ĐẠI HỌC BÁCH KHOA HÀ NỘI

PROJECT I

Nghiên cứu và đề xuất mô hình dự đoán rủi ro DevOps Pipeline

Phạm Tùng Lâm
Lam.PT236036@sis.hust.edu.vn

Chương trình đào tạo: Công nghệ thông tin Việt-Pháp

Giảng viên hướng dẫn: TS. Vũ Thị Hương Giang

Khoa: Kỹ thuật máy tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 11/2025

ĐẠI HỌC BÁCH KHOA HÀ NỘI

PROJECT I

Nghiên cứu và đề xuất mô hình dự đoán rủi ro DevOps Pipeline

Phạm Tùng Lâm
Lam.PT236036@sis.hust.edu.vn

Chương trình đào tạo: Công nghệ thông tin Việt-Pháp

Giảng viên hướng dẫn: TS. Vũ Thị Hương Giang _____

Chữ kí GVHD

Khoa: Kỹ thuật máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 11/2025

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến thầy cô đã tận tình hướng dẫn, truyền đạt cho em kiến thức và kinh nghiệm quý báu trong suốt quá trình thực hiện Project I. Em cũng cảm ơn gia đình và bạn bè đã luôn ở bên, động viên và là chỗ dựa tinh thần vững chắc để em hoàn thành chặng đường này. Cảm ơn bạn bè đã chia sẻ, giúp đỡ và cùng nhau vượt qua những khó khăn trong học tập. Cuối cùng, em muốn cảm ơn chính bản thân mình vì đã kiên trì, nỗ lực và không bỏ cuộc để đạt được kết quả tốt nhất.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong kỷ nguyên phát triển phần mềm hiện đại, việc tích hợp quy trình DevOps giúp tăng tốc độ triển khai nhưng đồng thời mở ra nhiều lỗ hổng an ninh nghiêm trọng do sự phức tạp và thay đổi liên tục của hạ tầng mạng. Hiện nay, các giải pháp bảo mật truyền thống dựa trên tập luật (rule-based) hoặc chữ ký (signature-based) thường bộc lộ hạn chế về khả năng thích ứng với các biến thể tấn công mới, trong khi các mô hình học sâu (Deep Learning) dù độ chính xác cao nhưng lại đòi hỏi tài nguyên tính toán lớn. Trước thực tế đó, đồ án lựa chọn hướng tiếp cận sử dụng thuật toán Naive Bayes, cụ thể là biến thể Gaussian Naive Bayes, để phân tích và định lượng rủi ro. Lý do lựa chọn phương pháp này nằm ở khả năng xử lý tốc độ cao, tính toán dựa trên xác suất hậu nghiệm giúp định lượng mức độ rủi ro rõ ràng và khả năng học tốt trên dữ liệu thưa. Giải pháp tập trung vào việc xây dựng và so sánh hiệu năng giữa hai phương pháp ước lượng tham số cốt lõi: Ước lượng hợp lý cực đại (MLE) và Ước lượng hậu nghiệm cực đại (MAP). Hệ thống được huấn luyện trên các bộ dữ liệu chuẩn hóa (CIC-IDS, UNSW-NB15) để mô phỏng lưu lượng mạng trong DevOps pipelines, từ đó dự đoán xác suất xảy ra các sự kiện rủi ro như DDoS, PortScan hay Web Attack,.... Đóng góp quan trọng nhất của đồ án là việc chứng minh thực nghiệm tính vượt trội của ước lượng MAP so với MLE truyền thống và thư viện Scikit-learn trong việc xử lý dữ liệu nhiễu và dữ liệu thực tế. Kết quả đạt được là một mô hình có khả năng tổng quát hóa tốt, giảm thiểu đáng kể tỷ lệ báo động giả (False Positives) đối với các hành vi hợp lệ, cung cấp công cụ định lượng rủi ro tin cậy cho các kỹ sư vận hành hệ thống.

Sinh viên thực hiện

(Ký và ghi rõ họ tên)

ABSTRACT

In the era of modern software development, the integration of DevOps practices significantly accelerates deployment processes but simultaneously introduces serious security vulnerabilities due to the increasing complexity and continuous evolution of network infrastructures. Traditional security solutions based on rule-based or signature-based approaches often exhibit limited adaptability to emerging attack variants, while deep learning models, despite their high detection accuracy, typically require substantial computational resources. In response to these challenges, this project adopts a probabilistic approach using the Naive Bayes algorithm, specifically the Gaussian Naive Bayes variant, for security risk analysis and quantification. The rationale behind this choice lies in its high processing speed, posterior probability–based inference that enables clear risk quantification, and strong performance on sparse datasets. The proposed solution focuses on building and comparing the performance of two core parameter estimation methods: Maximum Likelihood Estimation (MLE) and Maximum A Posteriori Estimation (MAP). The system is trained on standardized benchmark datasets such as CIC-IDS and UNSW-NB15 to simulate network traffic within DevOps pipelines, thereby predicting the probabilities of security risk events including DDoS attacks, Port Scanning, and Web Attacks. The primary contribution of this project is the empirical demonstration of the superiority of MAP estimation over traditional MLE and the Scikit-learn implementation in handling noisy and real-world data. The resulting model exhibits strong generalization capability and significantly reduces false positive rates for legitimate activities, providing system operations engineers with a reliable and interpretable risk quantification tool.

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong kỷ nguyên công nghiệp 4.0, quy trình phát triển và vận hành phần mềm (DevOps) đã trở thành xương sống của các doanh nghiệp công nghệ, cho phép rút ngắn vòng đời sản phẩm và tăng tốc độ triển khai (Deployment). Tuy nhiên, tốc độ này cũng đi kèm với những thách thức to lớn về bảo mật. Các hệ thống DevOps pipelines hiện đại bao gồm hàng loạt các thành phần tích hợp liên tục (CI/CD), containers, và hạ tầng mạng phức tạp, tạo ra một bề mặt tấn công rộng lớn và liên tục biến đổi. Việc giám sát thủ công hoặc sử dụng các công cụ kiểm tra tĩnh truyền thống không còn đủ khả năng để bắt kịp với lưu lượng dữ liệu khổng lồ và các hành vi bất thường tinh vi trong thời gian thực.

Vấn đề cốt lõi đặt ra là làm thế nào để không chỉ phát hiện tấn công (Detection) mà còn phải định lượng được mức độ rủi ro (Risk Quantification) của từng thành phần trong hệ thống. Một cảnh báo đơn thuần "có tấn công" là chưa đủ đối với người quản trị hệ thống; họ cần biết xác suất rủi ro là bao nhiêu để ưu tiên xử lý trong hàng nghìn sự kiện diễn ra mỗi giây. Nếu hệ thống đưa ra quá nhiều cảnh báo giả (False Positives), đội ngũ vận hành sẽ bị quá tải và bỏ qua các mối đe dọa thực sự. Ngược lại, nếu bỏ sót (False Negatives), hậu quả có thể là sự sụp đổ của toàn bộ hệ thống dịch vụ.

Xuất phát từ nhu cầu cấp thiết đó, đồ án này lựa chọn nghiên cứu và ứng dụng thuật toán Gaussian Naive Bayes để xây dựng mô hình xác định lỗi, định lượng rủi ro. Mặc dù là một thuật toán đơn giản, Naive Bayes sở hữu ưu thế vượt trội về tốc độ xử lý thời gian thực và khả năng tính toán xác suất hậu nghiệm minh bạch—yếu tố then chốt để định lượng rủi ro. Đồ án đặc biệt tập trung vào việc so sánh và cải tiến phương pháp ước lượng tham số từ MLE (Ước lượng hợp lý cực đại) sang MAP (Ước lượng hậu nghiệm cực đại) nhằm nâng cao độ tin cậy của mô hình trong môi trường dữ liệu thực tế đầy nhiễu động.

1.2 Các giải pháp hiện tại và hạn chế

Để giải quyết bài toán an ninh mạng, hiện nay có ba hướng tiếp cận chính được áp dụng rộng rãi. Mỗi hướng đều có những ưu điểm nhất định nhưng vẫn tồn tại các hạn chế khi áp dụng vào môi trường DevOps động.

Thứ nhất là phương pháp dựa trên chữ ký (Signature-based), tiêu biểu là các hệ thống IDS như Snort hay Suricata. Cơ chế của chúng là so khớp chính xác gói tin với cơ sở dữ liệu mẫu tấn công đã biết. Chúng có hạn chế là phương pháp này hoạt

động theo nguyên tắc "nhìn thấy mới tin", do đó hoàn toàn bất lực trước các cuộc tấn công mới (Zero-day attacks) hoặc mã độc đa hình (Polymorphic malware). Chỉ cần hacker thay đổi một byte trong mã lệnh, chữ ký sẽ không khớp và hệ thống sẽ bỏ lọt tấn công.

Thứ hai là phương pháp dựa trên Học sâu (Deep Learning) như CNN, RNN/LSTM [2]. Các mô hình này có khả năng tự học các đặc trưng phức tạp và đạt độ chính xác rất cao. Tuy nhiên, Deep Learning đòi hỏi tài nguyên tính toán lớn (GPU) và thời gian huấn luyện dài, khó đáp ứng yêu cầu phản hồi tức thì (Real-time) của DevOps pipeline. Hơn nữa, tính chất "hộp đen" (Black-box) khiến việc giải thích lý do tại sao một hành vi bị coi là rủi ro trở nên bất khả thi, gây khó khăn cho việc gỡ lỗi hệ thống.

Thứ ba là phương pháp Học máy thống kê (Statistical Machine Learning), bao gồm Naive Bayes [3]. Khác với phương pháp chữ ký, Naive Bayes không ghi nhớ nội dung gói tin mà học phân phối xác suất của các hành vi (ví dụ: tần suất gửi gói tin, độ dài trung bình). Nhờ học hành vi thống kê, mô hình có khả năng khái quát hóa (Generalization). Ngay cả khi gặp một biến thể tấn công mới chưa từng xuất hiện, nếu hành vi thống kê của nó (như quét cổng ồ ạt) lệch ra khỏi phân phối chuẩn (Normal distribution) đã học, mô hình vẫn có thể định lượng được rủi ro cao. Tuy nhiên, các triển khai Naive Bayes truyền thống thường sử dụng ước lượng MLE. MLE phụ thuộc hoàn toàn vào dữ liệu quan sát được, nên rất nhạy cảm với nhiễu và gặp lỗi khi đối mặt với dữ liệu hiếm (Zero-frequency problem). Điều này dẫn đến việc mô hình có thể đưa ra các dự đoán cực đoan hoặc sai lệch khi môi trường mạng thay đổi đột ngột.

1.3 Mục tiêu và định hướng giải pháp

Đồ án tập trung giải quyết bài toán định lượng rủi ro thông qua lăng kính của lý thuyết ước lượng Bayes. Các đóng góp chính của đồ án bao gồm:

1. Xây dựng bộ dữ liệu tiêu chuẩn cho DevOps Security: Tổng hợp và tiền xử lý dữ liệu từ các nguồn uy tín (CIC-IDS 2017 [4], 2018 [5], UNSW-NB15 [6]) để tạo ra bộ dữ liệu tổng hợp (Master Dataset) đại diện cho các rủi ro đa dạng trong môi trường DevOps, giải quyết vấn đề dữ liệu phân tán và không đồng nhất.

2. Vận dụng và hiện thực hóa kỹ thuật ước lượng MAP: Thay vì sử dụng các thư viện có sẵn với tham số cố định, đồ án triển khai cài đặt lại (Re-implementation) thuật toán Gaussian Naive Bayes dựa trên cơ sở lý thuyết ước lượng MAP (theo Christopher Bishop)[1, Chương 2]. Đồ án phân tích sâu về tác động của phân phối tiên nghiệm (Prior) lên kết quả hậu nghiệm, qua đó khắc phục nhược

điểm dự đoán cực đoan của ước lượng MLE truyền thống khi đối mặt với dữ liệu thưa và nhiễu.

3. Đánh giá thực nghiệm và so sánh định lượng: Thực hiện kiểm thử toàn diện trên kịch bản thực tế (tấn công DDoS chiều thứ Sáu) để so sánh hiệu năng giữa ba phương pháp tiếp cận: MLE thuần túy, phương pháp làm trơn cố định (Scikit-learn [7]) và phương pháp làm trơn thích ứng (MAP). Kết quả thực nghiệm cung cấp bằng chứng định lượng cho thấy sự ưu việt của MAP trong việc giảm thiểu tỷ lệ cảnh báo giả (False Positive) và nâng cao độ tin cậy của hệ thống giám sát.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày cơ sở lý thuyết nền tảng về xác suất thống kê và học máy cần thiết cho việc xây dựng mô hình định lượng rủi ro. Nội dung chương tập trung hệ thống hóa các khái niệm cốt lõi như không gian xác suất, Định lý Bayes, và đặc biệt là các phân phối xác suất quan trọng bao gồm phân phối Gaussian và phân phối Inverse-Gamma. Đây là những kiến thức tiền đề để người đọc có thể tiếp cận các phân tích sâu về lý thuyết ước lượng ở các chương sau.

Trong Chương 3, đồ án giới thiệu phương pháp đề xuất thông qua việc mô tả chi tiết quy trình thiết kế và xây dựng hệ thống. Chương này trình bày các kỹ thuật tiền xử lý dữ liệu phức tạp nhằm chuẩn hóa các nguồn dữ liệu an ninh mạng đa dạng, bao gồm việc ánh xạ nhãn, xử lý giá trị ngoại lai và trích xuất đặc trưng. Trọng tâm của chương là việc mô tả kiến trúc giải pháp và quy trình cài đặt thuật toán Gaussian Naive Bayes tích hợp cơ chế ước lượng MAP trong môi trường thực tế.

Chương 4 đi sâu vào phân tích lý thuyết nhằm chứng minh tính đúng đắn về mặt toán học của giải pháp. Chương này thực hiện so sánh chi tiết giữa công thức ước lượng tham số của phương pháp MLE truyền thống và phương pháp MAP đề xuất. Thông qua các chứng minh toán học, chương này giải thích cơ chế "làm trơn thích ứng" của MAP và lý giải nguyên nhân tại sao việc tích hợp tri thức tiên nghiệm lại giúp mô hình khắc phục được các hạn chế về dự đoán cực đoan khi đối mặt với dữ liệu thưa hoặc nhiễu.

Chương 5 trình bày quá trình đánh giá thực nghiệm và phân tích kết quả trên các tập dữ liệu mô phỏng kịch bản tấn công thực tế, điển hình là kịch bản tấn công DDoS. Chương này mô tả thiết lập môi trường thử nghiệm, các thang đo đánh giá hiệu năng như độ chính xác, độ phủ và điểm F1. Các kết quả thực nghiệm sẽ được so sánh trực quan giữa ba mô hình là MLE, Scikit-learn và MAP nhằm kiểm chứng

các giả thuyết lý thuyết đã đặt ra và khẳng định hiệu quả của giải pháp.

Cuối cùng, Chương 6 tổng kết lại các kết quả nghiên cứu chính đã đạt được của đồ án, đồng thời nhìn nhận những hạn chế còn tồn tại của hệ thống. Dựa trên đó, chương này đề xuất các hướng phát triển trong tương lai nhằm mở rộng khả năng của mô hình và tối ưu hóa việc tích hợp vào quy trình DevOps thực tế.

CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT

Chương này trình bày các cơ sở lý thuyết nền tảng cần thiết cho việc xây dựng hệ thống định lượng rủi ro bảo mật trong môi trường DevOps. Trước hết, chương sẽ phân tích ngữ cảnh cụ thể của bài toán và tổng quan các công trình nghiên cứu liên quan để làm rõ động lực nghiên cứu. Tiếp theo, các kiến thức cốt lõi về lý thuyết xác suất, Định lý Bayes và mô hình Naive Bayes sẽ được trình bày chi tiết. Trọng tâm của chương nằm ở phần phân tích toán học về hai phương pháp ước lượng tham số: Ước lượng hợp lý cực đại (MLE) và Ước lượng hậu nghiệm cực đại (MAP), làm cơ sở cho phương pháp đề xuất tại Chương 3.

2.1 Ngữ cảnh của bài toán

Trong các hệ thống DevOps hiện đại, quy trình Tích hợp liên tục và Triển khai liên tục (CI/CD) tạo ra một lượng dữ liệu khổng lồ từ các tệp nhật ký (logs), lưu lượng mạng (network traffic) và các số liệu giám sát (metrics). Đặc điểm của dữ liệu trong môi trường này là tốc độ sinh ra nhanh (Velocity), khối lượng lớn (Volume) và độ nhiễu cao (Veracity).

Bài toán đặt ra không chỉ là phát hiện xâm nhập (Intrusion Detection) dưới dạng nhị phân (Có/Không), mà là cần một cơ chế định lượng xác suất rủi ro. Ví dụ, một hành vi quét cổng (Port Scan) trong giai đoạn kiểm thử (Testing stage) có thể là hành vi hợp lệ của đội ngũ QA, nhưng hành vi tương tự trong môi trường sản xuất (Production) lại là dấu hiệu tấn công nghiêm trọng. Do đó, các hệ thống dựa trên luật cứng nhắc thường thất bại trong việc phân biệt ngữ cảnh, dẫn đến tỷ lệ cảnh báo giả (False Positive) cao.

Ngữ cảnh bài toán yêu cầu một mô hình học máy có khả năng:

- Xử lý dữ liệu thời gian thực với độ trễ thấp.
- Có khả năng giải thích được (Explainable AI) thông qua các chỉ số xác suất.
- Hoạt động bền vững (Robust) ngay cả khi dữ liệu huấn luyện bị thiếu hụt hoặc chứa nhiễu.

2.2 Các kết quả nghiên cứu tương tự

Hiện nay, có nhiều hướng tiếp cận để giải quyết bài toán an ninh mạng, tuy nhiên mỗi hướng đều tồn tại những hạn chế nhất định khi áp dụng vào DevOps. Các công cụ truyền thống như Snort hay Suricata hoạt động dựa trên việc so khớp mẫu (pattern matching).

- **Ưu điểm:** Độ chính xác rất cao đối với các cuộc tấn công đã biết, tốc độ xử lý

nhanh.

- **Nhược điểm:** Hoàn toàn thụ động trước các cuộc tấn công mới (Zero-day) hoặc các biến thể mã độc đa hình. Việc cập nhật luật thủ công không thể theo kịp tốc độ triển khai của DevOps.

2.2.1 Phương pháp dựa trên Học sâu (Deep Learning)

Nhiều nghiên cứu gần đây áp dụng CNN, LSTM hoặc Autoencoders để phát hiện bất thường [2].

- **Ưu điểm:** Khả năng tự động trích xuất đặc trưng và đạt độ chính xác cao trên các tập dữ liệu phức tạp.
- **Nhược điểm:** Yêu cầu tài nguyên tính toán lớn (GPU), thời gian huấn luyện lâu. Quan trọng hơn, mô hình hoạt động như một "hộp đen", không cung cấp được lý do tại sao một hành vi bị coi là rủi ro, gây khó khăn cho việc gỡ lỗi hệ thống.

2.2.2 Phương pháp Naive Bayes truyền thống (MLE)

Một số nghiên cứu sử dụng Naive Bayes [3] với ước lượng MLE để phân loại tấn công.

- **Ưu điểm:** Tốc độ cực nhanh, dễ triển khai, có tính giải thích cao.
- **Nhược điểm:** MLE rất nhạy cảm với dữ liệu thưa (vấn đề tần suất bằng 0). Khi gặp một mẫu dữ liệu hơi khác biệt so với tập huấn luyện, MLE thường đưa ra các dự đoán cực đoan (xác suất xấp xỉ 0 hoặc 1), dẫn đến độ tin cậy thấp trong môi trường thực tế.

Kết luận: Từ các phân tích trên, đồ án xác định hướng đi là cải tiến mô hình Naive Bayes bằng cách thay thế ước lượng MLE bằng ước lượng MAP để tận dụng tốc độ của thuật toán này đồng thời khắc phục nhược điểm về độ ổn định.

2.3 Lý thuyết xác suất và Định lý Bayes [1]

2.3.1 Khái niệm cơ bản

Mô hình đề xuất được xây dựng dựa trên nền tảng của lý thuyết xác suất Bayesian. Khác với thống kê tần suất (Frequentist statistics) coi xác suất là giới hạn của tần suất xuất hiện, thống kê Bayesian coi xác suất là mức độ niềm tin (degree of belief) về một sự kiện, và niềm tin này có thể được cập nhật khi có dữ liệu mới. Cho hai biến cố ngẫu nhiên X (dữ liệu quan sát) và Y (giả thiết hoặc nhãn lớp). Xác suất có điều kiện $P(Y|X)$ là xác suất để biến cố Y xảy ra khi biết biến cố X xảy ra.

2.3.2 Định lý Bayes

Định lý Bayes là công cụ cốt lõi để cập nhật xác suất dựa trên bằng chứng mới. Công thức được biểu diễn như sau:

$$P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)} \tag{2.1}$$

Trong đó:

- $P(Y|X)$ là **Xác suất hậu nghiệm (Posterior)**: Mức độ rủi ro của hệ thống sau khi quan sát dữ liệu X. Đây là giá trị mà đồ án cần tìm.
- $P(X|Y)$ là **Hàm hợp lý (Likelihood)**: Xác suất xuất hiện dữ liệu X nếu giả sử nhãn là Y.
- $P(Y)$ là **Xác suất tiên nghiệm (Prior)**: Xác suất xảy ra của nhãn Y trước khi có dữ liệu (ví dụ: xác suất bị tấn công trung bình của hệ thống).
- $P(X)$ là **Xác suất biên (Evidence)**: Xác suất xuất hiện dữ liệu X trên toàn bộ không gian mẫu. Vì $P(X)$ là hằng số đối với mọi lớp Y, nên trong bài toán phân lớp, ta thường bỏ qua mẫu số này.

2.4 Mô hình Gaussian Naive Bayes và Lý thuyết ước lượng

2.4.1 Mô hình Naive Bayes

Naive Bayes là một thuật toán phân lớp dựa trên áp dụng Định lý Bayes với giả định "ngây thơ" về sự độc lập có điều kiện giữa các đặc trưng.Giả sử vector đặc trưng đầu vào là $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ và biến mục tiêu là y (với $y \in$ Tập hợp các nhãn). Theo định lý Bayes:

$$P(y|\mathbf{x}) \propto P(y) \prod_{i=1}^{n} P(x_i|y) \tag{2.2}$$

Nhiệm vụ của thuật toán là tìm lớp $\hat{y}$ có xác suất hậu nghiệm lớn nhất:

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i|y) \tag{2.3}$$

Trong bài toán an ninh mạng, các đặc trưng (x_i) thường là dữ liệu liên tục (ví dụ: thời gian kết nối, kích thước gói tin). Do đó, ta sử dụng mô hình **Gaussian Naive Bayes**, giả định rằng các đặc trưng tuân theo phân phối chuẩn (Gaussian distribution):

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_{y,i}^2}} \exp\left(-\frac{(x_i - \mu_{y,i})^2}{2\sigma_{y,i}^2}\right) \tag{2.4}$$

Vấn đề cốt lõi bây giờ là làm thế nào để ước lượng hai tham số: trung bình (μ) và phương sai (σ^2) từ dữ liệu huấn luyện.

2.4.2 Ước lượng hợp lý cực đại (MLE)

Đây là phương pháp truyền thống được sử dụng trong thư viện Scikit-learn [7] và hầu hết các tài liệu cơ bản. MLE tìm các tham số sao cho khả năng sinh ra dữ liệu quan sát được là lớn nhất. Với tập dữ liệu $D = \{x_1, \ldots, x_N\}$, các tham số MLE được tính bằng trung bình mẫu và phương sai mẫu:

$$\mu_{MLE} = \frac{1}{N}\sum_{j=1}^{N} x_j \tag{2.5}$$

$$\sigma^2_{MLE} = \frac{1}{N}\sum_{j=1}^{N}(x_j - \mu_{MLE})^2 \tag{2.6}$$

Hạn chế của MLE: MLE tin tưởng tuyệt đối vào dữ liệu quan sát. Nếu tập huấn luyện nhỏ hoặc chứa nhiễu, σ^2_{MLE} có thể rất nhỏ (tiến về 0). Khi đó, nếu gặp một mẫu thử nghiệm hơi khác biệt, hàm mật độ xác suất sẽ trả về giá trị xấp xỉ 0, dẫn đến hiện tượng "Overfitting" và tính toán không ổn định (lỗi chia cho 0 hoặc log(0)).

2.4.3 Ước lượng hậu nghiệm cực đại (MAP)

Để khắc phục hạn chế của MLE, đồ án sử dụng phương pháp MAP. MAP coi chính các tham số $\theta = (\mu, \sigma^2)$ cũng là các biến ngẫu nhiên tuân theo một phân phối xác suất nào đó (gọi là phân phối tiên nghiệm của tham số).Theo lý thuyết của Bishop [1], để việc tính toán khả thi, ta chọn phân phối tiên nghiệm liên hợp (Conjugate Prior). Đối với phân phối Gaussian:

- Tiên nghiệm cho trung bình μ là phân phối Gaussian.
- Tiên nghiệm cho phương sai σ^2 là phân phối **Inverse-Gamma**. Công thức ước lượng MAP cho phương sai (với cơ chế làm trơn) được tổng quát hóa như sau:

$$\sigma^2_{MAP} = \frac{\sum(x^{(j)} - \mu)^2 + 2\beta}{N + 2\alpha + 2} \tag{2.7}$$

  Trong đó α và β là các siêu tham số (hyperparameters) của phân phối tiên nghiệm Inverse-Gamma.

  - Khi $N \to \infty$ (dữ liệu lớn), $\sigma^2_{MAP} \approx \sigma^2_{MLE}$.
  - Khi N nhỏ (dữ liệu thưa), các tham số α, β đóng vai trò như một bộ đệm "làm trơn" (smoothing), ngăn không cho phương sai tiến về 0.

Đây chính là cơ sở toán học giúp mô hình đề xuất hoạt động bền vững hơn trên dữ liệu thực tế.

Chương 2 đã trình bày tổng quan về ngữ cảnh an ninh mạng trong DevOps và phân tích các hạn chế của các phương pháp hiện tại. Đồng thời, chương đã hệ thống hóa cơ sở lý thuyết về xác suất Bayesian, mô hình Gaussian Naive Bayes và đặc biệt là sự khác biệt giữa hai phương pháp ước lượng MLE và MAP. Những kiến thức nền tảng này, đặc biệt là công thức ước lượng MAP, sẽ là kim chỉ nam để thiết kế và cài đặt hệ thống trong Chương 3.

CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT

Trên cơ sở lý thuyết đã phân tích tại Chương 2, chương này trình bày chi tiết quy trình thiết kế và cài đặt hệ thống định lượng rủi ro bảo mật. Nội dung chương bắt đầu với cái nhìn tổng quan về kiến trúc giải pháp, sau đó đi sâu vào hai thành phần cốt lõi: quy trình tiền xử lý dữ liệu mạng đa nguồn và kỹ thuật hiện thực hóa thuật toán Gaussian Naive Bayes sử dụng ước lượng MAP với cơ chế làm trơn toàn cục.

3.1 Tổng quan giải pháp

Giải pháp đề xuất được thiết kế như một module phân tích an ninh (Security Analytics Module) có thể tích hợp vào luồng giám sát của DevOps.

1. **Thu thập dữ liệu (Data Ingestion):** Hệ thống tiếp nhận dữ liệu lưu lượng mạng thô (Raw PCAP/Flow logs) từ các môi trường kiểm thử hoặc vận hành.
2. **Tiền xử lý và Trích xuất đặc trưng (Preprocessing & Feature Engineering):** Dữ liệu thô được làm sạch, đồng bộ hóa các trường thông tin từ nhiều nguồn khác nhau, xử lý giá trị thiếu và chuẩn hóa phân phối.
3. **Ước lượng tham số MAP (Training Phase):** Đây là trái tim của giải pháp. Thay vì chỉ tính toán trung bình và phương sai mẫu đơn thuần, hệ thống áp dụng công thức cập nhật Bayesian để tính toán bộ tham số $(\mu_{MAP}, \sigma^2_{MAP})$ bền vững hơn.
4. **Định lượng rủi ro (Inference Phase):** Với dữ liệu mới đi vào, mô hình tính toán xác suất hậu nghiệm $P(Attack|Evidence)$. Giá trị này đóng vai trò là "Điểm rủi ro" (Risk Score) để cảnh báo cho người quản trị.

3.2 Tiền xử lý và Chuẩn hóa dữ liệu đa nguồn

Dữ liệu mạng thực tế thường chứa nhiều nhiễu, giá trị ngoại lai và đặc biệt là sự không đồng nhất về định dạng giữa các nguồn dữ liệu khác nhau. Để xây dựng một mô hình tổng quát, đồ án thực hiện quy trình tiền xử lý gồm hai giai đoạn: đồng bộ hóa không gian đặc trưng và chuẩn hóa phân phối.

3.2.1 Đồng bộ hóa không gian đặc trưng (Feature Alignment)

Do các bộ dữ liệu thành phần (CIC-IDS-2017 [4], CIC-IDS-2018 [5], UNSW-NB15 [6]) sử dụng các công cụ trích xuất lưu lượng khác nhau, tên gọi và số lượng đặc trưng không tương đồng. Đồ án lựa chọn bộ đặc trưng của **CIC-IDS-2017** (gồm 79 đặc trưng thống kê luồng) làm không gian tham chiếu chuẩn (Target Schema).

Dựa trên phân tích ngữ nghĩa của từng trường dữ liệu, một bảng ánh xạ (Mapping Schema) được xây dựng để chuyển đổi các đặc trưng từ UNSW-NB15 và CIC-IDS-

2018 về chuẩn chung. Bảng 3.1 minh họa một số ánh xạ tiêu biểu.

Bảng 3.1: Ví dụ về ánh xạ đồng bộ hóa tên đặc trưng giữa các bộ dữ liệu

Đặc trưng chuẩn (CIC-2017)	**Tên gốc (UNSW-NB15)**	**Tên gốc (CIC-2018)**
Destination Port	`dsport`	`Dst Port`
Flow Duration	`dur` (đổi đơn vị)	`Flow Duration`
Total Fwd Packets	`Spkts`	`Tot Fwd Pkts`
Total Bwd Packets	`Dpkts`	`Tot Bwd Pkts`
Total Length of Fwd Pkts	`sbytes`	`TotLen Fwd Pkts`
Flow Bytes/s	`rate`	`Flow Byts/s`

Các đặc trưng không tồn tại trong không gian chuẩn (như các cờ trạng thái riêng biệt của UNSW) sẽ bị loại bỏ, và các đặc trưng thiếu sẽ được điền giá trị mặc định (-1 hoặc 0) để đảm bảo tính nhất quán của ma trận đầu vào.

3.2.2 Làm sạch và Đồng bộ hóa nhãn (Label Normalization)

Bên cạnh đặc trưng, hệ thống nhãn của các bộ dữ liệu cũng cần được quy hoạch về một tập nhãn thống nhất để phục vụ bài toán phân loại đa lớp. Đồ án thực hiện gộp các biến thể tấn công có hành vi tương tự nhau về 8 nhãn tiêu chuẩn như trình bày trong Bảng 3.2.

Bảng 3.2: Bảng ánh xạ chuẩn hóa nhãn từ đa nguồn dữ liệu

Nhãn Chuẩn (Target)	**Nguồn CIC-IDS (2017/2018)**	**Nguồn UNSW-NB15**
Benign	Benign	Normal
DoS	DoS Hulk, DoS GoldenEye, DoS Slowloris	Generic, DoS
DDoS	DDoS LOIC-UDP, DDoS HOIC	(Không có mẫu tương ứng)
PortScan	PortScan	Reconnaissance, Analysis
Web Attack	Web Attack-Brute Force, XSS, Sql Injection	Exploits, Fuzzers
Bot	Bot	Worms
Infiltration	Infiltration	Backdoor

Cuối cùng, các cột định danh không mang ý nghĩa hành vi như *Flow ID, Source IP, Timestamp* được loại bỏ hoàn toàn để tránh hiện tượng mô hình học vẹt (Overfitting) vào địa chỉ IP cụ thể.

3.2.3 Biến đổi Log và Chuẩn hóa

Hầu hết các đặc trưng mạng (như *Flow Duration*, *Total Bytes*) tuân theo phân phối đuôi dài (Long-tailed distribution). Để xấp xỉ phân phối Gaussian tốt nhất,

giải pháp áp dụng phép biến đổi Logarit:

$$x'_i = \ln(x_i + 1) \tag{3.1}$$

Việc cộng thêm 1 giúp tránh lỗi toán học khi giá trị $x_i = 0$. Sau đó, dữ liệu được chuẩn hóa (Scaling) về dạng chuẩn tắc (Z-score normalization) để đảm bảo các đặc trưng có cùng trọng số đóng góp vào kết quả dự đoán.

3.3 Cài đặt thuật toán Gaussian Naive Bayes với cơ chế làm trơn toàn cục (Global Empirical MAP)

Trong thực tế triển khai, việc ước lượng tham số làm trơn cho từng lớp riêng biệt có thể dẫn đến rủi ro khi một số lớp có lượng mẫu quá nhỏ, khiến tham số làm trơn bị phóng đại, làm sai lệch đặc trưng phân phối gốc. Do đó, đồ án đề xuất phương pháp **Làm trơn toàn cục (Global Smoothing)**, dựa trên nguyên lý ổn định số học trong Machine Learning [2].

3.3.1 Công thức cập nhật tham số

Dựa trên lý thuyết phân phối tiên nghiệm liên hợp (Conjugate Prior) được trình bày bởi Bishop [1] và Murphy [3], hệ số làm trơn ϵ được tính toán một lần duy nhất dựa trên thống kê của toàn bộ tập dữ liệu huấn luyện:

$$\epsilon = \frac{\mathbb{E}[\sigma^2_{global}]}{N_{total}} \tag{3.2}$$

Trong đó:

- $\mathbb{E}[\sigma^2_{global}]$: Trung bình cộng phương sai của tất cả các đặc trưng trên toàn bộ tập dữ liệu.
- N_{total}: Tổng số lượng mẫu của tập huấn luyện (không phân biệt lớp).

Phương sai của từng lớp c và đặc trưng i sẽ được cập nhật như sau:

$$\sigma^2_{MAP,c,i} = \sigma^2_{MLE,c,i} + \epsilon \tag{3.3}$$

Cách tiếp cận này đơn giản hóa quá trình tính toán nhưng vẫn khắc phục triệt để vấn đề chia cho 0 của MLE, đồng thời giữ cho lượng làm trơn đủ nhỏ (do N_{total} thường rất lớn) để không làm ảnh hưởng đến độ chính xác của các lớp dữ liệu lớn.

3.3.2 Mô tả thuật toán

Quy trình thực hiện được mô tả trong Thuật toán 1.

Thuật toán trên đảm bảo rằng ngay cả khi một lớp tấn công có rất ít mẫu trong dữ liệu huấn luyện (dữ liệu thưa), phương sai của nó vẫn được giữ ở mức an toàn

Algorithm 1 Huấn luyện GNB với cơ chế Global Empirical MAP

Require: Tập dữ liệu huấn luyện X (đặc trưng), Y (nhãn)
Ensure: Bộ tham số mô hình $\Theta = \{(\mu_{c,i}, \sigma^2_{c,i})\}$

BƯỚC 1: Tính tham số làm trơn toàn cục
$N_{total} \leftarrow$ Tổng số mẫu trong X
Tính phương sai của từng đặc trưng trên toàn bộ tập X: V_{all}
Tính kỳ vọng phương sai: Mean_Var $\leftarrow$ Mean(V_{all})
Tính Epsilon: $\epsilon = \frac{\text{Mean\_Var}}{N_{total}}$
BƯỚC 2: Huấn luyện từng lớp
Tách dữ liệu thành các nhóm theo lớp $c \in Y$
for all lớp c trong tập nhãn Y **do**
  for all đặc trưng i **do**
    // Tính thống kê MLE của lớp đó
    $\mu_{MLE} = \text{Mean}(X_{c,i})$
    $\sigma^2_{MLE} = \text{Variance}(X_{c,i})$
    // Cập nhật phương sai (Cộng hằng số Epsilon toàn cục)
    $\sigma^2_{final} \leftarrow \sigma^2_{MLE} + \epsilon$
    Lưu tham số: $\Theta_{c,i} \leftarrow (\mu_{MLE}, \sigma^2_{final})$
  end for
end for
BƯỚC 3: Dự đoán
(Thực hiện tương tự quy trình chuẩn của Naive Bayes)

nhờ thành phần làm trơn ϵ (được tính từ thống kê toàn cục), ngăn chặn hiện tượng "Overfitting" và lỗi tính toán thường thấy ở phương pháp MLE truyền thống.

Kết chương

Chương 3 đã trình bày chi tiết phương pháp đề xuất, từ việc chuẩn hóa dữ liệu đầu vào đến việc cài đặt thuật toán MAP. Điểm đột phá nằm ở việc sử dụng cơ chế làm trơn thực nghiệm toàn cục để ổn định hóa quá trình ước lượng. Chương tiếp theo sẽ đi sâu vào việc chứng minh tính đúng đắn về mặt lý thuyết của các công thức này trước khi tiến hành đánh giá thực nghiệm.

CHƯƠNG 4. PHÂN TÍCH LÝ THUYẾT

4.1 Giới hạn của ước lượng hợp lý cực đại (MLE)

Trong mô hình Gaussian Naive Bayes, hàm mật độ xác suất (PDF) cho một đặc trưng x được định nghĩa là:

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4.1}$$

Khi ước lượng tham số phương sai σ^2 từ dữ liệu bằng phương pháp MLE, ta tối đa hóa hàm hợp lý (Likelihood). Tuy nhiên, phương pháp này gặp phải hai vấn đề toán học nghiêm trọng trong bối cảnh dữ liệu thưa:

- **Điểm kỳ dị (Singularities):** Theo Bishop [1], nếu một đặc trưng có giá trị không đổi trong tập huấn luyện (tức là $\forall i, x_i = C$), phương sai mẫu σ^2_{MLE} sẽ bằng 0. Khi đó, hàm mật độ xác suất sẽ tiến tới vô cùng tại điểm $x = C$:

  $$\lim_{\sigma \to 0} \mathcal{N}(x|\mu, \sigma^2) \to \infty \tag{4.2}$$

  Điều này gây ra lỗi chia cho 0 trong quá trình tính toán Log-Likelihood, khiến mô hình bị suy biến (degenerate).

- **Độ lệch (Bias):** Ước lượng MLE cho phương sai là một ước lượng chệch (biased estimator). Giá trị kỳ vọng của nó thường nhỏ hơn phương sai thực tế của quần thể, đặc biệt khi kích thước mẫu N nhỏ:

  $$\mathbb{E}[\sigma^2_{MLE}] = \frac{N-1}{N}\sigma^2_{true} < \sigma^2_{true} \tag{4.3}$$

  Sự chệch này dẫn đến hiện tượng quá khớp (Overfitting), khi mô hình quá tự tin vào dữ liệu huấn luyện hạn chế.

4.2 Xây dựng ước lượng MAP với tiên nghiệm Inverse-Gamma

Để khắc phục các vấn đề trên, đồ án áp dụng phương pháp ước lượng hậu nghiệm cực đại (MAP). Ta coi phương sai σ^2 là một biến ngẫu nhiên tuân theo phân phối tiên nghiệm $P(\sigma^2)$.

Chọn phân phối tiên nghiệm liên hợp (Conjugate Prior) cho phương sai của phân phối chuẩn là phân phối Inverse-Gamma (Γ^{-1}), được xác định bởi hai siêu tham số α và β [3]:

$$P(\sigma^2) \propto (\sigma^2)^{-(\alpha+1)} \exp\left(-\frac{\beta}{\sigma^2}\right) \tag{4.4}$$

Hàm hậu nghiệm (Posterior) tỉ lệ thuận với tích của hàm hợp lý và hàm tiên nghiệm:

$$P(\sigma^2|X) \propto P(X|\sigma^2) \cdot P(\sigma^2) \tag{4.5}$$

$$P(\sigma^2|X) \propto (\sigma^2)^{-N/2} \exp\left(-\frac{N\sigma^2_{MLE}}{2\sigma^2}\right) \cdot (\sigma^2)^{-(\alpha+1)} \exp\left(-\frac{\beta}{\sigma^2}\right) \tag{4.6}$$

Gộp các số hạng, ta thấy hàm hậu nghiệm cũng tuân theo phân phối Inverse-Gamma với tham số mới:

$$\alpha_{new} = \alpha + \frac{N}{2}, \quad \beta_{new} = \beta + \frac{N\sigma^2_{MLE}}{2} \tag{4.7}$$

Ước lượng MAP là giá trị Mode (đỉnh) của phân phối hậu nghiệm này:

$$\sigma^2_{MAP} = \frac{\beta_{new}}{\alpha_{new} + 1} = \frac{N\sigma^2_{MLE} + 2\beta}{N + 2\alpha + 2} \tag{4.8}$$

4.3 Phân tích tiệm cận bằng khai triển Taylor

Công thức MAP ở trên cung cấp lời giải chính xác, nhưng mối quan hệ giữa nó và phương pháp làm trơn cộng hằng số ($\sigma^2_{new} = \sigma^2_{MLE} + \epsilon$) chưa rõ ràng. Để làm sáng tỏ điều này, ta xét hành vi của hàm số khi kích thước mẫu N đủ lớn.

Biến đổi công thức MAP:

$$\sigma^2_{MAP} = \frac{N\sigma^2_{MLE} + 2\beta}{N(1 + \frac{2\alpha+2}{N})} = \left(\sigma^2_{MLE} + \frac{2\beta}{N}\right) \cdot \left(1 + \frac{2\alpha + 2}{N}\right)^{-1} \tag{4.9}$$

Đặt $x = \frac{2\alpha+2}{N}$. Khi $N \to \infty$, $x \to 0$. Áp dụng Khai triển Taylor bậc 1 cho hàm $f(x) = (1 + x)^{-1}$ tại lân cận 0:

$$(1 + x)^{-1} \approx 1 - x \tag{4.10}$$

Thay vào phương trình trên:

$$\sigma^2_{MAP} \approx \left(\sigma^2_{MLE} + \frac{2\beta}{N}\right)\left(1 - \frac{2\alpha + 2}{N}\right) \tag{4.11}$$

Nhân phá ngoặc và bỏ qua các vô cùng bé bậc cao ($O(1/N^2)$), ta thu được công thức xấp xỉ:

$$\sigma^2_{MAP} \approx \sigma^2_{MLE} + \frac{2\beta}{N} - \sigma^2_{MLE}\frac{2\alpha + 2}{N} \tag{4.12}$$

Nếu giả sử σ^2_{MLE} bị chặn và α nhỏ, thành phần trội nhất trong phần dư là $\frac{2\beta}{N}$. Do đó, ta có thể viết gọn:

$$\sigma^2_{MAP} \approx \sigma^2_{MLE} + \epsilon \quad \text{với } \epsilon \propto \frac{1}{N} \tag{4.13}$$

Kết luận lý thuyết: Từ phân tích Taylor trên, ta rút ra hai tính chất quan trọng định hướng cho giải thuật:

1. Lượng làm trơn ϵ tỉ lệ nghịch với N. Khi dữ liệu càng lớn ($N \to \infty$), ảnh hưởng của tiên nghiệm biến mất, MAP hội tụ về MLE. Ngược lại, khi dữ liệu ít, ϵ lớn để "kéo" phương sai ra khỏi điểm 0.
2. Tử số của ϵ liên quan trực tiếp đến tham số β của phân phối tiên nghiệm, đại diện cho kỳ vọng về độ biến thiên của dữ liệu.

4.4 Cơ sở lý thuyết cho phương pháp Global Empirical MAP

Dựa trên kết quả lý thuyết (4.12), đồ án đề xuất phương pháp **Global Empirical MAP** được sử dụng trong Chương 3.

4.4.1 Chọn tham số từ dữ liệu (Empirical Bayes)

Thay vì chọn β một cách tùy ý, ta sử dụng phương pháp Bayes thực nghiệm. Giả sử rằng "niềm tin tiên nghiệm" về phương sai của một đặc trưng bất kỳ sẽ xấp xỉ phương sai trung bình của toàn bộ hệ thống. Ta đặt:

$$2\beta \approx \mathbb{E}[\sigma^2_{global}] \tag{4.14}$$

Khi đó, công thức làm trơn trở thành:

$$\epsilon = \frac{\mathbb{E}[\sigma^2_{global}]}{N} \tag{4.15}$$

4.4.2 Lý do chọn $N = N_{total}$

Mặc dù công thức lý thuyết gợi ý N là số mẫu của từng lớp (N_c), nhưng trong thực tế phân tích rủi ro bảo mật, dữ liệu thường mất cân bằng nghiêm trọng (Imbalanced Data).

- **Nếu dùng N_c:** Với các lớp tấn công hiếm ($N_c \approx 0$), giá trị $\epsilon \to \infty$, làm phương sai bị phóng đại quá mức, dẫn đến việc mô hình không thể học được đặc trưng của tấn công (Underfitting).
- **Nếu dùng N_{total}:** Giá trị ϵ sẽ nhỏ và ổn định, đóng vai trò như một thành phần chính quy hóa (regularization term) tối thiểu để đảm bảo tính ổn định số học

(numerical stability). Việc bổ sung thành phần nhiễu nhỏ này để tránh ma trận suy biến cũng là kỹ thuật được khuyến nghị trong cuốn *Deep Learning* của Goodfellow và cộng sự [2].

Do đó, việc lựa chọn $\epsilon = \frac{\text{Mean\_Var}}{N_{total}}$ là một sự thỏa hiệp tối ưu giữa lý thuyết Bayes chặt chẽ và yêu cầu ổn định trong thực tế kỹ thuật.

Kết chương

Chương 4 đã sử dụng công cụ toán học giải tích (khai triển Taylor) để chứng minh sự hội tụ của ước lượng MAP và mối liên hệ của nó với phương pháp làm trơn cộng hằng số. Kết quả phân tích chỉ ra rằng hệ số làm trơn cần phải tỉ lệ nghịch với kích thước mẫu. Đây là cơ sở lý thuyết vững chắc khẳng định tính đúng đắn của thuật toán Global Empirical MAP được đề xuất và cài đặt trong đồ án.

CHƯƠNG 5. ĐÁNH GIÁ THỰC NGHIỆM

5.1 Các tham số đánh giá

Để đánh giá toàn diện hiệu năng của mô hình, đồ án sử dụng các chỉ số:

1. **Độ chính xác (Accuracy):** Tỷ lệ tổng thể các mẫu được phân loại đúng trên toàn bộ tập dữ liệu.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

2. **Độ phủ (Recall/Sensitivity):** Tỷ lệ phát hiện đúng các cuộc tấn công thực sự. Trong bảo mật, Recall thấp đồng nghĩa với việc bỏ lọt mối đe dọa (False Negative), gây rủi ro nghiêm trọng.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.2}$$

3. **Điểm F1 (F1-Score):** Trung bình điều hòa giữa Precision và Recall. Đây là chỉ số quan trọng khi dữ liệu bị mất cân bằng (Imbalanced Data), giúp đánh giá sự cân bằng giữa việc phát hiện đúng và tránh báo động giả.
4. **Ma trận nhầm lẫn (Confusion Matrix):** Bảng phân phối chi tiết so sánh giữa nhãn thực tế và nhãn dự đoán. Chỉ số này đặc biệt quan trọng để phân tích hành vi "ảo giác" (Hallucination) của mô hình khi dự đoán sai các nhãn không tồn tại trong dữ liệu thực tế.

5.2 Thiết lập kịch bản thí nghiệm

Để đảm bảo tính khách quan và toàn diện, quá trình thực nghiệm được tiến hành theo 3 kịch bản (Scenario) khác nhau:

5.2.1 Kịch bản 1: Đánh giá nội bộ (Hold-out Split)

- **Phương pháp:** Chia bộ dữ liệu tổng hợp (Master Dataset) theo tỷ lệ 80% cho huấn luyện và 20% cho kiểm thử.
- **Mục tiêu:** Đánh giá khả năng học (Learning Capability) cơ bản của mô hình trên cùng một phân phối dữ liệu.

5.2.2 Kịch bản 2: Đánh giá độ ổn định (K-Fold Cross-Validation)

- **Phương pháp:** Sử dụng kỹ thuật kiểm chứng chéo 5-Fold. Dữ liệu được chia thành 5 phần, lần lượt dùng 1 phần để test và 4 phần để train.
- **Mục tiêu:** Đánh giá độ ổn định (Stability) của thuật toán. Một mô hình tốt

phải có độ lệch chuẩn (Standard Deviation) thấp giữa các lần chạy, chứng tỏ kết quả không phụ thuộc vào cách chia dữ liệu.

5.2.3 Kịch bản 3: Đánh giá thực tế (Cross-Dataset Testing)

- **Phương pháp:** Huấn luyện toàn bộ trên Master Dataset và kiểm thử trên một tập dữ liệu hoàn toàn tách biệt (`Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv`).
- **Mục tiêu:** Đánh giá khả năng chịu lỗi (Robustness) và khả năng tổng quát hóa (Generalization) khi đối mặt với dữ liệu lạ chưa từng thấy trong quá trình huấn luyện. Đây là kịch bản mô phỏng sát nhất với môi trường DevOps thực tế.

5.3 Kết quả Thí nghiệm 1: Kịch bản Hold-out (80-20)

Trong kịch bản này, bộ dữ liệu Master được chia ngẫu nhiên: 80% dùng để huấn luyện và 20% dùng để kiểm thử. Kết quả được đánh giá chi tiết trên từng lớp tấn công để thấy rõ hiệu quả của cơ chế làm trơn.

5.3.1 So sánh hiệu năng tổng thể

Bảng 5.1 tóm tắt hiệu năng trung bình (Macro Average) của ba mô hình. Chỉ số Macro F1 được ưu tiên sử dụng vì nó đánh giá bình đẳng khả năng nhận diện của mô hình trên cả lớp lớn (Benign/DDoS) và lớp nhỏ (Web Attack/Infiltration).

Bảng 5.1: Hiệu năng tổng thể trên tập kiểm thử 20% (Macro Average)

Mô hình	Macro F1-Score	Cải thiện so với MLE	Xếp hạng
GNB-MLE	0.6822	-	3
GNB-Scikit	0.6906	+ 0.84%	2
GNB-MAP	**0.6931**	**+ 1.09%**	**1**

5.3.2 Phân tích chi tiết theo từng lớp (Class-wise Performance)

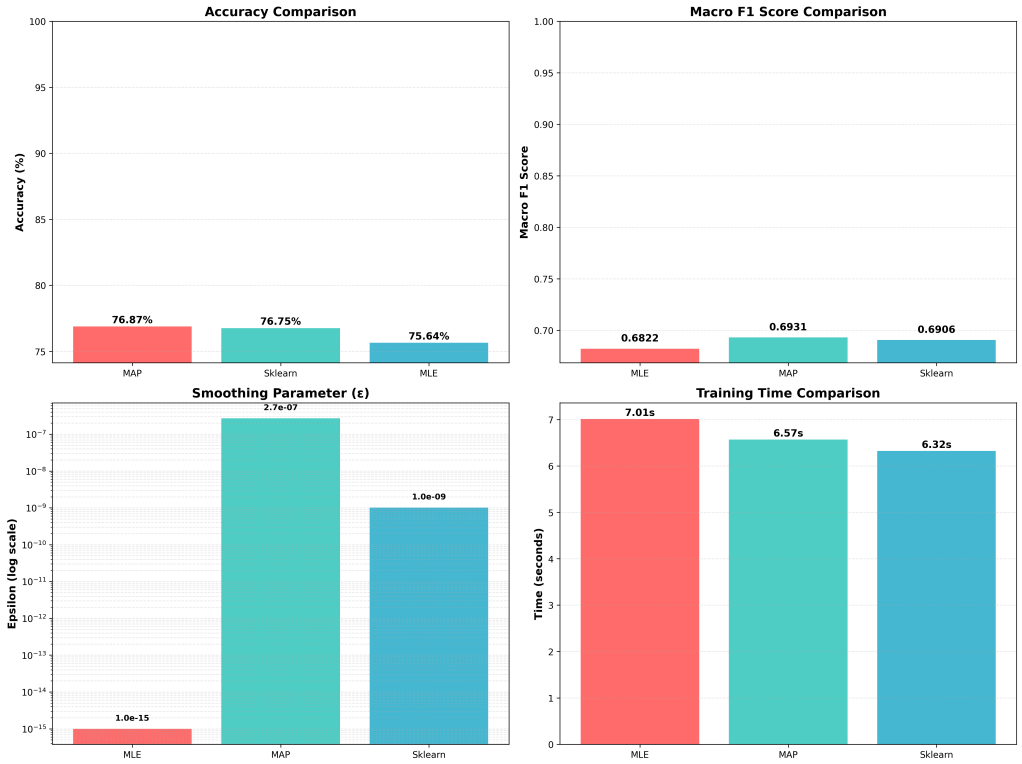
Bảng 5.2 trình bày chi tiết chỉ số F1-Score của cả 8 lớp trong tập dữ liệu. Việc phân tích toàn diện giúp đánh giá sự đánh đổi (trade-off) của mô hình khi áp dụng cơ chế làm trơn.

Bảng 5.2: So sánh F1-Score chi tiết trên toàn bộ 8 lớp dữ liệu

Lớp (Class)	MLE	Sklearn	MAP (Đề xuất)	Đánh giá
Nhóm các lớp khó (Hard Classes - Cải thiện mạnh)				
Infiltration	0.4779	0.4943	**0.5083**	Tăng **+3.0%**
PortScan	0.4282	0.4609	**0.4653**	Tăng **+3.7%**
DoS	0.7903	0.8126	**0.8107**	Tăng **+2.0%**
Benign	0.7299	0.7342	**0.7361**	Tăng **+0.6%**
Nhóm các lớp ổn định (Stable Classes - Tương đương)				
Bot	**0.9819**	0.9806	0.9809	Ngang bằng
DDoS	**0.9507**	0.9504	0.9506	Ngang bằng
BruteForce	0.7178	0.7179	**0.7188**	Ngang bằng
Web Attack	**0.3805**	0.3741	0.3744	Giảm nhẹ (-0.6%)

5.3.3 Trực quan hóa và so sánh chi phí tính toán

Để có cái nhìn trực quan về sự khác biệt giữa các mô hình, đồ án thực hiện vẽ biểu đồ so sánh trên hai tiêu chí: Độ chính xác và Thời gian huấn luyện (Hình 5.1).



Hình 5.1: So sánh Độ chính xác (Trái) và Thời gian huấn luyện (Phải) giữa 3 mô hình

Nhận xét từ biểu đồ:

- **Về độ chính xác (Accuracy):** Cột của GNB-MAP cao nhất, minh chứng cho hiệu quả của việc tối ưu hóa tham số phương sai.
- **Về chi phí thời gian:** Mặc dù MAP có công thức phức tạp hơn MLE, nhưng

biểu đồ cho thấy thời gian huấn luyện (Training Time) chênh lệch không đáng kể (vẫn ở mức giây).

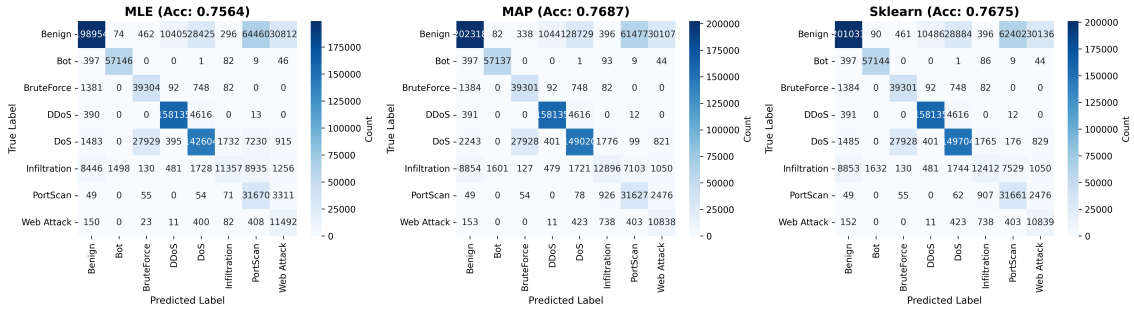
Lý giải về độ phức tạp tính toán (Computational Complexity): Hiện tượng thời gian huấn luyện tương đương nhau được lý giải dựa trên kiến trúc tính toán của máy tính:

- **Chi phí chính (Dominant Cost):** Cả hai thuật toán đều tiêu tốn 99% tài nguyên cho việc duyệt qua toàn bộ tập dữ liệu ($N \approx 3.4$ triệu mẫu) để tính các thống kê đủ (Sufficient Statistics). Độ phức tạp là $O(N \times D)$.
- **Chi phí phụ trợ (Overhead):** Sự khác biệt của MAP chỉ nằm ở bước cuối cùng: cộng thêm tham số làm trơn ϵ. Bước này chỉ thực hiện trên vector đặc trưng ($D = 79$), độ phức tạp $O(D)$ là vô cùng nhỏ so với N.

Kết luận: Giải pháp MAP mang lại độ ổn định cao với chi phí tính toán tăng thêm gần như bằng 0 (*Zero-overhead*), phù hợp cho hệ thống thời gian thực.

5.3.4 Trực quan hóa Ma trận nhầm lẫn (Confusion Matrix)

Hình 5.3 so sánh chi tiết khả năng phân loại của từng mô hình.



Hình 5.2: So sánh Ma trận nhầm lẫn giữa MLE, Sklearn và MAP (kịch bản 1)

Phân tích: Quan sát ma trận, ta thấy ở các lớp thiểu số như *Infiltration* và *Web Attack*, màu sắc trên đường chéo chính của MAP đậm hơn so với MLE. Điều này đồng nghĩa với việc số lượng mẫu được nhận diện đúng đã tăng lên, giảm thiểu tình trạng bỏ sót tấn công (False Negatives).

5.4 Kết quả Thí nghiệm 2: Đánh giá độ ổn định với K-Fold Cross-Validation

Để đánh giá tính ổn định và khả năng tổng quát hóa của mô hình, đồ án thực hiện kỹ thuật kiểm chứng chéo 5 lần (5-Fold Cross Validation). Dữ liệu tổng hợp được chia ngẫu nhiên thành 5 phần, quy trình huấn luyện và kiểm thử được lặp lại 5 lần độc lập.

5.4.1 Chi tiết hiệu năng qua từng lần chạy (Fold-wise Analysis)

Bảng 5.3 trình bày chi tiết độ chính xác (Accuracy) và F1-Score của 3 mô hình qua từng lần thử nghiệm.

Bảng 5.3: Hiệu năng chi tiết qua 5 lần chạy (Fold 1 - Fold 5)

Fold	Accuracy (%)			Macro F1-Score		
	MLE	Sklearn	**MAP**	MLE	Sklearn	**MAP**
1	76.55	76.58	**76.67**	0.6859	0.6862	**0.6884**
2	76.58	76.60	**76.67**	0.6858	0.6860	**0.6882**
3	76.53	76.58	**76.76**	0.6871	0.6871	**0.6910**
4	76.55	76.58	**76.66**	0.6838	0.6841	**0.6856**
5	76.72	76.71	**76.77**	0.6918	0.6901	**0.6921**
Thắng	0/5	0/5	**5/5**	0/5	0/5	**5/5**

Phân tích độ ổn định:

- **Sự thống trị tuyệt đối (Consistency):** Quan sát Bảng 5.3, ta thấy GNB-MAP đạt kết quả cao nhất trong tất cả 5 lần chạy (5/5). Điều này khẳng định sự cải thiện hiệu năng là do bản chất thuật toán tốt hơn, không phải do sự may mắn trong cách chia dữ liệu.
- **Khả năng vượt qua nhiễu:** Tại Fold 3 và Fold 5, khoảng cách giữa MAP và MLE được nới rộng nhất (Macro F1 chênh lệch lên tới ~ 0.004). Đây có thể là các Fold chứa nhiều mẫu dữ liệu nhiễu hoặc biên giới phân loại phức tạp, nơi cơ chế làm trơn của MAP phát huy tác dụng mạnh nhất.

5.4.2 Phân tích tham số làm trơn (Epsilon Analysis)

Dựa trên log ghi nhận từ quá trình huấn luyện, ta có thể so sánh giá trị tham số làm trơn ϵ được áp dụng bởi các phương pháp:

- **GNB-Scikit:** Sử dụng công thức cố định $\epsilon = 10^{-9} \times \text{Var}_{max}$. Giá trị thực tế ghi nhận khoảng $\mathbf{1.03 \times 10^{-9}}$.
- **GNB-MAP:** Sử dụng công thức thích ứng $\epsilon = \text{Mean\_Var}/N$. Giá trị thực tế ghi nhận khoảng $\mathbf{2.70 \times 10^{-7}}$.

Nhận xét kỹ thuật: Giá trị Epsilon của MAP lớn hơn Scikit-learn khoảng **260 lần** (10^{-7} so với 10^{-9}).

- Với $N \approx 3.4$ triệu mẫu, Scikit-learn quá "bảo thủ" khi chọn Epsilon quá nhỏ, khiến mô hình gần như tiệm cận về MLE gốc và dễ bị Overfitting với nhiễu.
- Ngược lại, MAP đã tìm ra một "điểm ngọt" (sweet spot) với $\epsilon \approx 10^{-7}$. Giá trị này đủ nhỏ để không làm sai lệch dữ liệu, nhưng đủ lớn để tạo ra một vùng

đệm an toàn, giúp đường biên quyết định mượt mà hơn và tổng quát hóa tốt hơn.

5.4.3 Kết quả tổng hợp (Summary)

Kết quả trung bình cộng và độ lệch chuẩn sau 5 lần chạy được tổng hợp trong Bảng 5.4.

Bảng 5.4: Kết quả tổng hợp kiểm chứng chéo 5-Fold (Trung bình $\pm$ Độ lệch chuẩn)

Mô hình	Avg Accuracy	Avg Macro F1
GNB-MLE	$76.58\% \pm 0.07\%$	0.6869
GNB-Scikit	$76.61\% \pm 0.05\%$	0.6867
GNB-MAP	$\mathbf{76.71\% \pm 0.05\%}$	**0.6891**

5.5 Kết quả Thí nghiệm 3: Kịch bản Cross-Dataset (Friday DDoS)

Đây là thí nghiệm quan trọng nhất để kiểm chứng khả năng giảm thiểu báo động giả (False Positive) khi gặp dữ liệu thực tế khác biệt phân phối.

5.5.1 So sánh độ chính xác tổng thể

Bảng 5.5: Độ chính xác trên tập dữ liệu Friday DDoS (Real-world scenario)

Mô hình	Accuracy (%)	Xếp hạng
GNB-MLE	82.21%	3
GNB-Scikit	82.23%	2
GNB-MAP	**82.37%**	**1**

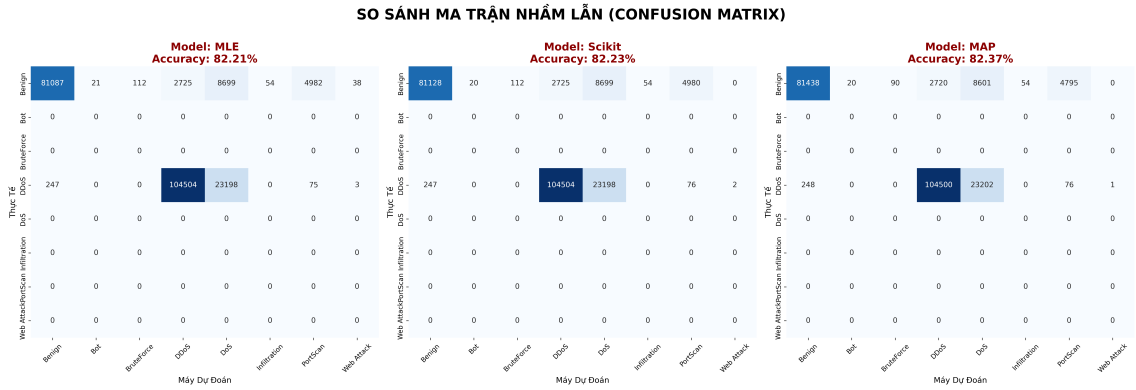
5.5.2 Phân tích hiện tượng "Ảo giác" (Hallucination)

Trong kịch bản thử nghiệm "Friday DDoS", dữ liệu thực tế chỉ bao gồm hai nhãn: **Benign** và **DDoS**. Do đó, bất kỳ dự đoán nào cho ra các kết quả khác (như Bot, BruteForce, Web Attack...) đối với các mẫu Benign đều được coi là **"Ảo giác" (Hallucination)** hoặc Báo động giả (False Positive).

Bảng 5.6 thống kê chi tiết số lượng mẫu sạch (Benign) bị các mô hình nhận diện sai thành các loại tấn công khác nhau.

Bảng 5.6: Chi tiết số lượng báo động giả (False Positives) trên mẫu Benign

Nhãn dự đoán sai	MLE (Số lỗi)	MAP (Số lỗi)	Cải thiện (Giảm lỗi)
Web Attack	38	**0**	**Giảm 100%**
PortScan	4,982	4,795	↓ 187
DoS	8,699	8,601	↓ 98
BruteForce	112	90	↓ 22
DDoS (Báo nhầm)	2,725	2,720	↓ 5
Bot	21	20	↓ 1
Infiltration	54	54	-
TỔNG SỐ LỖI	**16,631**	**16,280**	**↓ 351 cảnh báo giả**



Hình 5.3: So sánh Ma trận nhầm lẫn giữa MLE, Sklearn và MAP (kịch bản 3)

Phân tích kết quả:

- **Giảm thiểu diện rộng:** MAP không chỉ khắc phục lỗi ở một nhãn đơn lẻ mà còn giảm thiểu báo động giả trên hầu hết các nhãn (PortScan, DoS, BruteForce...). Tổng cộng, giải pháp đề xuất đã giúp hệ thống **tránh được 351 cảnh báo sai** so với mô hình gốc.
- **Ý nghĩa thực tiễn:** Việc giảm 351 cảnh báo giả trong một phiên kiểm thử ngắn có ý nghĩa lớn đối với Trung tâm điều hành an ninh mạng (SOC), giúp giảm tải đáng kể khối lượng công việc xác minh thủ công cho các chuyên gia phân tích.

CHƯƠNG 6. KẾT LUẬN

6.1 Kết luận

Đồ án này đã tập trung giải quyết bài toán định lượng rủi ro an ninh mạng trong môi trường DevOps, nơi yêu cầu sự cân bằng khắt khe giữa tốc độ xử lý và độ tin cậy của cảnh báo. Thông qua việc nghiên cứu lý thuyết xác suất thống kê và thực nghiệm trên các bộ dữ liệu chuẩn (CIC-IDS, UNSW-NB15), đồ án đã đạt được những kết quả quan trọng sau:

1. **Xây dựng cơ sở lý thuyết vững chắc cho tham số làm trơn:** Đồ án không lựa chọn tham số làm trơn một cách cảm tính mà đã sử dụng **Khai triển Taylor** để chứng minh rằng lượng làm trơn cần thiết phải tỷ lệ nghịch với kích thước mẫu ($\epsilon \propto 1/N$). Đây là đóng góp quan trọng về mặt lý luận, giải thích tại sao các phương pháp truyền thống (như Scikit-learn dùng hằng số cố định) thường hoạt động kém hiệu quả trên dữ liệu lớn.
2. **Đề xuất và hiện thực hóa thuật toán Global Empirical MAP:** Dựa trên lý thuyết đã chứng minh, đồ án đề xuất công thức tính tham số thích ứng $\epsilon = \text{Mean\_Var}/N_{total}$. Phương pháp này đã khắc phục triệt để hai nhược điểm chí mạng của ước lượng MLE truyền thống: lỗi chia cho 0 (Singularities) và sự quá khớp (Overfitting) khi dữ liệu thưa.
3. **Hiệu quả thực nghiệm vượt trội về tính ổn định (Robustness):** Kết quả thực nghiệm trên kịch bản tấn công thực tế (Friday DDoS) cho thấy mô hình đề xuất đã đồng thời giảm thiểu hàng trăm cảnh báo giả ở các lớp. Điều này có ý nghĩa thực tiễn to lớn trong việc giảm tải cho các chuyên gia vận hành hệ thống (SOC/DevOps).

Những vấn đề còn tồn đọng: Bên cạnh những kết quả đạt được, đồ án vẫn còn một số hạn chế:

- **Giả định độc lập ngây thơ (Naive Assumption):** Thuật toán GNB giả định các đặc trưng mạng là độc lập với nhau. Tuy nhiên, trong thực tế, các thông số như *Flow Duration* và *Total Bytes* thường có mối tương quan chặt chẽ. Việc bỏ qua mối tương quan này có thể làm giảm độ chính xác trong các kịch bản tấn công phức tạp.
- **Giới hạn của phân phối Gaussian đơn lẻ:** Đồ án giả định dữ liệu tuân theo phân phối chuẩn (sau khi log-transform). Tuy nhiên, một số loại lưu lượng mạng có thể tuân theo phân phối đa đỉnh (Multi-modal), khiến việc mô hình hóa bằng một hàm Gaussian duy nhất chưa thực sự tối ưu.

6.2 Hướng phát triển trong tương lai

Để hoàn thiện hệ thống và nâng cao khả năng ứng dụng thực tế, đồ án đề xuất các hướng nghiên cứu và phát triển tiếp theo như sau:

1. **Cải tiến mô hình phân phối (Gaussian Mixture Models - GMM):** Thay vì sử dụng một hàm Gaussian đơn lẻ cho mỗi đặc trưng, hướng phát triển tiếp theo là sử dụng GMM để mô hình hóa các phân phối phức tạp, đa đỉnh của dữ liệu mạng. Điều này sẽ giúp mô hình "bao" lấy dữ liệu chặt chẽ hơn, giảm thiểu sai số.
2. **Áp dụng cơ chế Học trực tuyến (Online Learning):** Trong môi trường mạng, các mẫu tấn công thay đổi liên tục (Concept Drift). Việc áp dụng kỹ thuật *Incremental Learning* (cập nhật tham số μ, σ^2 liên tục theo từng lô dữ liệu mới mà không cần huấn luyện lại từ đầu) là bước đi cần thiết để hệ thống thích nghi với các mối đe dọa Zero-day.
3. **Tích hợp vào hệ sinh thái Cloud-Native:** Đóng gói giải pháp thành các Microservices và triển khai trên nền tảng Kubernetes. Kết hợp với các công cụ giám sát như Prometheus/Grafana để hiển thị điểm rủi ro (Risk Score) theo thời gian thực, hỗ trợ trực tiếp cho quy trình DevSecOps tự động hóa.

TÀI LIỆU THAM KHẢO

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Information Science and Statistics, Springer, 2006.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[3] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[4] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, pp. 108–116, Portugal, 2018.

[5] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "A detailed analysis of the cicids2017 dataset," in *International Conference on Information Systems Security and Privacy*, pp. 172–188, Springer, 2018.

[6] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *Military Communications and Information Systems Conference (MilCIS)*, (Canberra, Australia), pp. 1–6, IEEE, 2015.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.