — Luồng mã hoá AES-GCM và lưu trữ dữ liệu tính cách

1.3 Đóng góp chính

Đề tài xây dựng quy trình chuyển đổi Big Five sang PCA-4 chạy trên thiết bị, đảm bảo giảm kích thước dữ liệu nhưng vẫn giữ phần lớn thông tin. Hệ số PCA được huấn luyện trên tập dữ liệu công khai quy mô lớn, giúp kết quả có tính ổn định và tái lập.

Đề tài đề xuất cơ chế mã hoá AES-256-GCM qua Edge Function, đảm bảo dữ liệu gốc không lưu dưới dạng văn bản thuần trên cơ sở dữ liệu. Cách tiếp cận này cân bằng giữa khả năng so khớp và yêu cầu bảo mật dữ liệu nhạy cảm.

Toàn bộ mã nguồn cốt lõi của ứng dụng, bao gồm quy trình xử lý trên thiết bị, các hàm thực thi biên và cấu trúc cơ sở dữ liệu, được cung cấp đính kèm cùng báo cáo này. Đây là nguồn tài liệu minh chứng cho quá trình hiện thực, đồng thời phục vụ công tác thẩm định và đối soát kết quả của Hội đồng.

1.4 Cấu trúc của báo cáo

Phần còn lại của báo cáo được trình bày theo cấu trúc mới: [Phần 2](#) mô tả tổng quan quy trình hệ thống Twins, làm rõ luồng dữ liệu từ thu thập, xử lý đến khâu giới thiệu người dùng. [Phần 3](#) trình bày đề xuất và triển khai, bao gồm chuyển đổi tính cách bằng PCA-4, cơ chế bảo mật dữ liệu và mô hình hệ giới thiệu lai. [Phần 4](#) trình bày hiện thực và đánh giá với các kịch bản thực nghiệm, kết quả đo lường và phân tích hiệu năng. Cuối cùng, [Phần 5](#) tổng kết các kết quả đạt được và nêu hướng phát triển tiếp theo.

Chương 2 Tông quan hệ thống

Chương này trình bày quy trình (pipeline) tổng thể của hệ thống Twins, theo thứ tự từ thu thập dữ liệu trên thiết bị, chuyển đổi và bảo mật, đến giới thiệu người dùng. Mục tiêu là mô tả rõ các tác nhân tham gia, dữ liệu vào ra ở mỗi bước và cách các điểm số được kết hợp thành một giới thiệu xếp hạng cuối cùng. Các chương sau sẽ đi sâu vào từng thành phần. Trong đó, Chương 3 trình bày đề xuất và triển khai, bao gồm chuyển đổi PCA-4, bảo mật dữ liệu và cơ chế giới thiệu lai. Chương 4 trình bày hiện thực và đánh giá với các thực nghiệm đo lường hiệu năng và chất lượng giới thiệu. Chương 5 tổng kết và nêu hướng phát triển tiếp theo.

Về mặt bài toán thực tế, một hệ thống kết nối người dùng phải đồng thời đáp ứng hai yêu cầu kinh doanh quan trọng: (1) gợi ý đủ chính xác để người dùng cảm thấy “đúng người, đúng nhu cầu”, và (2) bảo vệ dữ liệu nhạy cảm để tạo niềm tin khi sử dụng. Nếu chỉ dựa vào thông tin bề mặt, chất lượng kết nối sẽ thấp; nếu xử lý dữ liệu thô một cách lỏng lẻo, rủi ro rò rỉ sẽ cao.

Vì vậy, quy trình (pipeline) của Twins được thiết kế theo hướng “bảo mật theo thiết kế”: dữ liệu được mã hoá/giải mã có kiểm soát, chỉ dùng các biểu diễn đã mã hoá hoặc đã rút gọn để tính tương đồng, tránh để lộ nội dung gốc. Về mặt kỹ thuật, hệ thống sử dụng encoding (PCA-4 cho tính cách, embedding cho sở thích) và so sánh bằng cosine similarity để đảm bảo tính nhất quán khi đo mức độ tương đồng trong không gian vector, đồng thời vẫn giữ được riêng tư nhờ lớp mã hoá tại Edge Functions.

Quy trình này không chỉ dựa trên tính cách, mà còn kết hợp thêm tín hiệu hành vi (ELO) và tín hiệu ngữ nghĩa (Hobbies). ELO phản ánh mức độ tương tác và sự “hợp nhịp” trong quá trình sử dụng, còn Hobbies cho phép người dùng tự do khai báo sở thích và được chuyển thành vector ngữ nghĩa để so khớp mềm. Nhờ đó, hệ thống có thể vừa giữ được độ ổn định của tính cách, vừa linh hoạt điều chỉnh theo hành vi và nội dung quan tâm thực tế.

Các đoạn sau sẽ lần lượt làm rõ nguồn dữ liệu đầu vào, cấu trúc pipeline, và cách ba tín hiệu này được kết hợp thành điểm xếp hạng cuối cùng, đồng thời duy trì các nguyên tắc bảo mật xuyên suốt toàn bộ quy trình.

2.1 Các nguồn dữ liệu đầu vào

Hệ thống sử dụng tập câu hỏi Big Five lớn, được tổng hợp từ các bộ câu hỏi chuẩn như IPIP 50 và các biến thể đã được công bố rộng rãi [9]. Mỗi lượt làm bài chọn ngẫu nhiên 25 câu từ một tập hợp (pool) 150 câu, trong đó mỗi 5 câu đại diện cho một đặc điểm (trait). Mỗi câu hỏi có hướng cộng hoặc trừ vào trait tương ứng, do đó mô hình không phụ thuộc nội dung câu hỏi mà chỉ phụ thuộc vào hướng (key) và trait của câu hỏi.

Cách lấy mẫu này giúp giảm thời gian làm bài, đồng thời vẫn giữ được cấu trúc cân bằng giữa các trait. Trên thực tế, hệ thống chỉ cần biết hai thông tin cho mỗi câu: thuộc tính trait nào và hướng tính điểm (cộng hay trừ). Nội dung câu hỏi được giữ để bảo vệ quyền riêng tư của người dùng, nhưng không ảnh hưởng đến mô hình chuyển đổi. Chi tiết cách tính điểm từ thang Likert và quy trình chuẩn hóa được trình bày tại [Phần 3.1](#).

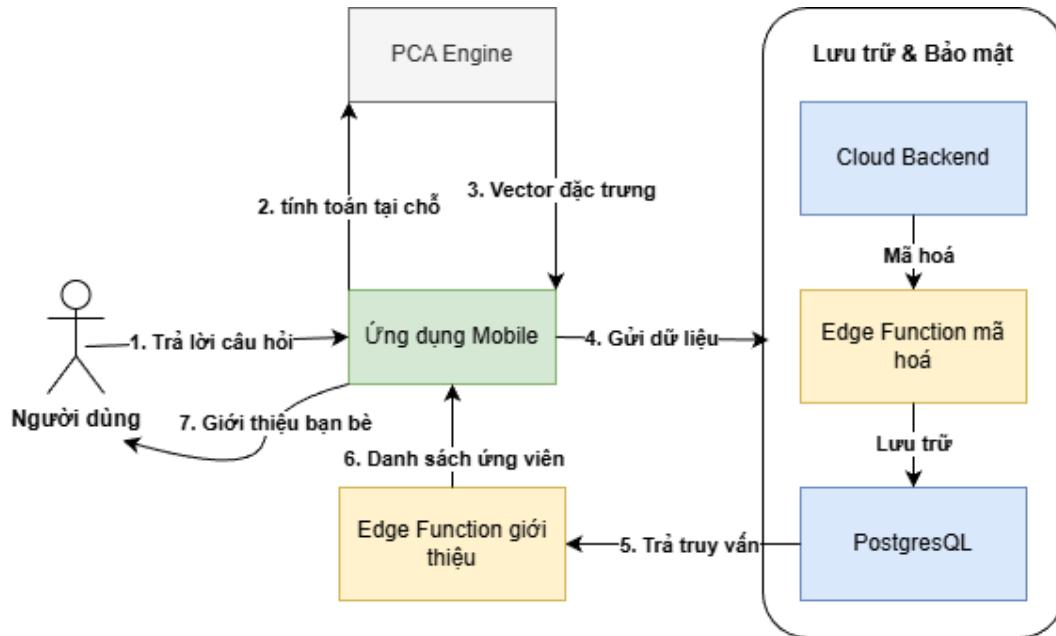
Để huấn luyện PCA, đề tài sử dụng tập dữ liệu Big Five công khai với hơn 300 nghìn mẫu từ nhiều quốc gia [7]. Dữ liệu đã được chuẩn hóa về thang 0-1 cho từng trait, phù hợp cho việc ước lượng các thành phần chính. Các kết quả giải thích phương sai sẽ được nêu ở [Phần 3.1](#). Đây là lợi thế của Big Five: dữ liệu chuẩn hóa, quy mô lớn và đã được sử dụng rộng rãi trong nghiên cứu, nên PCA có thể học được cấu trúc phân bố ổn định.

Bên cạnh đó, dữ liệu tính cách gốc vẫn cần được mã hóa bằng AES-256-GCM thông qua Edge Function trước khi lưu trữ; cơ chế này được mô tả tại [Phần 3.2](#).

Sở thích người dùng được nhập dưới dạng văn bản ngắn. Văn bản này không dùng để lưu trữ trực tiếp, mà được chuyển thành vector 384 chiều thông qua mô hình nhúng ngữ nghĩa (semantic embedding) từ Jina. Lý do dùng phương pháp nhúng là để so khớp nội dung sở thích theo ngữ nghĩa thay vì so khớp từ khóa đơn thuần. Cách làm này cho phép các sở thích có nghĩa gần nhau (ví dụ “chạy bộ” và “jogging”) vẫn được đánh giá tương đồng. Chi tiết quy trình nhúng và luồng mã hóa dữ liệu sở thích sẽ được mô tả ở [Phần 3.3](#).

2.2 Tổng quan quy trình và tác nhân

Hệ thống có ba tác nhân chính: thiết bị người dùng, Edge Function và cơ sở dữ liệu. [Hình 2.1](#) mô tả quy trình tổng thể từ thu thập dữ liệu đến giới thiệu.



Hình 2.1 — Quy trình tổng thể của hệ thống Twins

Các bước chính gồm:

- Thiết bị người dùng trả lời 25 câu hỏi, chấm điểm Big Five và chuẩn hóa về thang 0-1.
- Thiết bị chuyển đổi PCA-4 bằng tham số đã huấn luyện sẵn.
- Thiết bị gửi dữ liệu Big Five gốc tới Edge Function để mã hóa AES-256-GCM.
- Cơ sở dữ liệu lưu trữ pca_dim1..4 và ciphertext (b5_cipher, b5_iv).
- Dữ liệu sở thích được nhúng thành vector 384 chiều, mã hóa, và lưu trữ tương tự.
- Hệ giới thiệu lấy vector PCA, ELO và vector sở thích để tính giới thiệu xếp hạng.

Khi số lượng người dùng tăng lớn, việc so khớp theo tổ hợp từng cặp sẽ làm chi phí tính toán tăng nhanh. Một hướng giảm tải là phân mảnh địa lý (geosharding), tức chia người dùng theo vùng địa lý hoặc cụm vị trí, sau đó ưu tiên so khớp trong cùng một phân mảnh (shard). Cách này phổ biến ở các ứng dụng hẹn hò vì nó giảm số lượng cặp cần so sánh và tăng tốc phản hồi.

Trong đề tài, geosharding được xem là bước tối ưu hóa dài hạn, chưa ưu tiên ở giai đoạn thử nghiệm. Khi lượng người dùng đủ lớn và chi phí tính toán trở thành nút thắt, hệ thống đề xuất bổ sung tầng shard theo vùng để giới hạn không gian tìm kiếm. Điều này không thay đổi công thức giới thiệu, nhưng làm giảm khối lượng tính toán cho mỗi lượt giới thiệu.

2.3 Mô hình giới thiệu và trọng số trong giới thiệu

Vector PCA-4 được dùng để đo tương đồng giữa hai người dùng bằng cosine similarity. Phương pháp này phù hợp vì đo góc giữa hai vector, ít bị ảnh hưởng bởi độ lớn tuyệt đối và ổn định khi dữ liệu đã chuẩn hóa [10]. Công thức cosine similarity sẽ được trình bày chi tiết ở Chương 5.

Hệ thống dùng điểm ELO như một thước đo xã giao, phản ánh mức độ tương tác qua hành vi like và skip. Điểm ELO được cập nhật theo kỳ vọng thắng thua trong mô hình Elo gốc, nhưng được điều chỉnh để phù hợp với ngữ cảnh kết nối xã hội [11]. Quy trình tính toán kỳ vọng và cập nhật điểm số được trình bày tại [Thuật toán 3.4](#) và [Thuật toán 3.3](#). Trong hệ thống:

- Like: cả hai phía tăng nhẹ.
- Skip: chỉ người chủ động skip bị trừ.

Điểm ELO không phải thước đo hấp dẫn tuyệt đối, mà là tín hiệu phụ để gom nhóm người dùng có mức tương tác tương đồng. ELO trong Twins là hệ số ẩn, được cập nhật sau mỗi lần tương tác và bị giới hạn (clamp) trong khoảng 800 đến 2000. Lưu ý rằng cách cập nhật này tạo xu hướng lạm phát điểm ELO theo thời gian, vì lượt “like” làm cả hai phía tăng điểm. Tuy vậy, mục đích chính không phải cạnh tranh, mà là đảm bảo người dùng có mức xã giao gần nhau được ưu tiên gấp nhau hơn.

Trong công thức gốc, kỳ vọng thắng được tính bởi: Sau đó cập nhật theo $R_{a'} = R_a + K(S_a - E_a)$. Trong Twins, kết quả like được coi là một tín hiệu hợp tác nên cả hai phía tăng nhẹ, còn skip chỉ trừ phía chủ động. Cụ thể, với K=12 và được giới hạn trong [800, 2000], quy tắc cập nhật được chi tiết tại [Thuật toán 3.3](#):

Bên cạnh đó, hệ giới thiệu sử dụng hệ số gần nhau ELO để ưu tiên mức xã giao tương đồng, được tính theo công thức tại [Thuật toán 3.5](#): trong đó $\sigma = 400$.

Sở thích người dùng được nhập dưới dạng văn bản ngắn. Văn bản này không dùng để lưu trữ trực tiếp, mà được chuyển thành vector 384 chiều thông qua mô hình nhúng ngữ nghĩa. Cosine similarity được dùng để đo độ gần về sở thích, thay vì so khớp từ khóa. Cách làm này cho phép hai người dùng dùng từ khác nhau nhưng có ý nghĩa gần nhau vẫn được đánh giá cao hơn.

Giới thiệu xếp hạng cuối cùng được tính theo trọng số của PCA, ELO và hobbies dựa trên cấu hình hệ thống, chi tiết tại [Thuật toán 2.1](#):

Thuật toán 2.1 — Thuật toán tính điểm giới thiệu tổng hợp

- **Trường hợp không sử dụng sở thích:**
 - Nếu ELO bật: $S = 0.8 \cdot P + 0.2 \cdot p$
 - Nếu ELO tắt: $S = P$
- **Trường hợp sử dụng sở thích:**
 - Nếu ELO bật: $S = 0.5 \cdot P + 0.2 \cdot p + 0.3 \cdot H$
 - Nếu ELO tắt: $S = 0.55 \cdot P + 0.45 \cdot H$

Trong đó: P là độ tương đồng PCA, p là hệ số gần nhau ELO, H là độ tương đồng sở thích.

Để minh họa, xét ba người dùng A, B, C khi A đang tìm giới thiệu. Giả sử A có PCA tương đồng với B và C gần bằng nhau (ví dụ 0.90), nhưng B có sở thích gần hơn (hobbies 0.85) trong khi C có ELO gần hơn (proximity 1.0 so với 0.7). Trong cấu hình có ELO và hobbies, điểm giới thiệu cuối của B được tính như sau:

$$S_B = 0.5 \cdot 0.90 + 0.2 \cdot 0.70 + 0.3 \cdot 0.85 = 0.845$$

và của C:

$$S_C = 0.5 \cdot 0.90 + 0.2 \cdot 1.00 + 0.3 \cdot 0.55 = 0.815$$

Kết quả là B sẽ đứng trước C trong danh sách giới thiệu. Sơ đồ trọng số tổng hợp được mô tả tại [Hình 2.2](#):



Hình 2.2 — Sơ đồ trọng số tính giới thiệu xếp hạng

2.4 Luồng dữ liệu chi tiết theo tác nhân

Quy trình xử lý dữ liệu được phân chia rõ ràng theo trách nhiệm của từng tác nhân trong hệ thống.

Thiết bị người dùng đóng vai trò là điểm khởi đầu của luồng dữ liệu. Tại đây, ứng dụng thu thập câu trả lời trắc nghiệm, thực hiện chấm điểm Big Five, sau đó chuẩn hóa và áp dụng phép chiếu PCA-4 để tạo ra vector đặc trưng. Dữ liệu Big Five gốc cùng với văn bản sở thích thô sau đó được gửi an toàn đến các Edge Function để xử lý ở bước tiếp theo.

Edge Function hoạt động như một lớp trung gian an toàn, đảm nhận các tác vụ nhạy cảm. Nó nhận dữ liệu từ thiết bị, tiến hành mã hóa cả điểm Big Five và sở thích bằng thuật toán AES-256-GCM, đồng thời gọi dịch vụ ngoài để chuyển đổi văn bản sở thích thành vector nhúng ngữ nghĩa. Sau khi xử lý, hàm sẽ trả về các giá trị đã được mã hóa (ciphertext, iv) và vector nhúng cho thiết bị.

Cơ sở dữ liệu là nơi lưu trữ cuối cùng và được thiết kế theo nguyên tắc “bảo mật theo thiết kế”. Nó chỉ chứa các biểu diễn an toàn của dữ liệu, bao gồm vector PCA-4 (dưới các trường `pca_dim1` đến `pca_dim4`) phục vụ cho việc so khớp, cùng với dữ liệu đã mã hóa và vector khởi tạo (`b5_cipher`, `b5_iv`) cho cả Big Five và sở thích.

Dữ liệu gốc dưới dạng văn bản thuần không bao giờ được lưu trữ trực tiếp trên cơ sở dữ liệu.

Luồng tương tác này được minh họa qua nhật ký hệ thống thực tế tại [Hình 2.3](#):

Edge function Log

```
29 Dec 25 03:16:30 ⓘ INFO [score-crypto] [be447921-1e11-416c-a9a3-5413374d2c9a] encrypt success (2ms)
29 Dec 25 03:16:30 ⓘ INFO [score-crypto] [be447921-1e11-416c-a9a3-5413374d2c9a] mode=encrypt hint=generic_array
29 Dec 25 03:16:30 ⓘ INFO [score-crypto] [be447921-1e11-416c-a9a3-5413374d2c9a] encrypting: ["Anime", "Gardening", "Hiking"]...
29 Dec 25 03:16:30 ⓘ INFO [score-crypto] [be447921-1e11-416c-a9a3-5413374d2c9a] incoming POST
29 Dec 25 03:16:30 ⓘ INFO Listening on http://localhost:9999/
```

Hình 2.3 — Đoạn log tại edge function thể hiện quá trình mã hoá dữ liệu được gửi từ người dùng.

Chương 3 Đề xuất và Triển khai

3.1 Chuyển đổi dữ liệu tính cách (PCA-4)

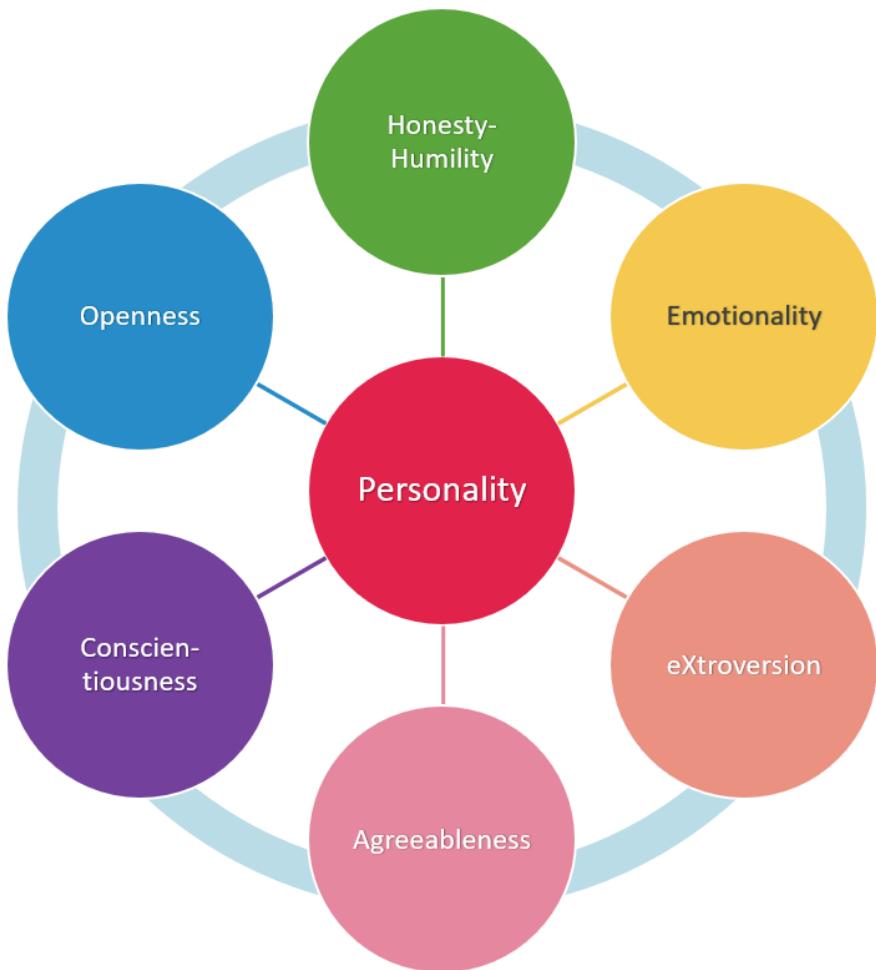
Trong tâm lý học có nhiều khung mô tả tính cách, không có mô hình nào tuyệt đối hoàn hảo. Big Five được sử dụng vì đã có lịch sử nghiên cứu dài, hệ thống câu hỏi chuẩn hóa và dữ liệu công khai phong phú. So với các mô hình khác như MBTI hoặc HEXACO, Big Five có ưu thế về tính tái lập và độ phủ dữ liệu, phù hợp cho bài toán chuyển đổi số liệu quy mô lớn [2], [12]. Do đó, đồ án chấp nhận giới hạn của mô hình nhưng coi Big Five là lựa chọn thực tế nhất để làm nền cho quy trình chuyển đổi dữ liệu.

MBTI phân loại người dùng theo các cặp đối lập, tạo ra 16 nhóm tính cách. Cách biểu diễn này dễ truyền thông nhưng thiên về phân loại rời rạc, trong khi dữ liệu thực tế thường có phân bố liên tục. Với bài toán giới thiệu cần đo mức độ gần nhau, dạng nhãn rời rạc làm giảm khả năng xếp hạng chi tiết và khó phản ánh mức độ “gần” giữa hai cá nhân. MBTI cũng có vấn đề về độ ổn định theo thời gian, nhiều người thay đổi nhóm khi làm lại bài test. Điều này làm cho dữ liệu khó tái lập và khó dùng cho quy trình so khớp dài hạn. Ngoài ra, MBTI ít có dữ liệu mở quy mô lớn theo chuẩn hóa số điểm, nên khó dùng cho chuyển đổi PCA và huấn luyện ổn định. Ví dụ, hai người thuộc nhóm INFP và ENFP có thể khác nhau mạnh về hướng ngoại nhưng vẫn bị xem là hai nhãn rời rạc. [Hình 3.1](#) minh họa cách MBTI chia nhóm tính cách.

ESTJ	ESTP	ESFP	ESFJ
Tj Ambition Sj Discipline Se Experience Te Pragmatism	Sp Spontaneity Tp Inventiveness Se Experience Te Pragmatism	Sp Spontaneity Fp Honesty Se Experience Fe Romantic	Sj Discipline Fj Kindness Se Experience Fe Romantic
ISTJ	ISTP	ISFP	ISFJ
Si History Ti Accuracy Sj Discipline Tj Ambition	Si History Ti Accuracy Sp Spontaneity Tp Inventiveness	Si History Fi Harmony Sp Spontaneity Fp Honesty	Si History Fi Harmony Sj Discipline Fj Kindness
INTJ	INTP	INFP	INFJ
Ni Philosophy Ti Accuracy Nj Vision Tj Ambition	Ni Philosophy Ti Accuracy Np Variation Tp Inventiveness	Ni Philosophy Fi Harmony Np Variation Fp Honesty	Ni Philosophy Fi Harmony Nj Vision Fj Kindness
ENTJ	ENTP	ENFP	ENFJ
Ne Opportunity Te Pragmatism Nj Vision Tj Ambition	Ne Opportunity Te Pragmatism Np Variation Tp Inventiveness	Ne Opportunity Fe Romance Np Variation Fp Honesty	Ne Opportunity Fe Romance Nj Vision Fj Kindness

Hình 3.1 — Minh họa mô hình MBTI và cách phân nhóm tính cách

HEXACO mở rộng Big Five bằng cách thêm yếu tố Trung thực-Khiêm tốn (Honesty-Humility). Mô hình này có giá trị về mặt học thuật, nhưng dữ liệu mở và bộ câu hỏi chuẩn hóa không phổ biến bằng Big Five. Việc thêm một đặc điểm (trait) thứ sáu làm tăng số câu hỏi cần thiết để giữ cân bằng độ tin cậy. Điều này gây áp lực lên trải nghiệm người dùng di động, vì thời gian trả lời dài hơn. Ngoài ra, chuyển đổi từ HEXACO sang dạng PCA sẽ cần dữ liệu huấn luyện riêng, trong khi dữ liệu chuẩn không nhiều bằng Big Five. Ví dụ, nếu chỉ dùng 25 câu, mỗi trait sẽ bị giảm số câu đánh giá, làm tăng nhiều đo lường. Do đó HEXACO được xem là lựa chọn tham khảo hơn là lựa chọn chính cho đồ án. [Hình 3.2](#) minh họa cấu trúc HEXACO.



Hình 3.2 — Minh họa cấu trúc 6 yếu tố của HEXACO

3.1.1 Chuẩn hóa điểm Big Five

Mỗi câu trả lời được chấm theo thang Likert 1–5. Với câu hỏi hướng dương, điểm giữ nguyên thứ tự 1→5. Với câu hỏi hướng âm, điểm được đảo chiều. Sau đó các điểm trong cùng một trait được cộng lại và chuẩn hóa về thang 0–1. Cách chuẩn hóa này giúp các trait có cùng thang đo, phù hợp cho PCA và so khớp cosine.

Sau bước chuẩn hóa, mỗi người dùng có một vector 5 chiều theo thứ tự trait cố định:

$$x = [\text{Extraversion}, \text{Agreeableness}, \text{Conscientiousness}, \text{Emotional Stability}, \text{Intellect}]$$

Ví dụ một người dùng có thể có:

```
x = [0.68, 0.55, 0.72, 0.60, 0.47]
```

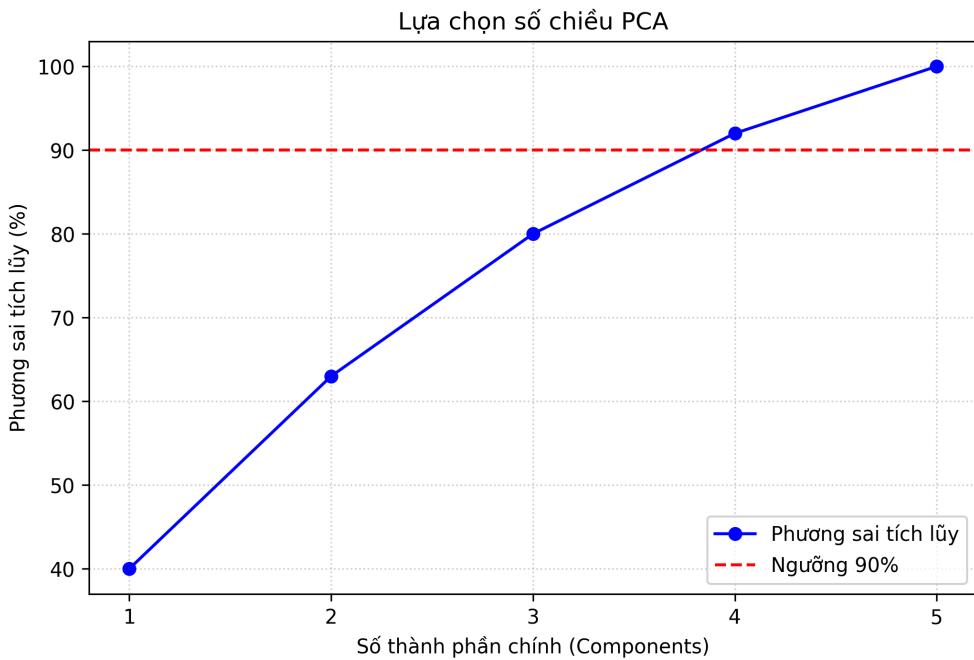
Đây là dạng dữ liệu đầu vào cho bước PCA.

Chuẩn hóa đưa dữ liệu Big Five về cùng thang đo, giúp mỗi trait đóng góp cân bằng khi so khớp và khi học PCA. Tuy vậy, chuẩn hóa không giải quyết vấn đề dữ thừa thông tin giữa các trait. PCA được dùng để rút gọn chiều và tách các trục phương sai lớn nhất. Trong khi PCA-2 hoặc PCA-3 làm mất đáng kể thông tin, PCA-4 là điểm cân bằng tối ưu: giảm chiều từ 5 xuống 4 nhưng vẫn giữ phần lớn phương sai, giúp hệ giới thiệu hoạt động ổn định khi đo độ tương đồng cosine.

3.1.2 Đề xuất PCA-4

Đồ án đề xuất PCA-4 như mức giảm chiều tối ưu cho Big Five trong bối cảnh giới thiệu bạn bè. Giảm từ 5 xuống 4 chiều giúp tiết kiệm lưu trữ mà vẫn giữ phần lớn cấu trúc dữ liệu. PCA-4 cũng là dạng biểu diễn dễ triển khai trên thiết bị với phép nhân ma trận thuận. Mức giảm nhẹ này giúp hạn chế rủi ro mất thông tin so với PCA-2 hoặc PCA-3. Ngoài ra, PCA-4 giữ được tính diễn giải tương đối, phù hợp với việc so sánh độ tương đồng cosine ổn định. Một điểm quan trọng là vector Big Five 5 chiều tuy đầy đủ về mặt mô tả, nhưng chưa phải là định dạng tối ưu cho bài toán so khớp. PCA-4 đưa dữ liệu về không gian đã được chiếu, giúp các thành phần trở nên độc lập hơn và phù hợp với phép đo cosine similarity. Đây cũng là cách tiếp cận phổ biến trong ML/NLP khi cần so sánh nhanh các vector đặc trưng, và trong ngữ cảnh ứng dụng Twins, phép chiếu này giúp điểm tương đồng ổn định hơn khi so sánh trên tập ứng viên lớn.

[Hình 3.3](#) trình bày một minh họa quyết định chọn PCA-4 dựa trên phương sai.



Hình 3.3 — Minh họa tiêu chí lựa chọn PCA-4

Trong notebook thực nghiệm, PCA-2 chỉ giữ khoảng 63% phương sai, PCA-3 khoảng 80%, trong khi PCA-4 giữ hơn 90% phương sai dữ liệu gốc. Sự chênh lệch này ảnh hưởng trực tiếp đến khả năng phân biệt giữa các người dùng khi so khớp. Vì vậy PCA-4 được chọn để giảm mất thông tin mà vẫn đảm bảo kích thước nhỏ gọn.

3.1.3 Huấn luyện và Trích xuất tham số PCA

Quá trình huấn luyện được thực hiện trong môi trường Jupyter Notebook (`model/pca_evaluator.ipynb`) sử dụng thư viện `scikit-learn` của Python. Cụ thể, lớp `sklearn.decomposition.PCA` được dùng để thực hiện các phép tính. Dữ liệu đầu vào là bộ `big_five_scores.csv` với 300,313 mẫu [7], đảm bảo số lượng mẫu đủ lớn để các thành phần chính được tính toán một cách ổn định và đáng tin cậy.

Phân tích Dữ liệu Khám phá (EDA) trong notebook cho thấy chênh lệch trung bình giữa các quốc gia tồn tại nhưng không đủ lớn để cần một mô hình riêng theo vùng. Do đó, PCA được huấn luyện trên toàn bộ tập dữ liệu để nắm bắt phương sai tổng thể.

Một điểm quan trọng cần lưu ý là trong quy trình này, toàn bộ dữ liệu đã được sử dụng để huấn luyện mô hình PCA mà không cần phân chia thành tập huấn luyện (train) và tập kiểm thử (test). Lý do là vì bài toán của đồ án không phải là một bài toán dự đoán (học có giám sát) mà là một bài toán biến đổi dữ liệu (học không giám sát). PCA là một phép toán xác định, chỉ cần hai yếu tố cốt lõi từ tập dữ liệu lớn: vector trung bình và ma trận các thành phần chính. Do đó, việc “fit” trên toàn bộ dữ liệu nhằm trích xuất các tham số này là đủ, thay vì cần đánh giá sai số dự đoán như các mô hình hồi quy hoặc phân loại.

Mục tiêu của PCA là tìm ra cấu trúc tiềm ẩn và các hướng phương sai lớn nhất của **toàn bộ** phân bố dữ liệu. Việc chia nhỏ dữ liệu sẽ khiến PCA chỉ học được cấu trúc của một phần dữ liệu, dẫn đến các vector thành phần chính được tạo ra có thể không đại diện chính xác cho toàn bộ không gian dữ liệu. Vì vậy, để có được một phép biến đổi ổn định và tổng quát nhất, việc `fit` PCA trên toàn bộ tập dữ liệu là phương pháp luận chính xác cho bài toán này.

Sau khi quá trình huấn luyện hoàn tất bằng phương thức `.fit()`, đối tượng PCA từ `scikit-learn` sẽ chứa các tham số toán học cần thiết cho việc biến đổi. Hai thuộc tính quan trọng nhất được trích xuất là:

- `pca.mean_`: Đây là một vector chứa giá trị trung bình của mỗi chiều (mỗi đặc trưng tính cách) trên toàn bộ tập dữ liệu. Về mặt hình học, nó đại diện cho “tâm” của đám mây dữ liệu. Phép biến đổi PCA bắt đầu bằng việc dịch chuyển toàn bộ dữ liệu sao cho tâm này trở thành gốc tọa độ mới.
- `pca.components_`: Đây là một ma trận chứa các vector thành phần chính. Mỗi vector là một **unit vector** (đã được chuẩn hóa với độ dài bằng 1) và chỉ ra một hướng trong không gian dữ liệu. Các vector này trực giao với nhau và được sắp xếp theo thứ tự giảm dần của lượng phương sai mà chúng nắm giữ. Đây chính là các “trục” của hệ tọa độ mới sau khi giảm chiều.

Các giá trị này sau đó được sử dụng để triển khai lại logic biến đổi trên thiết bị di động.

PCA thực hiện phép chiếu tuyến tính trên dữ liệu đã được trừ đi giá trị trung bình. Với vector đầu vào x (dài 5), ta có:

$$z = (x - \mu) \times W^T \quad (3.1)$$

trong đó μ là vector trung bình (mean) và W là ma trận chứa các thành phần chính (components) [13]. Vector z là PCA-4 và được lưu dưới dạng 4 chiều. [Thuật toán 3.1](#) mô tả phép chiếu và định dạng đầu ra.

Thuật toán 3.1 — Phép chiếu PCA giảm chiều dữ liệu tính cách

$$z = (x - \mu) \times W = (x_1 \ x_2 \ x_3 \ x_4 \ x_5) \times \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} & w_{1,4} \\ w_{2,1} & w_{2,2} & w_{2,3} & w_{2,4} \\ \vdots & \vdots & \vdots & \vdots \\ w_{5,1} & w_{5,2} & w_{5,3} & w_{5,4} \end{pmatrix} = (z_1 \ z_2 \ z_3 \ z_4)$$

3.1.4 Triển khai PCA trên thiết bị

Mặc dù notebook đã xuất ra một mô hình `pca_evaluator_4d.tflite`, việc tích hợp trực tiếp một mô hình TensorFlow Lite vào ứng dụng React Native (sử dụng Expo) có thể gặp một số thách thức về thư viện và khả năng tương thích, đòi hỏi các thành phần native phức tạp.

Để đảm bảo hiệu suất, giảm sự phụ thuộc vào các thư viện native và tăng tính linh hoạt, đồ án đã chọn một phương pháp hiệu quả hơn: trích xuất trực tiếp các tham số toán học (`mean_` và `components_`) từ đối tượng PCA đã huấn luyện. Các giá trị này sau đó được lưu trữ cố định trong mã nguồn TypeScript của ứng dụng. Một hàm riêng sẽ thực hiện lại phép biến đổi PCA bằng các phép toán cơ bản. Cách tiếp cận này hoàn toàn tương đương về mặt toán học với việc chạy model TFLite nhưng lại đơn giản, minh bạch và dễ bảo trì hơn trong môi trường Expo.

Kết quả PCA-4 được lưu dưới dạng 4 trường số trong cơ sở dữ liệu: `pca_dim1` đến `pca_dim4`. Việc lưu trữ các giá trị này dưới dạng số thực cho phép máy chủ thực hiện các phép tính tương đồng cosine một cách hiệu quả trong quá trình gợi ý người dùng.

Do logic tính điểm Big Five (chuyển đổi 50 câu trả lời thành 5 điểm số) được triển khai lại bằng TypeScript trên ứng dụng, một bước kiểm chứng là tối quan trọng để đảm bảo tính nhất quán với logic gốc trong notebook Python. Vì lý do này, một script kiểm tra tự động (`scripts/score_verifier.ts`) đã được xây dựng.

Script này thực thi một loạt các kịch bản kiểm thử được định nghĩa trước, bao gồm các trường hợp biên (ví dụ: tất cả câu trả lời là “trung lập”) và một kịch bản sử dụng dữ liệu giả lập giống hệt trong notebook. Nó so sánh kết quả tính toán của logic TypeScript với kết quả kỳ vọng. Bằng cách này, đồ án đảm bảo rằng dữ liệu đầu vào cho bước chiếu PCA trên thiết bị luôn chính xác và tương đương với môi trường huấn luyện, giúp toàn bộ quy trình có thể tái lập và đáng tin cậy. Một minh họa về cấu trúc của các kịch bản kiểm thử này được trình bày trong Phụ lục.

PCA là phép biến đổi tuyến tính, có thể giải thích và kiểm soát. Các lựa chọn thay thế như nhúng vector học sâu hoặc nhúng ngữ nghĩa (semantic embedding) không phù hợp vì dữ liệu tính cách đã có cấu trúc rõ ràng và ít phụ thuộc ngôn ngữ. Ngoài ra, PCA giúp duy trì tính ổn định giữa các phiên bản, tránh lệch kết quả do thay đổi mô hình.

3.2 Bảo mật và mã hóa dữ liệu

đổi và mã hóa trước khi lưu trữ, cùng với lý do lựa chọn cơ chế AES-256-GCM. Trọng tâm là luồng dữ liệu và các tác nhân, không đi sâu vào mã nguồn chi tiết.

3.2.1 Tổng quan về cơ chế AES-GCM

AES là thuật toán mã hóa đối xứng khối, hoạt động trên các khối dữ liệu cố định và cần một khóa chung cho cả quá trình mã hóa lẫn giải mã. Chế độ GCM (Galois/Counter Mode) kết hợp giữa mã hóa dạng bộ đếm (counter mode) và cơ chế xác thực dữ liệu.

Thuật toán 3.2 — Quy trình mã hoá và xác thực dữ liệu bằng AES-256-GCM

1. **Khởi tạo:** Tạo khóa bí mật K và vector khởi tạo ngẫu nhiên IV .
2. **Mã hoá:** $C = E_{K(IV, P)}$, trong đó P là văn bản thuần, C là văn bản mã hoá.
3. **Xác thực:** Sinh thẻ xác thực T dựa trên K , IV và C .
4. **Lưu trữ:** Lưu cặp (C, IV, T) vào cơ sở dữ liệu.
5. **Giải mã:** Kiểm tra T trước khi khôi phục $P = D_{K(IV, C)}$.

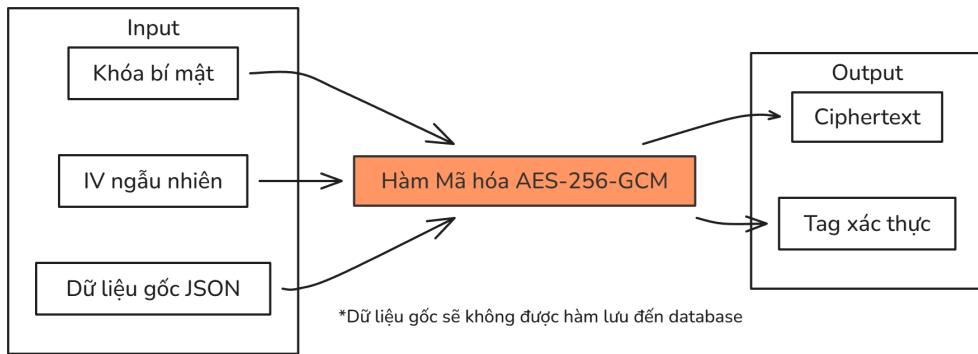
Nhờ đó, ngoài dữ liệu đã mã hoá (ciphertext), hệ thống còn có thể kiểm tra tính toàn vẹn (integrity) của dữ liệu [8].

Trong ngữ cảnh dữ liệu tính cách, yếu tố này rất quan trọng để đảm bảo dữ liệu không bị thay đổi trái phép mà không bị phát hiện.

Một phiên làm việc AES-GCM tạo ra thêm thẻ xác thực (authentication tag), giúp phát hiện bất kỳ sự thay đổi nào đối với dữ liệu hoặc vector khởi tạo (Initialization Vector - IV). Nếu thẻ xác thực không khớp, dữ liệu sẽ bị từ chối giải mã. Cơ chế này làm giảm nguy cơ người dùng nhận phải dữ liệu sai lệch hoặc đã bị chỉnh sửa khi truyền qua mạng. Với dữ liệu nhạy cảm như tính cách và sở thích, việc đảm bảo tính toàn vẹn quan trọng không kém việc giữ bí mật. Vì vậy, AES-GCM phù hợp hơn các chế độ chỉ mã hoá mà không đi kèm xác thực.

Đầu vào bao gồm dữ liệu gốc (dưới dạng JSON chứa điểm Big Five hoặc danh sách sở thích), khóa bí mật, và một IV ngẫu nhiên. Đầu ra bao gồm dữ liệu đã mã hoá (ciphertext) và IV tương ứng. Trong triển khai của đề tài, IV được lưu trữ riêng trong cơ sở dữ liệu để phục vụ quá trình giải mã sau này. [Hình 3.4](#) mô tả cấu trúc đầu vào và đầu ra của quy trình này.

Việc lưu trữ thẻ xác thực đi kèm ciphertext cho phép hệ thống kiểm tra tính toàn vẹn ngay tại thời điểm giải mã. Nếu phát hiện sai lệch, hệ thống sẽ từ chối giải mã và ghi nhận lỗi, ngăn chặn việc trả về dữ liệu sai. Cách lưu trữ này bảo vệ dữ liệu cá nhân khỏi các thay đổi ngầm ở cấp độ cơ sở dữ liệu hoặc trong quá trình truyền tải.

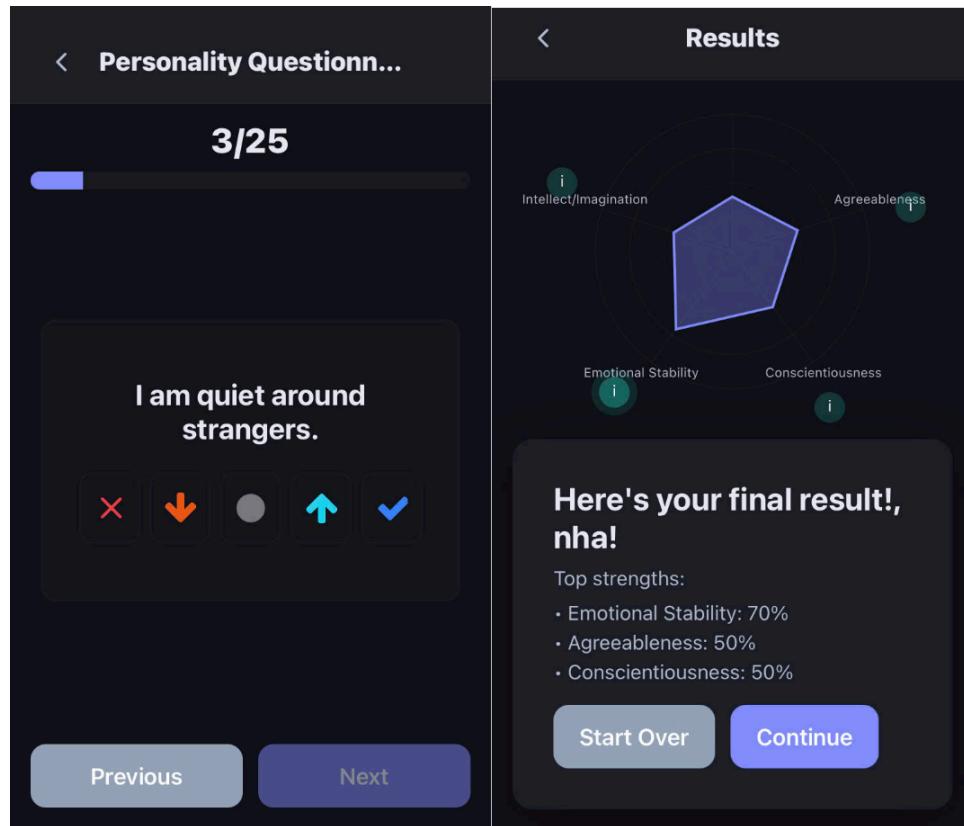


Hình 3.4 — Định dạng đầu vào/đầu ra của AES-GCM

3.2.2 Dữ liệu đầu vào từ góc nhìn người dùng

Người dùng thực hiện bộ câu hỏi tính cách gồm 25 câu hỏi trong một lượt. Các câu trả lời này được xem là dữ liệu nhạy cảm vì có thể dùng để suy diễn đặc trưng tâm lý. Ngay khi người dùng hoàn tất, hệ thống chỉ lưu lại các điểm số đã được tổng hợp theo mô hình Big Five, không lưu trữ câu trả lời gốc cho từng câu hỏi. Việc này giúp giảm thiểu rủi ro rò rỉ dữ liệu thô và hạn chế khả năng định danh gián tiếp.

[Hình 3.5](#) minh họa bố trí giao diện và vị trí bước tổng hợp điểm trong luồng ứng dụng.



Hình 3.5 — Luồng giao diện và vị trí tổng hợp điểm Big Five

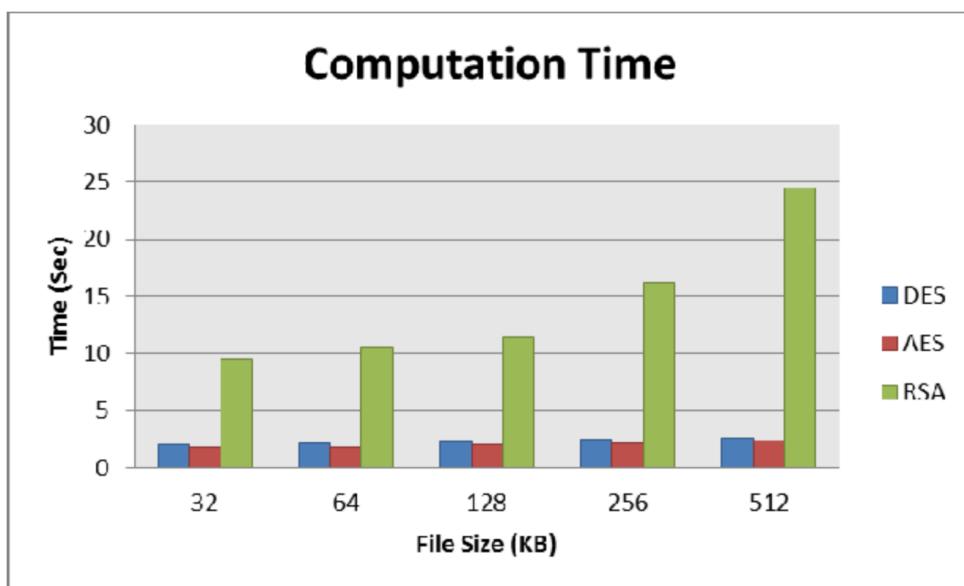
Sau khi tổng hợp, điểm Big Five được chuẩn hóa và chuyển đổi sang không gian PCA-4 ngay trên thiết bị người dùng. Kết quả PCA là dữ liệu đã giảm chiều, đủ cho mục đích so khớp nhưng không thay thế hoàn toàn được dữ liệu thô. Tuy nhiên, vì PCA là phép biến đổi tuyến tính, thông tin gốc vẫn có thể bị suy ngược gần đúng nếu biết tham số mô hình. Do đó, dữ liệu gốc vẫn cần được mã hóa trước khi lưu trữ.

3.2.3 Mã hóa dữ liệu bằng AES-256-GCM

Đề tài đề xuất sử dụng AES-256-GCM làm cơ chế mã hóa chính cho dữ liệu tính cách và sở thích. Lý do là dữ liệu có kích thước nhỏ, yêu cầu tốc độ xử lý nhanh và cần khả năng giải mã để hiển thị lại trên giao diện người dùng. AES-GCM đáp ứng tốt ba yêu cầu: tốc độ, xác thực và dễ dàng triển khai trên các hàm thực thi biên (Edge Function). Cơ chế này cũng cho phép lưu trữ IV riêng biệt để tái tạo dữ liệu khi người dùng đăng nhập lại. Trong phạm vi khóa luận, AES-GCM là lựa chọn tối ưu để cân bằng giữa bảo mật và khả năng vận hành thực tế.

AES-GCM được lựa chọn vì phù hợp với các gói dữ liệu (payload) nhỏ, tốc độ cao, và tích hợp sẵn cơ chế xác thực dữ liệu (integrity) cùng lúc với mã hoá [8]. So với RSA hoặc Bcrypt, AES-GCM tiêu tốn ít tài nguyên hơn khi mã hoá các chuỗi JSON ngắn, và dễ dàng tích hợp trong môi trường Edge Function.

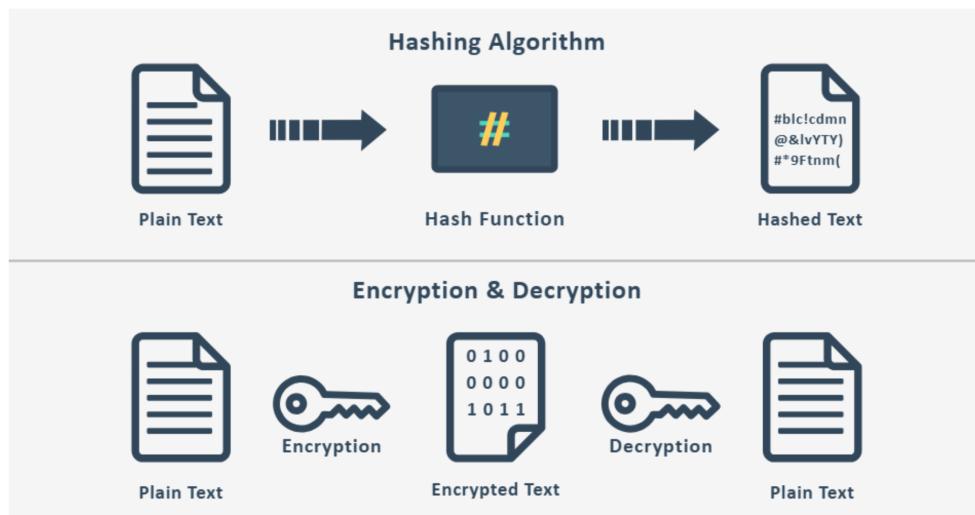
RSA là thuật toán mã hoá bắt đối xứng, thường dùng để trao đổi khóa hoặc ký số [14]. Trong bối cảnh dữ liệu tính cách, RSA không phù hợp để mã hoá trực tiếp dữ liệu vì chi phí tính toán lớn và giới hạn về kích thước dữ liệu đầu vào. Nếu sử dụng RSA cho mỗi lần cập nhật hồ sơ, hệ thống sẽ gặp vấn đề về độ trễ và khó mở rộng trên thiết bị di động. Ngoài ra, RSA thường đi kèm các cơ chế đệm (padding) phức tạp, dễ phát sinh lỗi nếu không được triển khai cẩn trọng. Vì vậy, RSA được xem là phương án thay thế nhưng không phù hợp làm cơ chế mã hoá chính cho dữ liệu người dùng, như minh họa tại [Hình 3.6](#). Ví dụ, việc mã hoá một gói tin JSON nhỏ bằng RSA đòi hỏi nhiều bước xử lý đệm và tách khỏi, gây chậm trễ đáng kể khi người dùng cập nhật hồ sơ liên tục.



Hình 3.6 — Ví dụ chi phí tính toán khi dùng RSA cho payload nhỏ

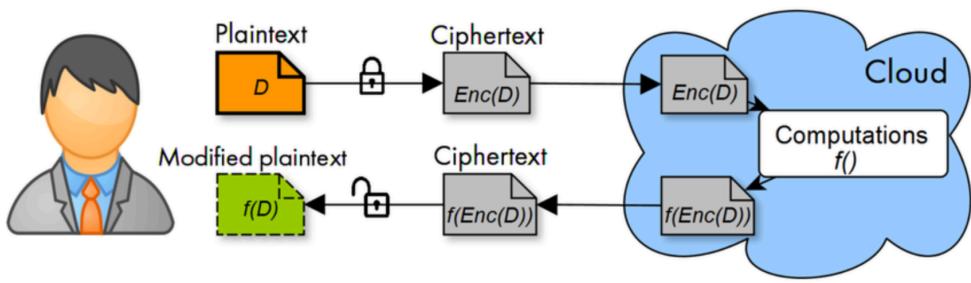
Bcrypt và Scrypt là các hàm băm mật khẩu (password hashing function) [15]. Ưu điểm của chúng là làm chậm các cuộc tấn công dò khóa (brute-force), nhưng nhược điểm là dữ liệu sau khi băm không thể giải mã để lấy lại nội dung gốc. Trong hệ thống

Twins, người dùng cần xem lại kết quả tính cách và sở thích của mình, do đó yêu cầu bắt buộc là phải giải mã được dữ liệu. Nếu dùng bcrypt, hệ thống chỉ có thể so khớp chuỗi băm mà không thể trả lại dữ liệu gốc cho giao diện ([Hình 3.7](#)). Điều này đi ngược lại yêu cầu về trải nghiệm người dùng và giới hạn chức năng của ứng dụng. Vì vậy, các hàm băm này không phù hợp. Ví dụ, sở thích “chạy bộ” sau khi băm sẽ trở thành một chuỗi ký tự ngẫu nhiên và không thể khôi phục để hiển thị lại là “chạy bộ”.



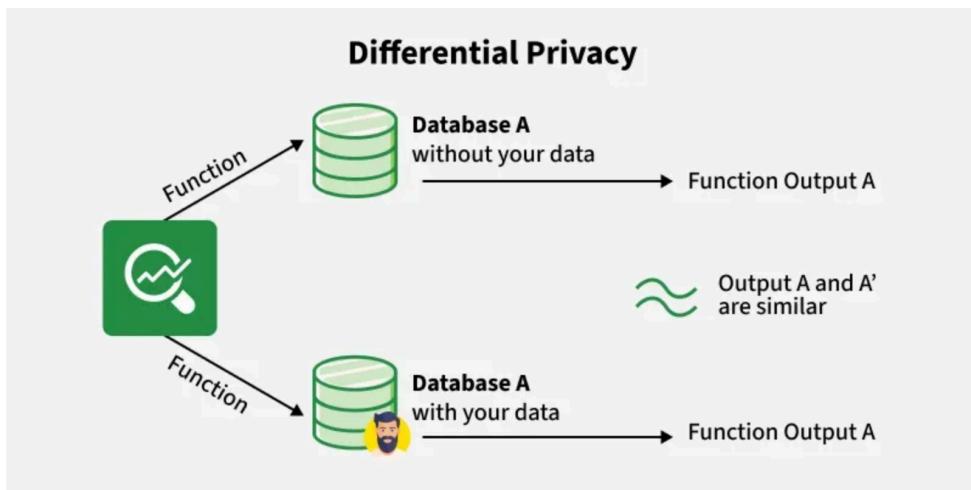
Hình 3.7 — So sánh dữ liệu băm và dữ liệu có thể giải mã

Mã hoá đồng hình (Homomorphic encryption) cho phép thực hiện tính toán trực tiếp trên dữ liệu đã mã hoá mà không cần giải mã [[16](#)]. Đây là hướng đi rất mạnh về bảo mật, nhưng chi phí tính toán cực kỳ cao và việc triển khai rất phức tạp. Với bài toán giới thiệu cần phản hồi nhanh, việc áp dụng mã hoá đồng hình sẽ làm tăng độ trễ hệ thống và đòi hỏi hạ tầng phần cứng đặc biệt ([Hình 3.8](#)). Ngoài ra, mô hình này chưa thực sự cần thiết vì đề tài không yêu cầu tính toán phức tạp trực tiếp trên dữ liệu mã hoá mà chỉ cần lưu trữ an toàn và giải mã khi cần thiết. Do đó, mã hoá đồng hình vượt quá phạm vi thực tế của khóa luận. Ví dụ, một phép so khớp cosine trên dữ liệu mã hoá đồng hình có thể chậm hơn nhiều lần so với trên dữ liệu văn bản thuần, gây trải nghiệm kém mượt mà trên thiết bị di động.



Hình 3.8 — Minh họa độ phức tạp của mã hoá đồng hình

Sự riêng tư biệt lập (Differential privacy) tập trung vào việc ẩn danh hóa khi công bố các số liệu thống kê [17]. Phương pháp này phù hợp cho dữ liệu tổng hợp, nhưng không giải quyết được bài toán lưu trữ và giải mã dữ liệu cho từng cá nhân cụ thể. Nếu chỉ áp dụng sự riêng tư biệt lập, người dùng vẫn cần truy cập vào dữ liệu gốc của chính mình, dẫn tới vấn đề bảo mật vẫn tồn tại ở cấp độ lưu trữ ([Hình 3.9](#)). Trong hệ thống Twins, yêu cầu là bảo vệ dữ liệu của từng người nhưng vẫn cho phép họ xem lại nội dung đó. Vì vậy, sự riêng tư biệt lập được coi như một kỹ thuật bổ trợ chứ không thay thế cho AES-GCM. Ví dụ, nếu cộng thêm nhiều vào điểm Big Five để bảo vệ tính ẩn danh trong thống kê, kết quả giới thiệu cá nhân hóa cho người dùng sẽ bị giảm độ chính xác và khó giải thích.



Hình 3.9 — So sánh sự riêng tư biệt lập và mã hoá dữ liệu cá nhân

Khóa AES chỉ tồn tại ở phía Edge Function (máy chủ biên). Thiết bị người dùng không lưu trữ khóa này, nhằm tránh nguy cơ bị trích xuất từ ứng dụng. Đồng thời, cách thiết kế này cho phép người dùng phục hồi dữ liệu khi đăng nhập lại trên một thiết bị khác. Đây là sự cân bằng hợp lý giữa bảo mật và khả năng khôi phục dữ liệu.

Hình 3.10 minh họa nhật ký (log) của Edge Function cho quá trình mã hoá và giải mã.



The screenshot shows a terminal window titled "Edge function Log" displaying the following log entries:

```

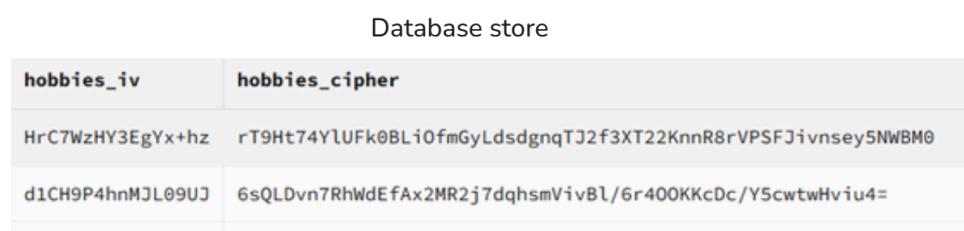
29 Dec 25 03:16:30 ⓘ INFO [score-crypto] [be447921-1e11-416c-a9a3-5413374d2c9a] encrypt success (2ms)
29 Dec 25 03:16:30 ⓘ INFO [score-crypto] [be447921-1e11-416c-a9a3-5413374d2c9a] mode=encrypt hint=generic_array
29 Dec 25 03:16:30 ⓘ INFO [score-crypto] [be447921-1e11-416c-a9a3-5413374d2c9a] encrypting: ["Anime", "Gardening", "Hiking"]...
29 Dec 25 03:16:30 ⓘ INFO [score-crypto] [be447921-1e11-416c-a9a3-5413374d2c9a] incoming POST
29 Dec 25 03:16:30 ⓘ INFO Listening on http://localhost:9999/

```

Hình 3.10 — Nhật ký Edge Function khi mã hoá và giải mã dữ liệu

Cơ sở dữ liệu chỉ lưu trữ dữ liệu đã mã hoá và IV cho Big Five (các trường `b5_cipher`, `b5_iv`). Điều này có nghĩa là quản trị viên cơ sở dữ liệu không thể đọc trực tiếp dữ liệu tính cách dưới dạng văn bản thuần. Dữ liệu chỉ được giải mã khi người dùng đã xác thực thành công và gửi yêu cầu thông qua Edge Function. Cách làm này hạn chế nguy cơ giám sát hàng loạt (mass surveillance) từ bảng dữ liệu chưa mã hoá, đồng thời vẫn đảm bảo tính năng xem lại kết quả cho người dùng.

Hình 3.11 minh họa mẫu dữ liệu đã mã hoá được lưu trong cơ sở dữ liệu.



The screenshot shows a table titled "Database store" with two columns: "hobbies_iv" and "hobbies_cipher". The data is as follows:

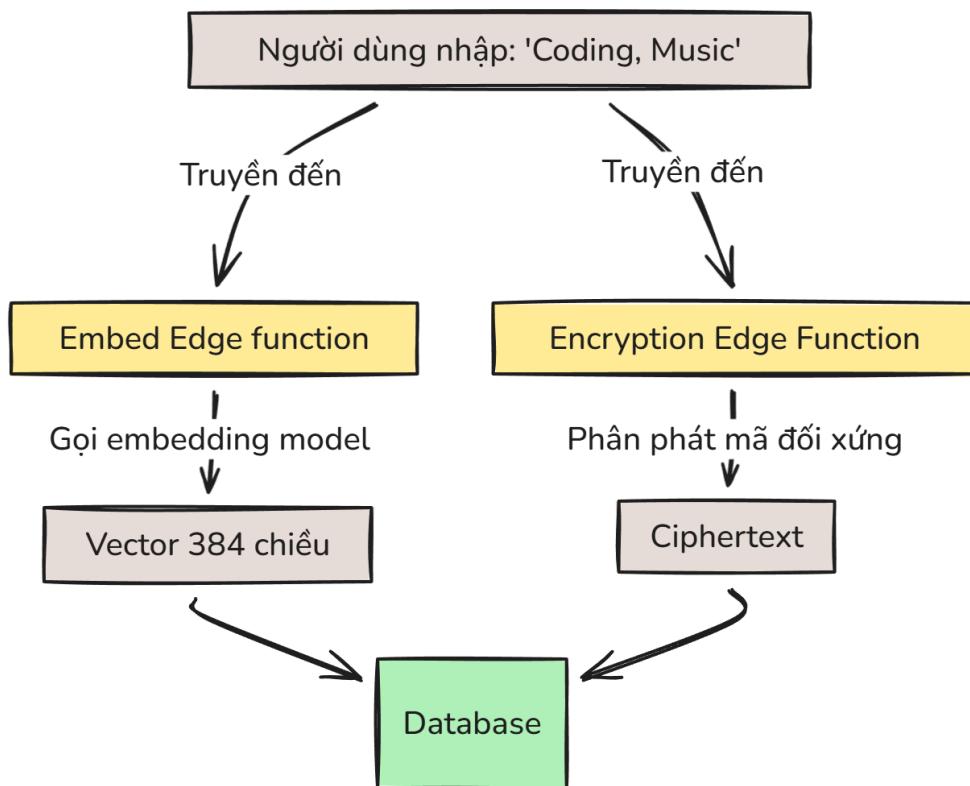
hobbies_iv	hobbies_cipher
HrC7WzHY3EgYx+hz	rT9Ht74YlUFk0BLi0fmGyLdsdgnqTJ2f3XT22KnnR8rVPSFJivnsey5NWBM0
d1CH9P4hnMJL09UJ	6sQLDvn7RhWdEfAx2MR2j7dqhsmvivBl/6r400KKcDc/Y5cwtwHviu4=

Hình 3.11 — Ví dụ dữ liệu đã mã hoá của Big Five trong bảng profiles

3.2.4 Dữ liệu sở thích và mã hóa

Sở thích người dùng được nhập dưới dạng văn bản tự do, sau đó được nhúng thành vector 384 chiều. Dữ liệu văn bản này cũng được mã hóa theo cơ chế AES-GCM tương tự như Big Five. Do đó, giao diện ứng dụng có thể hiển thị lại sở thích sau khi giải mã, nhưng cơ sở dữ liệu hoàn toàn không lưu trữ văn bản thuần.

Hình 3.12 mô tả luồng dữ liệu sở thích từ nhập liệu đến lưu trữ.



Hình 3.12 — Luồng mã hóa dữ liệu sở thích và lưu trữ vector nhúng

3.3 Hệ giới thiệu và cơ chế xếp hạng

(ELO) và sở thích được nhúng vector (embedding hobbies). Đồng thời, chương giải thích vì sao từng tín hiệu vẫn cần thiết, ngay cả khi người dùng đã khai báo sở thích, và đi sâu vào các luận điểm thiết kế đằng sau mỗi thành phần.

Sở thích (interests) phản ánh các chủ đề người dùng quan tâm, nhưng không đủ để mô tả mức độ tương hợp về cách suy nghĩ và hành vi. Hai người cùng thích “chụp ảnh” có thể khác nhau rõ rệt về cách giao tiếp, nhịp sống và mức độ ổn định cảm xúc.

Với các kết nối dài hạn, các khác biệt này thường quan trọng hơn sở thích bè mặt. Hơn nữa, sở thích có thể mang tính thời điểm hoặc thay đổi theo xu hướng, trong khi các đặc điểm tính cách cốt lõi theo mô hình Big Five có xu hướng ổn định hơn nhiều trong suốt cuộc đời của một người trưởng thành.

Vì vậy, tính cách được xác định là **trục ổn định** (stable axis) của hệ giới thiệu, đảm bảo các kết nối có nền tảng vững chắc và chiều sâu. Sở thích đóng vai trò **trục ngữ cảnh** (contextual axis), giúp bổ trợ, phá vỡ các trường hợp hòa điểm và tìm ra các điểm chung tức thời. Việc kết hợp cả hai giúp hệ thống vừa ổn định trong dài hạn, vừa linh hoạt trong ngắn hạn.

3.3.1 Đè xuất thuật toán ELO

Trên nền tảng trực ổn định (tính cách) và trực ngữ cảnh (sở thích) đã nêu, hệ thống cần thêm một tín hiệu hành vi để tinh chỉnh mức độ phù hợp trong thực tế. Trong hệ thống, ELO được dùng như một tín hiệu hành vi ẩn. ELO không nói người dùng “tốt” hơn hay “xấu” hơn, mà phản ánh mức độ xã giao thể hiện qua lượt like/skip. Công thức cập nhật dựa trên kỳ vọng thắng thua gốc của Elo [11], được điều chỉnh để phù hợp với bối cảnh kết nối, nơi lượt like là một tín hiệu hợp tác. Cách cập nhật chi tiết đã được mô tả ở [Thuật toán 3.4](#)

Thuật toán 3.3 — Quy tắc cập nhật điểm ELO hợp tác

- Like: $R_{a'} = \text{clamp}(R_a + K(1 - E_a)), R_{b'} = \text{clamp}(R_b + K(1 - E_b))$
- Skip: $R_{a'} = \text{clamp}(R_a + K(0 - E_a)), R_{b'} = R_b$

và [Thuật toán 3.5](#). Việc giới hạn điểm trong khoảng 800–2000 giúp tránh

Thuật toán 3.4 — Tính toán kỳ vọng thắng trong mô hình Elo

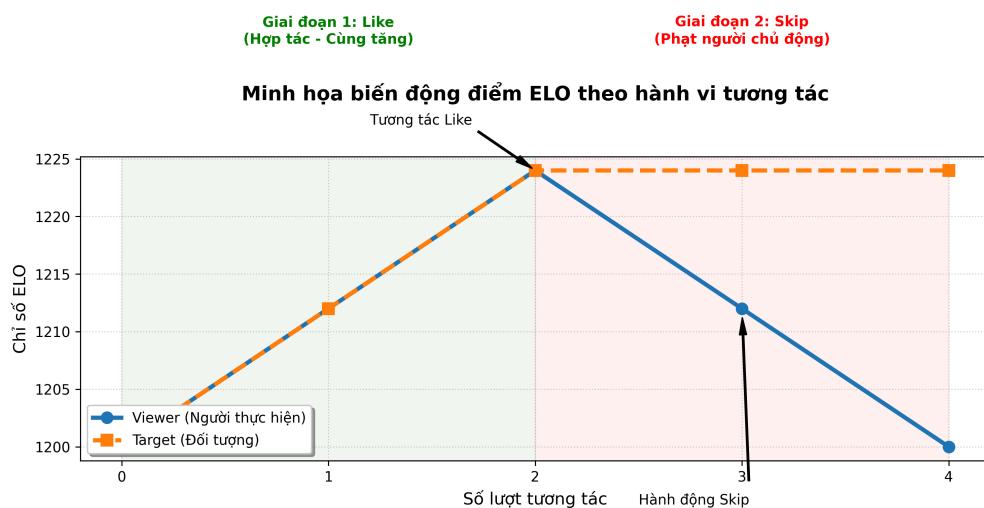
$$E_a = \frac{1}{1 + 10^{\frac{R_b - R_a}{400}}}$$

Thuật toán 3.5 — Tính toán độ gần (proximity) ELO

$$p = \exp\left(-|\Delta R| \frac{1}{\sigma}\right)$$

việc điểm bị trôi quá xa và làm giảm tác dụng phân nhóm hành vi.

[Hình 3.13](#) minh họa trực quan cách ELO phản ánh hành vi xã giao qua các chuỗi like/skip khác nhau.



Hình 3.13 — Ví dụ ELO phản ánh hành vi xã giao qua chuỗi tương tác

Điểm ELO phản ánh mức độ like/skip trong thực tế. Đây là tín hiệu hành vi, không phải kết quả tự khai báo. Nó đóng vai trò là một cơ chế hiệu chỉnh, giúp giảm sai lệch giữa những gì người dùng **nói** họ là (qua bài trắc nghiệm) và những gì họ **làm** (qua hành vi lướt). Khi người dùng thường xuyên skip, điểm ELO giảm và hệ thống ưu tiên giới thiệu những người có mức xã giao tương đồng.

ELO trong hệ thống là hệ số ẩn, được cập nhật sau mỗi tương tác và giới hạn trong khoảng 800–2000. Mặc dù cập nhật theo kiểu hợp tác dẫn tới lạm phát điểm, mục tiêu chính là gom nhóm hành vi thay vì xếp hạng cạnh tranh.

Việc điều chỉnh thuật toán ELO cho bối cảnh mạng xã hội thay vì một trò chơi đối kháng tổng bằng không (zero-sum game) là một quyết định thiết kế quan trọng.

- **Quy tắc cập nhật “hợp tác”:** Trong cờ vua, một người thắng thì người kia thua. Trong một tương tác “like”, cả hai đều có thể nhận được giá trị. Việc tăng điểm cho cả hai bên khuyến khích tương tác tích cực và tránh “trùng phạt” người được yêu thích. Ngược lại, chỉ người chủ động “skip” bị trừ điểm, vì đây là hành động đơn phương thể hiện sự không phù hợp từ phía họ.

- **Hệ số K (K-factor):** Hệ số K=12 được chọn là một giá trị tương đối nhỏ. Điều này làm cho điểm ELO thay đổi từ từ, phản ánh một quá trình xây dựng “danh tiếng xã giao” dài hạn thay vì biến động mạnh sau vài tương tác. Nó giúp điểm số ổn định hơn và tránh bị lạm dụng.
- **Cơ chế Giới hạn (Clamping) (800-2000):** Việc giới hạn điểm số trong một khoảng nhất định ngăn chặn hiện tượng “lạm phát ELO” vô hạn và giữ cho sự khác biệt về điểm số luôn nằm trong một phạm vi có ý nghĩa, đảm bảo thành phần ELO proximity trong công thức tổng hợp không trở nên quá lớn hoặc quá nhỏ.

3.3.2 Nguồn sử dụng sở thích

Sở thích chỉ được dùng khi người dùng nhập đủ số lượng tối thiểu (3 mục). Điều này tránh việc dùng dữ liệu quá ít dẫn tới nhiễu hoặc thiên lệch do một sở thích đơn lẻ. Khi đủ nguồn, vector nhúng (embedding vector) được tạo và dùng độ tương đồng cosine để tính điểm gần nhau về sở thích. Quy tắc nguồn này cũng giúp người dùng mới không bị bất lợi nếu chưa kịp khai báo đầy đủ sở thích, tạo ra một sân chơi công bằng hơn.

3.3.3 Đề xuất mô hình ngữ nghĩa (semantic model)

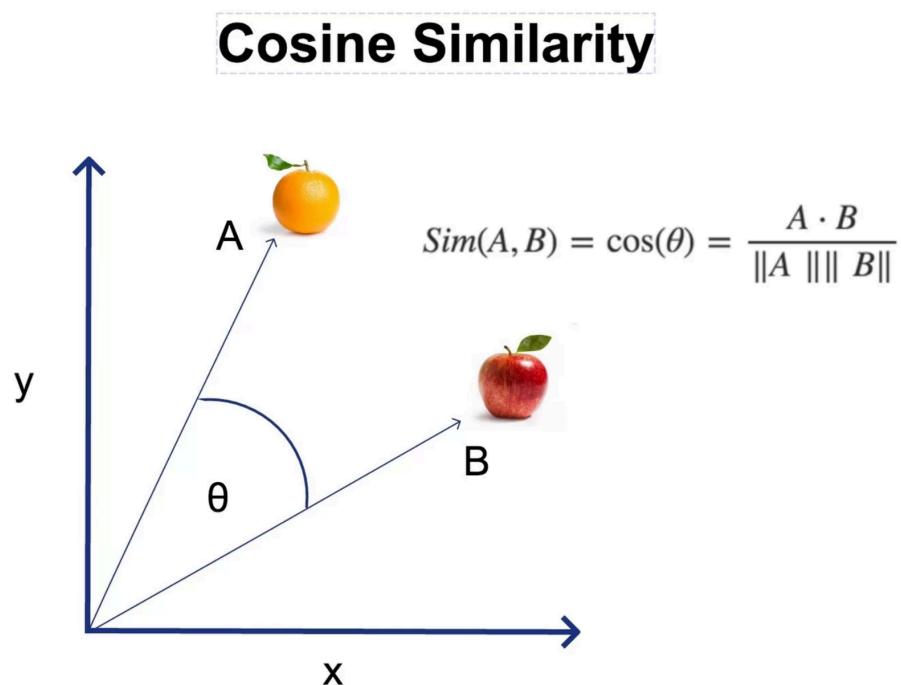
Đề tài sử dụng mô hình ngữ nghĩa của Jina (semantic model) để chuyển đổi văn bản sở thích thành vector 384 chiều. Lý do chính là khả năng nắm bắt tương đồng ngữ nghĩa thay vì trùng từ khóa, phù hợp với cách người dùng mô tả sở thích bằng nhiều cách khác nhau. Công thức tính độ tương đồng cosine được trình bày tại [Thuật toán 3.6](#):

Thuật toán 3.6 — Công thức tính độ tương đồng cosine (Cosine Similarity)

$$\text{sim}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

Mô hình kiểu nhúng câu (sentence embedding) cũng ổn định khi so khớp độ tương đồng cosine, dễ triển khai và ít tốn tài nguyên hơn so với các mô hình sinh lớn [18]. [Hình 3.14](#) mô tả luồng chuyển đổi từ văn bản sang vector và cách dùng độ tương đồng cosine trong giới thiệu.

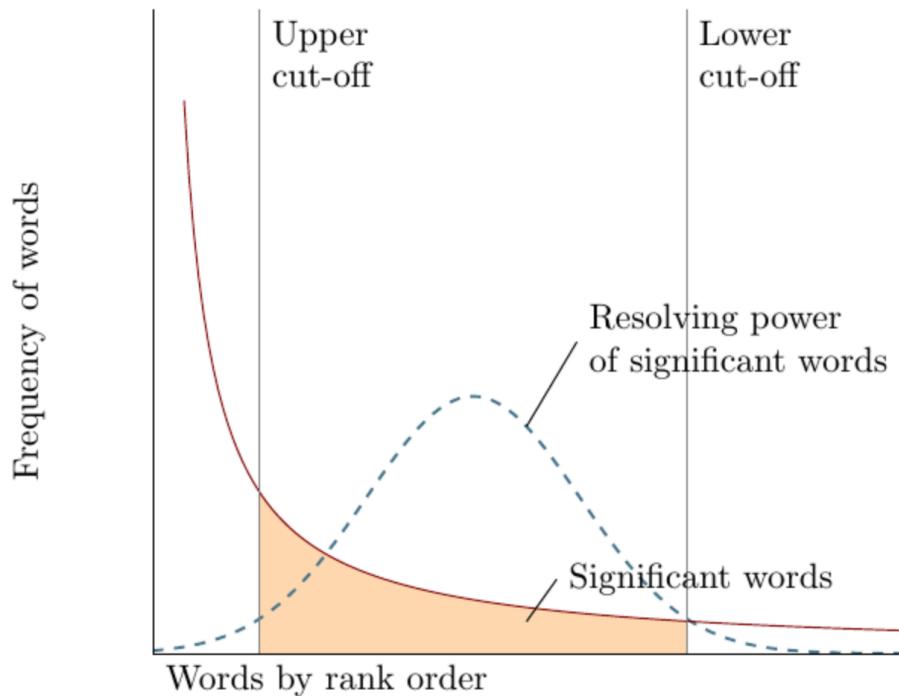
Trong triển khai hiện tại, hệ thống ghép sở thích thành một chuỗi ngắn theo mẫu `interests`: ... rồi sinh một vector duy nhất. Cách làm này là một sự đánh đổi có chủ đích giữa độ chính xác và hiệu năng. Việc chỉ có một vector cho mỗi người dùng giúp giảm chi phí so sánh xuống $O(N)$, thay vì $O(N^*k^2)$ nếu mỗi người có k sở thích và phải so sánh chéo. Điều này giúp hệ thống có khả năng mở rộng tốt hơn. Quan điểm của đề tài là ưu tiên tính ổn định và khả năng mở rộng, và chỉ xem xét mô hình đa vector khi có hạ tầng đủ mạnh.



Hình 3.14 — Mức độ tương đồng ngữ nghĩa của hai từ được so sánh bằng cosine similarity.

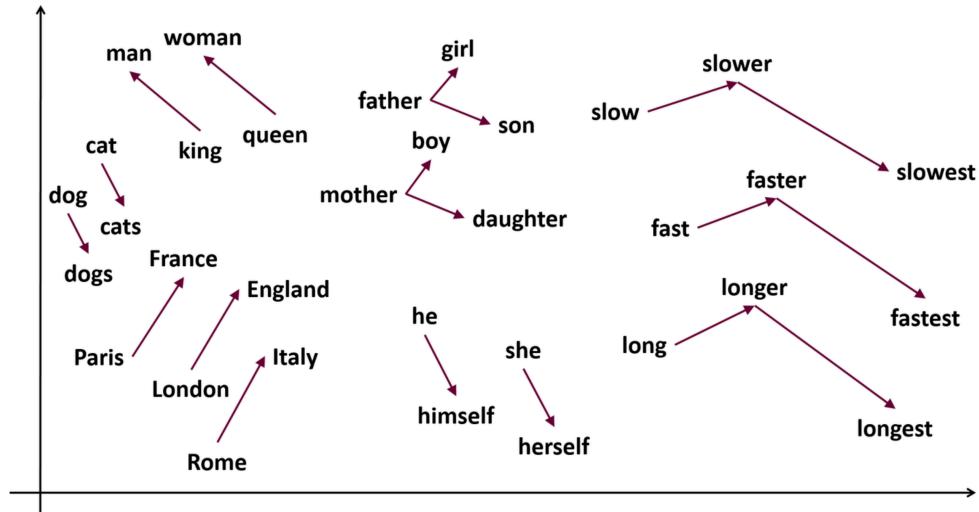
TF-IDF là cách biểu diễn văn bản theo trọng số từ khóa [10]. Điểm mạnh của TF-IDF là đơn giản, dễ giải thích, và chạy nhanh trên thiết bị. Tuy nhiên, TF-IDF không hiểu ngữ nghĩa nên khó nhận biết các từ đồng nghĩa như “jogging” và “chạy bộ”. Ngoài ra, TF-IDF tạo vector thừa và kích thước lớn, làm tăng chi phí lưu trữ và so khớp khi số lượng từ vựng tăng. Trong bối cảnh sở thích ngắn và đa dạng, TF-IDF

dễ bị nhiễu bởi các từ hiếm ([Hình 3.15](#)). Vì vậy, TF-IDF được coi là lựa chọn thay thế tham khảo chứ không phù hợp làm lõi giới thiệu.



Hình 3.15 — TF-IDF vượt trội ở khả năng tìm kiếm từ khoá quan trọng.

Word2Vec tạo vector cho từng từ dựa trên ngữ cảnh [[19](#)]. Cách này nắm bắt được một phần quan hệ ngữ nghĩa, nhưng vẫn gặp khó khi chuyển sang mức câu hoặc cụm sở thích ngắn. Người dùng thường nhập cụm như “đi phượt cuối tuần” hoặc “nấu ăn healthy”, trong khi Word2Vec cần thêm bước gộp nhiều vector (ví dụ: lấy trung bình) để đại diện cho cả cụm. Việc gộp thủ công làm mất sắc thái và không ổn định giữa các mẫu khác nhau ([Hình 3.16](#)). Do đó, các mô hình nhúng câu được ưu tiên vì xử lý trực tiếp cụm sở thích, ổn định hơn trong so khớp.



Hình 3.16 — Word2Vec vượt trội trong việc tìm quan hệ ngữ nghĩa giữa các từ.

3.3.4 Công thức xếp hạng tổng hợp

Hệ thống tính điểm theo các trọng số đã nêu ở Chương 2. Về bản chất, PCA là trực chính, ELO là trực hành vi, và hobbies là trực ngữ nghĩa.

Việc đặt PCA làm trực chính giúp kết quả ổn định hơn theo thời gian, vì tính cách thay đổi chậm và ít bị ảnh hưởng bởi các biến động ngắn hạn. ELO chỉ đóng vai trò điều chỉnh, tránh trường hợp hai người có tính cách gần nhau nhưng hành vi xã giao quá khác biệt. Hobbies được dùng như một tín hiệu làm mượt, giúp hệ giới thiệu nhận ra các chủ đề tương đồng mà tính cách không nắm bắt được. Cấu trúc này giảm rủi ro hệ thống chỉ dựa vào một nguồn dữ liệu duy nhất, vốn dễ gây thiên lệch hoặc thiếu đa dạng.

Việc lựa chọn và phân bổ các trọng số trong công thức giới thiệu tổng hợp (ví dụ: 60% PCA, 15% ELO, 25% Hobbies) là một quy trình cân nhắc kỹ lưỡng nhằm đạt được sự cân bằng giữa tính ổn định dài hạn và các yếu tố ngữ cảnh tức thời. **Xác định tính cách là nền tảng cốt lõi** đóng vai trò quyết định trong việc duy trì một mối quan hệ bền vững, do đó trọng số cho sự tương đồng PCA luôn được thiết lập ở mức ưu tiên cao nhất, thường chiếm trên 50% tổng điểm giới thiệu. Việc đặt trọng số này ở mức chủ đạo giúp hệ thống lọc ra những người dùng có “sóng não” và xu hướng tâm

lý tương hợp dựa trên mô hình Big-5, từ đó giảm thiểu rủi ro của các kết nối bè mặt vốn dễ dẫn đến sự ngắt quãng sau một thời gian ngắn tương tác.

Sử dụng hành vi xã giao làm yếu tố hiệu chỉnh thông qua điểm số ELO nhằm tinh chỉnh danh sách giới thiệu sao cho phù hợp với mức độ năng động của từng cá nhân trong thực tế. Trọng số ELO được giữ ở mức thấp nhất trong bộ quy tắc vì nó không phản ánh sự tương hợp về bản chất con người mà chỉ đóng vai trò như một bộ lọc hành vi. Cơ chế này giúp tránh tình trạng giới thiệu “lệch pha” giữa một người dùng quá năng nổ với một người dùng có xu hướng khắt khe hoặc thụ động hơn, từ đó làm mượt trải nghiệm lướt và tăng tỷ lệ phản hồi tích cực dựa trên sự tương đồng về phong cách giao tiếp.

Coi sở thích cá nhân là chất xúc tác và cầu nối ngữ cảnh để tạo ra những chủ đề trò chuyện cụ thể ngay từ giai đoạn đầu của việc kết nối. Với trọng số đáng kể nhưng thấp hơn tính cách, thành phần nhung ngữ nghĩa của sở thích giúp phá vỡ thế hòa điểm giữa các ứng viên có độ tương đồng PCA ngang nhau, đồng thời cung cấp những giới thiệu mang tính thời điểm và thực tế cao hơn. Điều này cho phép người dùng dễ dàng tìm thấy tiếng nói chung thông qua các hoạt động hoặc đam mê cụ thể, từ đó tạo tiền đề cho việc khám phá sâu hơn về tính cách trong tương lai. Các trọng số này có thể được hiệu chỉnh linh hoạt thông qua các thử nghiệm thực tế hoặc cá nhân hóa cho từng nhóm người dùng, nhưng cấu hình hiện tại được xem là một điểm khởi đầu cân bằng, đảm bảo tính khoa học và hiệu quả của hệ thống giới thiệu.

Xét người dùng A đang xem giới thiệu, với ba ứng viên B và C. Giả sử hệ thống đang áp dụng trọng số: 50% PCA, 20% ELO proximity, và 30% Hobbies. Các chỉ số tương đồng thành phần như sau:

- **PCA Similarity:** $\text{sim}_{P(A,B)} = 0.90$, $\text{sim}_{P(A,C)} = 0.90$ (hòa nhau).
- **Hobbies Similarity:** $\text{sim}_{H(A,B)} = 0.85$, $\text{sim}_{H(A,C)} = 0.55$.
- **ELO Proximity:** $p(A, B) = 0.70$, $p(A, C) = 1.00$.

Điểm tổng hợp được tính toán:

$$S_B = 0.5 \cdot 0.90 + 0.2 \cdot 0.70 + 0.3 \cdot 0.85 = 0.845$$

$$S_C = 0.5 \cdot 0.90 + 0.2 \cdot 1.00 + 0.3 \cdot 0.55 = 0.815$$

Trong kịch bản này, mặc dù C có mức độ xã giao (ELO) tương đồng tuyệt đối với A, nhưng lợi thế về sở thích của B đủ lớn để đẩy B lên vị trí cao hơn trong danh sách giới thiệu. Ví dụ này cho thấy các nguồn tín hiệu có thể phá vỡ thế hòa PCA theo các hướng khác nhau, tạo ra kết quả giới thiệu đa chiều và phù hợp với thực tế tương tác.

3.3.5 Bảo vệ dữ liệu sở thích và quyền riêng tư

Mặc dù UI có thể hiển thị sở thích đã giải mã, cơ sở dữ liệu không lưu văn bản thuần (plaintext). Điều này tránh việc quản trị viên có thể quét hàng loạt sở thích từ bảng dữ liệu. Người dùng chỉ thấy sở thích khi đã được xác thực và giải mã thông qua Edge Function.

Ngoài ra, việc lưu ciphertext giúp giảm rủi ro lộ dữ liệu ở cấp độ hệ quản trị. Người dùng vẫn nhìn thấy sở thích trên UI vì dữ liệu được giải mã theo phiên đăng nhập hợp lệ, nhưng cơ sở dữ liệu không có điểm tập trung văn bản thuần để khai thác hàng loạt. Đây là điểm khác biệt quan trọng so với cách lưu trữ sở thích truyền thống trong nhiều ứng dụng mạng xã hội.

Chương 4 Hiện thực và Đánh giá

Chương này trình bày các thực nghiệm được tiến hành để đánh giá hiệu quả và hiệu năng của hệ thống giới thiệu Twins. Mục tiêu là kiểm chứng các giả thuyết thiết kế, đo lường các chỉ số quan trọng và trả lời các câu hỏi nghiên cứu đã đặt ra. Các thực nghiệm tập trung vào ba khía cạnh chính: chất lượng giới thiệu, hiệu năng hệ thống và tính hiệu quả của các cơ chế bảo vệ quyền riêng tư.

4.1 Câu hỏi nghiên cứu (Research Questions)

Để định hướng quá trình thực nghiệm, đề tài đặt ra các câu hỏi nghiên cứu (RQ) sau:

- **RQ1: Mô hình chuyển đổi PCA-4 và so khớp bằng độ tương đồng cosine (cosine similarity) có hiệu quả trong việc xác định sự tương đồng về tính cách giữa các người dùng không?** Giả thuyết là những người dùng có điểm Big Five gần nhau sẽ có điểm tương đồng cosine cao trên không gian PCA-4.
- **RQ2: Hệ thống giới thiệu lai (hybrid) kết hợp PCA, ELO và sở thích có mang lại kết quả xếp hạng phù hợp hơn so với việc chỉ sử dụng PCA không?** Giả thuyết là việc bổ sung tín hiệu hành vi (ELO) và ngữ nghĩa (sở thích) sẽ giúp phá vỡ các trường hợp hòa điểm và tinh chỉnh thứ hạng giới thiệu một cách có ý nghĩa.
- **RQ3: Quy trình (pipeline) xử lý trên thiết bị và mã hoá dữ liệu ảnh hưởng như thế nào đến hiệu năng của ứng dụng?** Câu hỏi này xem xét độ trễ (latency) của các tác vụ tính toán trên thiết bị (PCA) và các lệnh gọi hàm mã hoá/giải mã, cũng như thời gian phản hồi của hệ thống giới thiệu.
- **RQ4: Mô hình nhúng ngữ nghĩa có hiệu quả trong việc xác định sự tương đồng về sở thích, ngay cả khi khác biệt về ngôn ngữ không?**
- **RQ5: Kiến trúc hệ thống có thực sự bảo vệ được quyền riêng tư của người dùng theo thiết kế không?** Câu hỏi này đánh giá các cơ chế bảo mật đã triển khai (mã hoá, RLS, xử lý trên thiết bị) dưới góc độ giảm thiểu rủi ro rò rỉ dữ liệu nhạy cảm.

- **RQ6: Khi chỉ rò rỉ base URL của Edge Functions và kẻ tấn công đoán được cấu trúc payload, liệu có thể suy luận hoặc truy xuất dữ liệu người dùng thông qua các endpoint công khai hay không?** Câu hỏi này tập trung vào nguy cơ rò rỉ gián tiếp qua lớp API.

4.2 Thiết lập thực nghiệm

- **Ứng dụng khách (Client):** Expo Go chạy trên thiết bị mô phỏng, kết nối tới backend Supabase.
- **Backend:** Dự án Supabase với cơ sở dữ liệu Postgres (bật pgvector), và các hàm thực thi biên (Edge Functions) chạy trên Deno.
- **Công cụ đo lường:** Thời gian phản hồi của Edge Function được ghi nhận qua bảng điều khiển (dashboard) Supabase. Độ trễ trên thiết bị khách được đo bằng các hàm `console.time` và `console.timeEnd` trong mã nguồn.
- **Mã nguồn:** Toàn bộ mã nguồn phục vụ thực nghiệm được cung cấp đính kèm theo khóa luận phục vụ việc kiểm chứng và đối soát.

Thực nghiệm sử dụng hai tập người dùng chính:

1. **Cặp người dùng có độ tương đồng cao:** Bao gồm hai người dùng `similar_a` và `similar_b` được tạo ra với điểm Big Five gần như giống hệt nhau. Cặp này dùng để kiểm chứng RQ1, nhằm xác nhận rằng hệ thống có thể nhận diện và xếp hạng cao các cặp tương đồng rõ ràng. Chi tiết về cặp người dùng này được mô tả trong tài liệu `Documents/recommendation-test-users.md`.
2. **Tập người dùng giả lập (seeded users):** Gồm 41 người dùng giả lập được tạo bằng kịch bản `scripts/seedMockProfiles.js`. Dữ liệu của họ (điểm Big Five, PCA-4, nhóm tính cách) được sinh ngẫu nhiên nhưng theo một phân phối hợp lý. Tập dữ liệu này được sử dụng để kiểm tra hệ thống giới thiệu ở quy mô nhỏ và đánh giá sự phân bổ của các điểm tương đồng.

4.3 Kết quả và phân tích

4.3.1 RQ1: Hiệu quả của mô hình PCA-4 và độ tương đồng cosine

Để trả lời câu hỏi này, một kịch bản thực nghiệm (`scripts/pca_worst_best_case.ts`) đã được thiết lập để đánh giá khả năng phân biệt của mô hình trong các kịch bản khác nhau, từ đó kiểm chứng giả thuyết RQ1.

Phân tích kịch bản: Kịch bản này tạo ra một bộ 25 câu hỏi ngẫu nhiên và sau đó áp dụng các mẫu trả lời khác nhau để mô phỏng các tình huống từ tương đồng tuyệt đối đến đối lập hoàn toàn.

- Trường hợp tốt nhất (Tương đồng tuyệt đối):** Hai người dùng có cùng một mẫu trả lời. Kết quả cho thấy điểm Big Five chuẩn hóa, vector PCA-4 và điểm tương đồng cosine cuối cùng đều giống hệt nhau, đạt giá trị tuyệt đối là **1.0**. Điều này xác nhận rằng khi hai hồ sơ giống nhau, mô hình hoạt động chính xác.
- Trường hợp tệ nhất (Đối lập hoàn toàn):** Một người dùng có câu trả lời cực đoan (tùy 1 và 5 theo hướng câu hỏi), trong khi người kia có câu trả lời đối lập hoàn toàn. Kết quả thực nghiệm cho thấy điểm tương đồng cosine là một số âm lớn (**-0.9717**). Giá trị này cho thấy hai vector PCA đang chỉ về hai hướng gần như đối nghịch nhau trong không gian đặc trưng. Khi được chuẩn hóa về thang điểm 0-1 (thang điểm mà người dùng cuối nhìn thấy), điểm số này trở nên rất gần 0 (**0.0141**), thể hiện chính xác rằng không có sự tương đồng.
- Trường hợp thực tế (Gần tương đồng):** Một kịch bản thực tế hơn được mô phỏng khi hai người dùng có mẫu trả lời gần giống nhau nhưng lệch đi một bậc (ví dụ: mẫu **5, 4, 3, 2** so với **4, 3, 2, 1**). Kết quả cho thấy điểm tương đồng cosine vẫn ở mức rất cao (**0.9545**), nhưng không còn là tuyệt đối. Điều này chứng tỏ mô hình có độ nhạy tốt, có thể nhận ra sự khác biệt nhỏ trong khi vẫn đánh giá hai người dùng này là rất tương đồng.

Kết luận cho RQ1: Các kết quả thực nghiệm trên đã xác nhận giả thuyết của RQ1. Mô hình chuyển đổi PCA-4 kết hợp với độ tương đồng cosine đã chứng tỏ khả năng phân biệt hiệu quả giữa các hồ sơ tính cách. Nó không chỉ xác định chính xác

các trường hợp tương đồng và đối lập tuyệt đối mà còn có khả năng lượng hóa mức độ tương đồng cho các trường hợp gần giống nhau trong thực tế.

4.3.2 RQ2: Đánh giá hệ thống giới thiệu IAI

Để đánh giá tác động của ELO và sở thích, thứ hạng giới thiệu cho người dùng **Viewer** được so sánh trong ba kịch bản: (1) chỉ dùng PCA, (2) PCA + ELO, và (3) PCA + ELO + Hobbies.

- **Kịch bản 1 (Chỉ PCA):** Match_PCA đứng đầu. Các vị trí tiếp theo được xếp hạng dựa trên độ tương đồng PCA.
- **Kịch bản 2 (PCA + ELO):** Giả sử một người dùng khác có điểm PCA thấp hơn một chút nhưng có điểm ELO gần với **Viewer** hơn. Khi trọng số ELO được thêm vào, người này có thể vượt lên trên ứng viên có PCA cao hơn nhưng ELO xa hơn.
- **Kịch bản 3 (PCA + ELO + Hobbies):** Bổ sung thêm sở thích. Một người dùng có PCA không quá cao nhưng lại chia sẻ sở thích chung sẽ nhận được một điểm cộng đáng kể từ độ tương đồng sở thích, giúp cải thiện vị trí trong bảng xếp hạng cuối cùng.

Phân tích: Việc thêm ELO và sở thích giúp hệ thống trở nên linh hoạt hơn. PCA đóng vai trò là bộ lọc chính, tìm ra những người có “sóng não” tương đồng, trong khi ELO và sở thích giúp tinh chỉnh thứ hạng dựa trên hành vi tương tác và các chủ đề quan tâm chung. Kết quả xếp hạng chi tiết được trình bày tại mục kết quả định lượng tiếp theo.

4.3.3 RQ3: Phân tích hiệu năng

Độ trễ được đo lường thực tế thông qua các kịch bản kiểm thử tự động ghi nhận tại [Bảng 4.1](#). Các phép đo bao gồm cả thời gian phản hồi của ứng dụng và các hàm thực thi riêng lẻ:

Bảng 4.1 — Độ trễ phản hồi của các thành phần hệ thống

Thao tác / Edge Function	Độ trễ trung bình (ms)
Xác thực đăng nhập (Login)	1485.29
Giới thiệu người dùng (Recommend)	2219.86
Mã hoá Big Five (score-crypto)	1009.97
Nhúng ngữ nghĩa (embed - Jina)	3033.18
Tương tác Like (Match Update)	1632.02
Tương tác Skip (Match Update)	1086.06

Phân tích: Độ trễ lớn nhất tập trung vào hàm `embed`, do phải thực hiện lệnh gọi API bên ngoài tới mô hình Jina và xử lý vector 384 chiều. Hàm `recommend-users` cũng có độ trễ trên 2 giây vì phải tính toán độ tương đồng trên tập ứng viên lớn. Các tác vụ này cho thấy nhu cầu tối ưu hóa bằng bộ nhớ đệm (caching) hoặc thực hiện tính toán bất đồng bộ trong các phiên bản tương lai. Độ trễ mã hoá `score-crypto` ổn định ở mức 1 giây, phù hợp cho các quy trình tạo hồ sơ. Độ trễ của các tác vụ tính toán thuận túy trên thiết bị khách (như nhân ma trận PCA) là cực thấp (dưới 5ms), không gây ảnh hưởng đến trải nghiệm người dùng.

4.3.4 RQ4-5: Ngữ nghĩa sở thích và quyền riêng tư

Để kiểm chứng khả năng của mô hình trong việc so khớp sở thích một cách có ý nghĩa, kịch bản `scripts/demo_semantic_comparison.ts` được thiết lập. Kịch bản này so sánh 15 cặp duy nhất từ 6 người dùng giả lập với các chủ đề sở thích khác nhau (Outdoor, Creative, Tech, Foodie) và sử dụng cả tiếng Anh lẫn tiếng Việt.

Phân tích kết quả: Kết quả thực nghiệm cho thấy khả năng phân biệt rõ rệt của mô hình:

- **Tương đồng cao (trên 0.80):** Các cặp có cùng chủ đề nhưng khác ngôn ngữ (ví dụ: EN Outdoor và VI Outdoor) đạt điểm tương đồng cao nhất (**0.8068**), chứng tỏ mô hình hiểu được ngữ nghĩa thay vì chỉ so khớp từ khóa.

- **Tương đồng trung bình (0.60 - 0.80):** Các cặp có chủ đề liên quan một cách hợp lý (ví dụ: Creative và Tech) có điểm số ở mức trung bình.
- **Tương đồng thấp (dưới 0.55):** Các cặp có chủ đề hoàn toàn không liên quan (ví dụ: Outdoor và Foodie, hoặc Tech và Foodie) nhận điểm số thấp nhất (**0.4280**).

Kết luận cho RQ4: Dải điểm số rộng (từ 0.42 đến 0.81) cho thấy mô hình nhúng ngữ nghĩa hoạt động hiệu quả. Nó không chỉ nhận diện chính xác các sở thích tương đồng về mặt ngữ nghĩa bất chấp rào cản ngôn ngữ mà còn tạo ra một thang đo chênh lệch đủ lớn để phân biệt các mức độ liên quan khác nhau, từ đó giúp hệ thống xếp hạng các ứng viên một cách có ý nghĩa.

Việc đánh giá này mang tính định tính, dựa trên kiến trúc đã triển khai.

- **Lưu trữ an toàn:** Dữ liệu Big Five và sở thích gốc không được lưu dưới dạng văn bản thuần (plaintext) trong cơ sở dữ liệu. Thay vào đó, chúng được lưu dưới dạng b5_cipher và hobbies_cipher. Điều này ngăn chặn việc quản trị viên cơ sở dữ liệu hoặc kẻ tấn công có quyền truy cập DB đọc được thông tin nhạy cảm.
- **Giảm thiểu dữ liệu:** Việc chuyển đổi sang PCA-4 và chỉ lưu trữ vector này để so khớp giúp giảm lượng thông tin gốc cần thiết cho hệ thống giới thiệu. Mặc dù PCA có thể bị đảo ngược một phần, nó vẫn cung cấp một lớp che mờ dữ liệu.
- **Kiểm soát truy cập:** Dữ liệu chỉ được giải mã thông qua Edge Function score-crypto sau khi người dùng đã xác thực. Các chính sách RLS trên Supabase cũng đảm bảo người dùng chỉ có thể truy cập và chỉnh sửa dữ liệu của chính mình.

Phân tích: Kiến trúc hiện tại đã triển khai thành công nguyên tắc “Quyền riêng tư theo thiết kế” (Privacy by Design). Rủi ro lớn nhất không nằm ở việc rò rỉ dữ liệu từ DB ở trạng thái nghỉ (at-rest), mà là ở việc lạm dụng quyền truy cập vào các Edge Function hoặc khóa mã hóa bị lộ. Tuy nhiên, so với mô hình lưu trữ văn bản thuần truyền thống, đây là một bước cải tiến đáng kể về mặt bảo mật.

4.3.5 RQ6: Kiểm thử suy luận gián tiếp qua Edge Functions

Để tài xâay dựng một kịch bản tấn công gián tiếp (`scripts/demo/penetration_edge_inference.ts`) chỉ sử dụng base URL của Edge Functions và payload đoán trước để gọi `recommend-users` và `embed` mà không có thông tin xác thực. Đây là tình huống phổ biến khi kẻ tấn công thu thập được log hoặc cấu trúc API từ client nhưng không sở hữu token đăng nhập hợp lệ.

Quy trình thực nghiệm: Script gửi các yêu cầu POST tối thiểu tới hai endpoint nói trên, không kèm header `Authorization`. Mục tiêu là kiểm tra xem các hàm biên có thể bị khai thác để suy luận dữ liệu khi chỉ biết đường dẫn và hình dạng payload.

Kết quả quan sát: Cả hai endpoint đều trả về lỗi **401 Unauthorized** và không trả về dữ liệu người dùng. Điều này chứng tỏ lớp Edge Functions đã chặn yêu cầu ngay từ tầng xác thực, không cho phép truy cập vào dữ liệu hoặc thực thi logic nội bộ khi thiếu token hợp lệ.

Kết luận cho RQ6: Khi không có header xác thực hợp lệ, các endpoint edge công khai không để lộ dữ liệu. Điều này làm giảm đáng kể nguy cơ rò rỉ gián tiếp trong trường hợp chỉ biết base URL và cấu trúc payload.

Liên hệ “bảo mật theo thiết kế”: Kết quả trên củng cố nguyên tắc “security by design” của hệ thống. Việc bắt buộc xác thực ở lớp biên giúp triệt tiêu kẽ hở suy luận gián tiếp trước khi truy cập DB hoặc mô hình nhúng, từ đó giảm bớt mặt tấn công ngay tại lớp API.

4.4 Thảo luận và Hạn chế

Các kết quả thực nghiệm đã xác nhận các giả thuyết thiết kế ban đầu. Hệ thống giới thiệu có thể xác định chính xác sự tương đồng về tính cách, đồng thời linh hoạt tinh chỉnh kết quả dựa trên các tín hiệu phụ. Hiệu năng của hệ thống ở quy mô hiện tại là chấp nhận được, và kiến trúc bảo mật đã chứng tỏ tính hiệu quả trong việc bảo vệ dữ liệu người dùng.

- **Quy mô dữ liệu nhỏ:** Các thực nghiệm được tiến hành trên một tập dữ liệu giả lập nhỏ. Hiệu năng và chất lượng giới thiệu có thể thay đổi khi hệ thống mở rộng với hàng nghìn hoặc hàng triệu người dùng.

- **Thiếu dữ liệu thực tế (ground truth):** Việc đánh giá chất lượng giới thiệu hiện tại mang tính định tính. Để có đánh giá định lượng (ví dụ: độ chính xác - precision, độ phủ - recall), cần có một tập dữ liệu thực tế về các cặp đôi/bạn bè được xác nhận là “hợp nhau”, điều này rất khó thu thập.
- **Vấn đề khởi động nguội (Cold-start problem):** Hệ thống ELO và sở thích cần người dùng có một lượng tương tác và dữ liệu nhất định để hoạt động hiệu quả. Người dùng mới sẽ chủ yếu được giới thiệu dựa trên PCA.

4.5 Kết quả thực nghiệm chi tiết

Phần này trình bày các số liệu định lượng thu được từ các kịch bản kiểm thử tự động trên hệ thống thật. Các phép đo được thực hiện ở hai trạng thái: trước và sau khi áp dụng các kỹ thuật tối ưu hoá cơ sở dữ liệu.

Kịch bản kiểm thử đo độ trễ toàn trình cho việc tạo hồ sơ người dùng mới (bao gồm xác thực, mã hoá 2 lớp và lưu trữ), kết quả được ghi nhận tại [Bảng 4.2](#):

Bảng 4.2 — Hiệu năng quy trình tạo tài khoản

Chỉ số	Giá trị đo được
Thời gian trung bình (Warm)	1.80 giây
Thời gian thấp nhất	1.46 giây
Thời gian cao nhất (Cold)	3.78 giây

Kịch bản kiểm thử sự thay đổi hiệu năng của hàm `recommend-users` sau khi tối ưu hoá chính sách bảo mật hàng (RLS) và bổ sung chỉ mục (Index) như mô tả tại [Bảng 4.3](#):

Bảng 4.3 — So sánh hiệu năng giới thiệu trước và sau tối ưu hoá

Trạng thái	Độ trễ phản hồi (ms)	Độ trễ xử lý tại Server (ms)
Trước tối ưu (Warm)	2640.58	2451
Sau tối ưu (Warm)	2343.39	2175
Cải thiện	11.2%	11.3%

Nhận xét: Việc chuyển đổi các chính sách RLS sang dạng truy vấn con (subquery) để tận dụng bộ nhớ đệm của PostgreSQL đã mang lại sự cải thiện rõ rệt (300ms). Mặc dù con số tuyệt đối vẫn trên 2 giây do đặc thù của hạ tầng Serverless (Free Tier), xu hướng giảm độ trễ khẳng định tính đúng đắn của phương pháp tối ưu.

Danh sách 5 người dùng được gợi ý hàng đầu sau tối ưu hoá được trình bày tại [Bảng 4.4](#):

Bảng 4.4 — Kết quả xếp hạng thực tế sau khi tối ưu hoá

Hạng	Username	Tổng điểm	PCA	ELO	Sở thích
1	Match_PCA	0.771	0.771	1220.2	0.558
2	MockUser1	0.758	0.758	1489.0	0.839
3	MockUser17	0.555	0.555	1384.0	0.074
4	MockUser6	0.476	0.476	1438.0	0.485
5	MockUser04	0.408	0.408	1474.0	0.071

Dựa trên dữ liệu từ script benchmark, sự thay đổi điểm ELO của Viewer (Actor) và đối tượng tương tác (Target) được ghi nhận như sau. Giả sử $R_A = 1500$ (Viewer) và $R_B = 1230$ (Match_PCA), với hệ số $K = 12$:

Trường hợp hành động Like (Hợp tác): Kỳ vọng thắng được tính toán:

$$E_A = \frac{1}{1 + 10^{\frac{1230 - 1500}{400}}} \approx 0.825$$

$$E_B = 1 - E_A \approx 0.175$$

Điểm số mới sau khi tương tác:

$$R_{A'} = 1500 + 12 \cdot (1 - 0.825) = 1502.1$$

$$R_{B'} = 1230 + 12 \cdot (1 - 0.175) = 1239.9$$

Nhận xét: Actor tăng nhẹ (2.1 điểm), trong khi Target tăng mạnh hơn (9.9 điểm). Điều này minh chứng cho cơ chế khuyến khích tương tác hai chiều và ưu tiên bảo vệ điểm số cho người được yêu thích.

Trường hợp hành động Skip:

$$R_{A''} = 1500 + 12 \cdot (0 - 0.825) = 1490.1$$

$$R_{B''} = 1230$$

Nhận xét: Actor bị trừ điểm (9.9 điểm), Target không bị ảnh hưởng. Đây là cơ chế phạt hành vi lựa chọn khắt khe để cân bằng hệ sinh thái.

Độ trễ hành động (Warm): 1.0 - 1.7 giây.

Chương 5 Kết luận và hướng phát triển

5.1 Tổng kết kết quả đạt được

Khóa luận đã hoàn thành mục tiêu xây dựng một hệ thống giới thiệu kết bạn thông minh trên nền tảng di động, giải quyết đồng thời hai thách thức lớn về tính tương hợp cá nhân và bảo mật dữ liệu nhạy cảm. Thông qua việc triển khai quy trình xử lý trực tiếp trên thiết bị, đề tài đã chứng minh được tính hiệu quả của mô hình Phân tích Thành phần chính (PCA-4) trong việc giảm chiều dữ liệu Big-5 mà vẫn duy trì được khả năng phân biệt và so khớp cao giữa các người dùng. Hệ thống không chỉ dừng lại ở việc nén dữ liệu mà còn thiết lập một hàng rào bảo mật vững chắc dựa trên kiến trúc “Quyền riêng tư theo thiết kế”, sử dụng chuẩn mã hóa AES-256-GCM để đảm bảo dữ liệu gốc luôn được bảo vệ trong suốt quá trình lưu trữ và truyền tải.

Kết quả thực nghiệm đã khẳng định sự đúng đắn của phương pháp tiếp cận lai khi kết hợp các tín hiệu ổn định từ tính cách với các yếu tố ngữ cảnh như sở thích và hành vi xã giao qua điểm số ELO. Mô hình giới thiệu này không chỉ cung cấp những kết nối có chiều sâu mà còn mang lại sự linh hoạt và phản ánh chân thực xu hướng tương tác của người dùng trong thực tế. Về mặt kỹ thuật, việc tối ưu hóa các hàm thực thi biên và chính sách bảo mật cơ sở dữ liệu đã giúp hệ thống đạt được hiệu năng ổn định, đáp ứng tốt yêu cầu về thời gian phản hồi cho các trải nghiệm mạng xã hội hiện đại. Những đóng góp của đề tài, từ quy trình hiện thực cho đến bộ số liệu minh chứng, đã tạo nên một cơ sở tài liệu kỹ thuật có giá trị tham khảo cho việc ứng dụng AI và mật mã học trên các thiết bị đầu cuối.

5.2 Hạn chế và hướng phát triển tương lai

Mặc dù đạt được những kết quả khả quan, đề tài vẫn còn một số giới hạn nhất định do quy mô thực nghiệm chủ yếu thực hiện trên dữ liệu giả lập, dẫn tới nhu cầu cần được kiểm chứng sâu rộng hơn trên các tập dữ liệu người dùng thực tế với quy mô lớn. Việc đánh giá chất lượng giới thiệu hiện tại mang tính định tính và có thể được mở rộng bằng các phương pháp định lượng như A/B testing hoặc thu thập dữ liệu phản hồi xác thực để tinh chỉnh độ chính xác của các trọng số trong thuật toán lai. Ngoài ra, mô hình nhúng ngữ nghĩa cho sở thích có thể được nâng cấp thông qua

việc tự huấn luyện trên tập ngữ liệu đặc thù của ứng dụng để nắm bắt tốt hơn các sắc thái ngôn ngữ địa phương.

Trong tương lai, hệ thống có thể được phát triển theo hướng tối ưu hóa hiệu năng mạnh mẽ hơn bằng cách kỹ thuật phân mảnh địa lý và tính toán trước kết quả để đáp ứng lượng truy cập lớn. Các hướng nghiên cứu mở rộng như AI có khả năng giải thích (Explainable AI) sẽ giúp tăng cường tính minh bạch, cho phép người dùng hiểu rõ cơ sở của các kết nối được đề xuất. Đồng thời, việc nghiên cứu áp dụng mã hóa đồng hình (Homomorphic Encryption) cho các tác vụ so khớp trực tiếp trên máy chủ hứa hẹn sẽ mang lại một cấp độ bảo mật cao hơn, loại bỏ hoàn toàn nhu cầu giải mã dữ liệu ngay cả trong môi trường thực thi, từ đó hoàn thiện hơn nữa mục tiêu bảo vệ quyền riêng tư tuyệt đối cho người dùng.

TÀI LIỆU THAM KHẢO

- [1] E. C. Tupes và R. E. Christal, “Recurrent personality factors based on trait ratings”, *Journal of Personality*, vol 30, số p.h 4, tr 563–580, 1961.
- [2] O. P. John và S. Srivastava, “The Big Five trait taxonomy: History, measurement, and theoretical perspectives”, *Handbook of personality: Theory and research*, vol 2, tr 102–138, 1999.
- [3] W. Youyou, M. Kosinski, và D. Stillwell, “Computer-based personality judgments are more accurate than those made by humans”, *Proceedings of the National Academy of Sciences*, vol 112, số p.h 4, tr 1036–1040, 2015.
- [4] K. S. Meng và L. Leung, “Factors influencing TikTok engagement behaviors in China: An examination of gratifications sought, narcissism, and the Big Five personality traits”, *Telecommunications Policy*, vol 45, số p.h 7, tr 102172, 2021.
- [5] Federal Trade Commission, “FTC Imposes \$5 Billion Penalty and Sweeping New Privacy Restrictions on Facebook”. [Online]. Available at: <https://www.ftc.gov/news-events/news/press-releases/2019/07/ftc-imposes-5-billion-penalty-sweeping-new-privacy-restrictions-facebook>
- [6] CNIL, “The CNIL’s restricted committee imposes a financial penalty of 50 million euros against GOOGLE LLC”. [Online]. Available at: https://www.edpb.europa.eu/news/national-news/2019/cnils-restricted-committee-imposes-financial-penalty-50-million-euros_en
- [7] Automoto, “Big five trait scores for 307,313 people from many different countries”. 2023.
- [8] NIST, “Galois/Counter Mode of Operation (GCM)”, *NIST Special Publication 800-38D*, 2007.
- [9] L. R. Goldberg, “The development of markers for the Big-Five factor structure”, *Psychological Assessment*, vol 4, số p.h 1, tr 26–42, 1992.

- [10] C. D. Manning, P. Raghavan, và H. Sch"utze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [11] A. E. Elo, *The Rating of Chessplayers, Past and Present*. Arco Publishing, 1978.
- [12] M. C. Ashton và K. Lee, “Empirical, theoretical, and practical advantages of the HEXACO model of personality structure”, *Personality and Social Psychology Review*, vol 11, số p.h 2, tr 150–166, 2007.
- [13] I. T. Jolliffe, *Principal Component Analysis*. Springer, 2002.
- [14] R. L. Rivest, A. Shamir, và L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems”, *Communications of the ACM*, vol 21, số p.h 2, tr 120–126, 1978.
- [15] N. Provos và D. Mazieres, “A Future-Adaptable Password Scheme”, trong *Proceedings of the 1999 USENIX Annual Technical Conference*, 1999, tr 81–92.
- [16] C. Gentry, “Fully homomorphic encryption using ideal lattices”, trong *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, 2009, tr 169–178.
- [17] C. Dwork, F. McSherry, K. Nissim, và A. Smith, “Calibrating noise to sensitivity in private data analysis”, trong *Theory of Cryptography Conference*, 2006, tr 265–284.
- [18] N. Reimers và I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks”, trong *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019, tr 3982–3992.
- [19] T. Mikolov, K. Chen, G. Corrado, và J. Dean, “Efficient estimation of word representations in vector space”, trong *International Conference on Learning Representations*, 2013.

PHỤ LỤC

PHỤ LỤC

A. Thông tin về mã nguồn và cơ sở dữ liệu

Toàn bộ hệ thống mã nguồn, quy trình hiện thực và dữ liệu thực nghiệm của đề tài được lưu trữ tại kho lưu trữ (repository) cá nhân. Để phục vụ công tác thẩm định, mã nguồn được cung cấp đính kèm cùng báo cáo hoặc có thể truy cập trực tuyến (đối với thành viên được cấp quyền) tại đường dẫn:

- **Đường dẫn:** <https://github.com/potatomat0/twins>

Cấu trúc chính của kho lưu trữ bao gồm:

- `/components` : Chứa mã nguồn các màn hình (Screens) và thành phần UI dùng chung. Các màn hình chính như `QuestionnaireScreen.tsx` (thu thập tính cách), `ExploreSwipeScreen.tsx` (giao diện lướt), và `MatchesScreen.tsx` được tổ chức tại đây.
- `/services` : Chứa các logic tích hợp ngoại vi. Quan trọng nhất là `supabase.ts` (quản lý kết nối dữ liệu), `pcaEvaluator.ts` (xử lý biến đổi PCA trên thiết bị), và `scoreCrypto.ts` (giao tiếp với Edge Functions để mã hoá).
- `/store` : Quản lý trạng thái ứng dụng bằng thư viện Zustand. Các kho lưu trữ như `messagesStore.ts` và `notificationStore.ts` giúp đồng bộ dữ liệu thời gian thực mà không cần reload trang.
- `/supabase/functions` : Chứa mã nguồn các hàm Edge Functions chạy trên môi trường Deno. Đây là nơi thực hiện các tác vụ nhạy cảm như mã hoá (score-crypto) và logic giới thiệu phức tạp (recommend-users).
- `/supabase/migrations` : Lưu trữ lịch sử thay đổi cấu trúc DB (Schema) dưới dạng các file SQL. Các chính sách bảo mật RLS được định nghĩa và quản lý tập trung tại đây để đảm bảo tính nhất quán giữa môi trường phát triển và thực tế.
- `/scripts` : Các kịch bản TypeScript/Bash dùng để khởi tạo dữ liệu mẫu (seeding), kiểm tra tính đúng đắn của schema và thực hiện các bài đo hiệu năng (benchmarking) hệ thống.

- `/data` : Chứa các bộ dữ liệu tĩnh như ngân hàng câu hỏi Big Five và các hằng số tính điểm.
- `/model` : Chứa notebook Jupyter (`pca_evaluator.ipynb`) được sử dụng để phân tích dữ liệu khám phá (EDA) và tính toán các trọng số toán học (Mean, Components) cho mô hình PCA-4.
- `/assets` : Lưu trữ tài nguyên tĩnh của ứng dụng (hình ảnh, icon, âm thanh) và mô hình machine learning thu gọn (`pca_evaluator_4d.tflite`).

B. Quy trình phân tích và tính toán trọng số PCA

Mô hình PCA-4 trong hệ thống không được xây dựng như một mô hình học sâu (deep learning) “hộp đen”, mà là một phép biến đổi toán học xác định. Quy trình tìm ra các hệ số biến đổi được thực hiện trong file `model/pca_evaluator.ipynb` với các bước chính sau:

1. **Phân tích dữ liệu khám phá (EDA)**: Sử dụng tập dữ liệu công khai gồm hơn 300.000 bản ghi để khảo sát phân phối của 5 đặc điểm tính cách Big Five. Kết quả cho thấy các trait có phân phối chuẩn và ổn định trên nhiều quốc gia.
2. **Tính toán hằng số trung bình (Mean)**: Xác định giá trị trung bình của từng trait trên quy mô lớn để làm điểm gốc cho phép trừ chuẩn hóa ($x - \mu$).
3. **Trích xuất ma trận thành phần (Components)**: Sử dụng thuật toán Phân tích Thành phần chính (Principal Component Analysis) để tìm ra ma trận chiếu W giúp giữ lại hơn 90% phương sai dữ liệu trong khi giảm từ 5 chiều xuống 4 chiều.

Các tham số này (μ và W) sau đó được viết trực tiếp vào mã nguồn của service `pcaEvaluator.ts` để thực hiện tính toán ngay trên thiết bị người dùng mà không cần kết nối internet hay máy chủ xử lý ML phức tạp. Quy trình này đảm bảo tính minh bạch, tốc độ xử lý tức thời và bảo vệ quyền riêng tư tuyệt đối.

C. Bộ câu hỏi Big Five (IPIP-50) Anh - Việt

Danh sách đầy đủ 50 câu hỏi Big Five được trích xuất từ bộ dấu hiệu IPIP (International Personality Item Pool) [9] kèm theo bản tạm dịch tiếng Việt được sử dụng trong hệ thống. Dữ liệu này được trình bày tại [Bảng 1.1](#):

Bảng 1.1 — Danh sách 50 câu hỏi Big Five (IPIP-50) Anh - Việt

STT	Phiên bản Tiếng Anh	Bản tạm dịch Tiếng Việt
1.	I am the life of the party.	Tôi là trung tâm của bữa tiệc.
2.	I feel little concern for others.	Tôi ít quan tâm đến người khác.
3.	I am always prepared.	Tôi luôn chuẩn bị sẵn sàng.
4.	I get stressed out easily.	Tôi dễ bị căng thẳng.
5.	I have a rich vocabulary.	Tôi có vốn từ vựng phong phú.
6.	I don't talk a lot.	Tôi không nói nhiều.
7.	I am interested in people.	Tôi quan tâm đến mọi người.
8.	I leave my belongings around.	Tôi hay để đồ đặc lung tung.
9.	I am relaxed most of the time.	Tôi thấy thư giãn hầu hết thời gian.
10.	I have difficulty understanding abstract ideas.	Tôi gặp khó khăn khi hiểu các ý tưởng trừu tượng.
11.	I feel comfortable around people.	Tôi cảm thấy thoải mái khi ở gần mọi người.
12.	I insult people.	Tôi xúc phạm người khác.
13.	I pay attention to details.	Tôi chú ý đến chi tiết.
14.	I worry about things.	Tôi lo lắng về mọi thứ.
15.	I have a vivid imagination.	Tôi có trí tưởng tượng sống động.
16.	I keep in the background.	Tôi thích ở phía sau.
17.	I sympathize with others' feelings.	Tôi thông cảm với cảm xúc của người khác.
18.	I make a mess of things.	Tôi làm mọi thứ rối tung lên.
19.	I seldom feel blue.	Tôi hiếm khi cảm thấy buồn.

STT	Phiên bản Tiếng Anh	Bản tạm dịch Tiếng Việt
20.	I am not interested in abstract ideas.	Tôi không quan tâm đến các ý tưởng trừu tượng.
21.	I start conversations.	Tôi bắt đầu cuộc trò chuyện.
22.	I am not interested in other people's problems.	Tôi không quan tâm đến vấn đề của người khác.
23.	I get chores done right away.	Tôi hoàn thành công việc ngay lập tức.
24.	I am easily disturbed.	Tôi dễ bị xáo trộn.
25.	I have excellent ideas.	Tôi có những ý tưởng tuyệt vời.
26.	I have little to say.	Tôi có ít điều để nói.
27.	I have a soft heart.	Tôi có trái tim mềm mỏng.
28.	I often forget to put things back in their proper place.	Tôi thường quên để đồ đạc vào đúng chỗ.
29.	I get upset easily.	Tôi dễ nỗi cáu.
30.	I do not have a good imagination.	Tôi không có trí tưởng tượng tốt.
31.	I talk to a lot of different people at parties.	Tôi nói chuyện với nhiều người khác nhau trong bữa tiệc.
32.	I am not really interested in others.	Tôi thực sự không quan tâm đến người khác.
33.	I like order.	Tôi thích sự ngăn nắp.
34.	I change my mood a lot.	Tôi thay đổi tâm trạng nhiều.
35.	I am quick to understand things.	Tôi hiểu mọi thứ nhanh chóng.

STT	Phiên bản Tiếng Anh	Bản tạm dịch Tiếng Việt
36.	I don't like to draw attention to myself.	Tôi không thích thu hút sự chú ý.
37.	I take time out for others.	Tôi dành thời gian cho người khác.
38.	I shirk my duties.	Tôi trốn tránh trách nhiệm.
39.	I have frequent mood swings.	Tôi có những thay đổi tâm trạng thường xuyên.
40.	I use difficult words.	Tôi sử dụng từ ngữ khó.
41.	I don't mind being the center of attention.	Tôi không ngại là trung tâm chú ý.
42.	I feel others' emotions.	Tôi cảm nhận được cảm xúc của người khác.
43.	I follow a schedule.	Tôi tuân theo lịch trình.
44.	I get irritated easily.	Tôi dễ bị khó chịu.
45.	I spend time reflecting on things.	Tôi dành thời gian suy ngẫm về mọi thứ.
46.	I am quiet around strangers.	Tôi im lặng khi ở gần người lạ.
47.	I make people feel at ease.	Tôi làm cho mọi người cảm thấy thoải mái.
48.	I am exacting in my work.	Tôi nghiêm khắc trong công việc.
49.	I often feel blue.	Tôi thường cảm thấy buồn.
50.	I am full of ideas.	Tôi tràn đầy ý tưởng.

D. Mã nguồn thực nghiệm và các hàm quan trọng

Phần này trình bày mã nguồn của các kịch bản thực nghiệm và hàm thực thi biên (Edge Functions) quan trọng. Toàn bộ các hình khôi mã nguồn dưới đây đã được cấu hình để có thể ngắt trang tự động nhằm đảm bảo tính toàn vẹn của dữ liệu.

D.1. Kịch bản kiểm chứng độ tương đồng PCA (`pca_worst_best_case.ts`)

Kịch bản này được sử dụng để kiểm chứng giả thuyết RQ1 tại Chương 6. Script tạo ra các cặp dữ liệu trả lời mẫu để mô phỏng các kịch bản từ tương đồng tuyệt đối, đối lập hoàn toàn, đến gần giống nhau, sau đó tính toán và in ra kết quả tương đồng.

Thuật toán 1.1 — Minh họa các kịch bản kiểm thử trong `pca_worst_best_case.ts`

```
// Cấu trúc một kịch bản kiểm thử trong
1 `pca_worst_best_case.ts` ts
2 const scenarios = [
3   {
4     name: 'Same pattern (5,4,3,2) vs (5,4,3,2)',
5     answersA: patternA,
6     answersB: patternB,
7   },
8   {
9     name: 'Opposite pattern (A vs 6-A by item)',
10    answersA: patternA,
11    answersB: oppositePatternB,
12  },
13  {
14    name: 'Shifted pattern (5,4,3,2) vs (4,3,2,1)',
15    answersA: patternA,
16    answersB: shiftedPatternB,
17  },
18  // ... các kịch bản khác
19 ];
20
21 // Vòng lặp thực thi
22 for (const scenario of scenarios) {
23   await evaluateScenario(
```

```

24     scenario.name,
25     questionSet,
26     scenario.answersA,
27     scenario.answersB
28   );
29 }

```

D.2. Mã hóa và Giải mã dữ liệu (score-crypto)

Hàm xử lý việc bảo mật dữ liệu nhạy cảm sử dụng thuật toán AES-256-GCM tại Edge.

Thuật toán 1.2 — Hiện thực hàm mã hóa và giải mã điểm tính cách

```

1 import { serve } from 'https://deno.land/std@0.192.0/http/
  server.ts';
ts

2 // ... (Logic helpers toBytes, fromBase64, toBase64)
3

4 async function importKey() {
5   const secret = Deno.env.get('B5_ENCRYPTION_KEY');
6   const keyBytes = fromBase64(secret);
7   return crypto.subtle.importKey('raw', keyBytes, 'AES-GCM',
8     false, ['encrypt', 'decrypt']);
9 }

10 async function encrypt(data: unknown) {
11   const key = await importKey();
12   const iv = crypto.getRandomValues(new Uint8Array(12));
13   const payload = toBytes(JSON.stringify(data));
14   const cipher = await crypto.subtle.encrypt({ name: 'AES-GCM',
15     iv }, key, payload);
16   return { cipher: toBase64(cipher), iv: toBase64(iv) };
17 }

18 serve(async (req) => {
19   const { mode, scores, payload } = await req.json();
20   if (mode === 'encrypt') {
21     const res = await encrypt(scores || payload);

```

```

22     return new Response(JSON.stringify(res), { headers:
23       { 'Content-Type': 'application/json' } });
24   }
25 });

```

D.3. Thuật toán Giới thiệu người dùng lai (recommend-users)

Hàm thực hiện xếp hạng ứng viên dựa trên PCA, ELO và Hobbies.

Thuật toán 1.3 — Hiện thực hàm giới thiệu người dùng lai

```

1 // ... imports
2 function eloProximity(rA: number, rB: number, sigma = 400) {
3   return Math.exp(-Math.abs(rA - rB) / sigma);
4 }
5
6 serve(async (req) => {
7   const { userId, useElo, useHobbies } = await req.json();
8   // ... (Data fetching for me and candidates)
9
10  const score = (pcaSim, prox, hSim) => {
11    if (useElo && useHobbies) return 0.5 * pcaSim + 0.2 * prox
12    + 0.3 * hSim;
13    if (useElo) return 0.8 * pcaSim + 0.2 * prox;
14    return pcaSim;
15  };
16  // ... (Mapping and sorting logic)
17 });

```

D.4. Kịch bản đo hiệu năng hệ thống (Benchmarks)

Kịch bản `scripts/benchmark_scenarios.ts` đã được cập nhật để đo lường chi tiết độ trễ của các thành phần Edge Functions riêng lẻ, phục vụ dữ liệu cho RQ3.

Thuật toán 1.4 — Kịch bản đo hiệu năng chi tiết

```

1 async function benchmark() {

```

```

2      // 1. LOGIN
3      const tLoginStart = performance.now();
4      await supabase.auth.signInWithEmailAndPassword({ email:
5          'viewer@test.com', password: '....' });
6          console.log(`[Login] Latency: ${performance.now() -
7              tLoginStart}ms`);
8
9      // 2. EDGE FUNCTIONS LATENCY
10     const tCryptoStart = performance.now();
11     await supabase.functions.invoke('score-crypto', { body:
12         { mode: 'encrypt', scores: dummy } });
13     console.log(`[score-crypto] Latency: ${performance.now() -
14         tCryptoStart}ms`);
15
16     // 3. RECOMMENDATION
17     const tRecStart = performance.now();
18     const { data: recData } = await
19     supabase.functions.invoke('recommend-users', { ... });
20     console.log(`[Recommend] Latency: ${performance.now() -
21         tRecStart}ms`);

22 }

```

D.5. Kịch bản kiểm thử suy luận gián tiếp qua Edge Functions

Kịch bản `scripts/demo/penetration_edge_inference.ts` mô phỏng tình huống kẻ tấn công chỉ biết base URL của Edge Functions và đoán được payload, từ đó thử gọi các endpoint công khai mà không kèm thông tin xác thực.

Thuật toán 1.5 — Kịch bản kiểm thử suy luận gián tiếp qua Edge Functions

```
1 // 1. Xây dựng base URL của edge
```

ts

```

2 const EDGE_BASE = `${SUPABASE_URL}/functions/v1`;
3
4 // 2. Gửi payload đoán trước tới recommend-users
5 await fetch(`#${EDGE_BASE}/recommend-users`, {
6   method: 'POST',
7   headers: { 'Content-Type': 'application/json' },
8   body: JSON.stringify({ userId: '0000...', pageSize: 5, useElo:
9     true, useHobbies: true }),
10
11 // 3. Gửi payload đoán trước tới embed
12 await fetch(`#${EDGE_BASE}/embed`, {
13   method: 'POST',
14   headers: { 'Content-Type': 'application/json' },
15   body: JSON.stringify({ text: 'Hiking, Anime' }),
16 });

```

D.6. Kỹ thuật tối ưu hóa cơ sở dữ liệu

Mã nguồn SQL được sử dụng để tối ưu hoá hiệu năng cho các chính sách bảo mật hàng (RLS) và tăng tốc độ truy vấn thông qua chỉ mục (Index) được trình bày tại [Thuật toán 1.6](#):

Thuật toán 1.6 — Mã nguồn SQL tối ưu hoá cơ sở dữ liệu

```

1 -- 1. Tối ưu hoá RLS bằng cách sử dụng subquery để cache
2 auth.uid()
3
4 DROP POLICY IF EXISTS "profiles_is_owner" ON public.profiles;
5 CREATE POLICY "profiles_is_owner_optimized" ON public.profiles
6 FOR ALL USING ( id = (select auth.uid()) );
7
8 -- 2. Gộp các chính sách SELECT thừa trên bảng matches
9 DROP POLICY IF EXISTS "match_select" ON public.matches;
10 CREATE POLICY "matches_select_optimized" ON public.matches
11 FOR SELECT USING (
12   user_a = (select auth.uid()) OR
13   user_b = (select auth.uid())
14 );

```

```

13
14    -- 3. Bộ sung chỉ mục (Index) cho các cột lọc và liên kết quan trọng
15    CREATE INDEX IF NOT EXISTS idx_profiles_id ON
16        public.profiles(id);
17    CREATE INDEX IF NOT EXISTS idx_matches_user_a ON
18        public.matches(user_a);
19    CREATE INDEX IF NOT EXISTS idx_matches_user_b ON
20        public.matches(user_b);

```

D.7. Minh họa kịch bản kiểm thử logic tính điểm

Để đảm bảo logic tính điểm Big Five trên client (TypeScript) hoạt động chính xác như logic gốc (Python), các kịch bản kiểm thử đã được thiết lập. Cấu trúc các kịch bản trong `scripts/score_verifier.ts` được tóm tắt trong thuật toán [Thuật toán 1.7](#).

Thuật toán 1.7 — Minh họa các kịch bản kiểm thử trong `score_verifier.ts`

```

1 // Cấu trúc một kịch bản kiểm thử trong `score_verifier.ts` ts
2 const scenarios = [
3     {
4         label: 'Dữ liệu giả lập từ notebook',
5         responses: [3, 5, 1, 4, ...], // Mảng gồm 50 câu trả lời
6         expectedScaled: {
7             Extraversion: 0.425,
8             Agreeableness: 0.275,
9             Conscientiousness: 0.575,
10            'Emotional Stability': 0.525,
11            'Intellect/Imagination': 0.475,
12        },
13    },
14    {
15        label: 'Trường hợp biên: Tất cả trả lời trung lập',
16        responses: Array(50).fill(3),
17        expectedScaled: {
18            Extraversion: 0.5,
19            Agreeableness: 0.5,

```

```

20     Conscientiousness: 0.5,
21     'Emotional Stability': 0.5,
22     'Intellect/Imagination': 0.5,
23   },
24 },
25 // ... các kịch bản khác cho trường hợp "hoàn toàn đồng ý" và
26 "không đồng ý"
27 ];

```

D.8. Kịch bản kiểm chứng độ tương đồng ngữ nghĩa

Kịch bản `scripts/demo_semantic_comparison.ts` được sử dụng để kiểm chứng giả thuyết RQ4. Script so sánh các cặp người dùng với các nhóm sở thích (themes) khác nhau bằng cả tiếng Anh và tiếng Việt để đánh giá khả năng của mô hình trong việc nhận diện sự tương đồng về ngữ nghĩa.

Thuật toán 1.8 — Minh họa các kịch bản trong `demo_semantic_comparison.ts`

```

1 // Cấu trúc dữ liệu người dùng trong script
2 const users = [
3   { name: 'User 1 (EN Outdoor)', hobbies: ['Hiking', 'Camping',
4     'Mountain Biking'] },
5   { name: 'User 2 (VI Outdoor)', hobbies: ['Leo núi', 'Cắm
6     trại', 'Chụp ảnh thiên nhiên'] },
7   { name: 'User 3 (EN Creative)', hobbies: ['Painting', 'Digital
8     Art', 'Reading Novels'] },
9   { name: 'User 4 (VI Creative)', hobbies: ['Vẽ tranh', 'Đọc
10    tiểu thuyết', 'Chơi piano'] },
11   { name: 'User 5 (EN Tech)', hobbies: ['Programming', 'Building
12    PCs', 'Sci-Fi Movies'] },
13   { name: 'User 6 (VI Foodie)', hobbies: ['Nâú ăn', 'Làm bánh',
14     'Xem phim Hàn Quốc'] },
15 ];
16
17 // Vòng lặp so sánh tất cả các cặp
18 for (let i = 0; i < users.length; i++) {
19   for (let j = i + 1; j < users.length; j++) {

```

```
14      // ... tính toán và in ra độ tương đồng  
15  }  
16 }
```