Phát hiện URL Phishing dựa trên mô hình BERT

Article · August 2025

DOI: 10.59266/houjs.2025.581

CITATIONS
0

READS
47

4 authors:

Vũ Xuân Hạnh
Hanoi Open University
15 PUBLICATIONS **42** CITATIONS
SEE PROFILE

Trinh Duy Đỗ
Hanoi Open University, Viet Nam
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Son Ngo Van
Hanoi Open University
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Nguyễn Anh Tuấn
Hanoi Open University
1 PUBLICATION **0** CITATIONS
SEE PROFILE

PHÁT HIỆN URL PHISHING DỰA TRÊN MÔ HÌNH BERT

Vũ Xuân Hạnh[1], Đỗ Duy Trinh[1], Ngọ Văn Sơn[2], Nguyễn Anh Tuấn[3]
Email: hanhvx@hou.edu.vn

Ngày tòa soạn nhận được bài báo: 04/12/2024
Ngày phản biện đánh giá: 12/06/2025
Ngày bài báo được duyệt đăng: 27/06/2025

DOI: 10.59266/houjs.2025.581

Tóm tắt: *Trong bối cảnh các cuộc tấn công mạng ngày càng gia tăng và phức tạp, đặc biệt là các hình thức lừa đảo qua không gian mạng, việc phát triển các mô hình phát hiện tấn công là một nhu cầu cấp thiết. Bài báo này đề xuất phương pháp phát hiện URL Phishing dựa trên kiến trúc transformer, so sánh với phương pháp phát hiện dựa trên học máy có giám sát sử dụng đặc trưng. Nhóm tác giả đã trích xuất 36 đặc trưng chia thành hai nhóm chính: đặc trưng URL và đặc trưng Domain. Các thuật toán Random Forest, XGBoost, và mô hình BERT được huấn luyện, kiểm thử và đánh giá trên bộ dữ liệu đa dạng, bao gồm cả URL Phishing, URL Malware và Defacement. Kết quả cho thấy mô hình BERT đạt độ chính xác 99,05%, cùng tỷ lệ phát hiện cao 99,45% với độ ổn định, chứng minh tính hiệu quả của phương pháp dựa trên kiến trúc transformer.*

Từ khóa: *URL Phishing, phát hiện URL Phishing, kiến trúc transformer, BERT, XGBoost, Machine Learning, Random Forest*

I. Đặt vấn đề

Trong những năm gần đây, các cuộc tấn công mạng ngày càng gia tăng cả về số lượng và mức độ tinh vi, trong đó các cuộc tấn công lừa đảo (phishing) thông qua URL trở thành một trong những mối đe dọa nghiêm trọng đối với an ninh mạng. Các URL Phishing thường được thiết kế để giả mạo các trang web hợp pháp nhằm đánh cắp thông tin nhạy cảm như tài khoản đăng nhập, thông tin tài chính hoặc dữ liệu cá nhân của người dùng. Theo báo cáo của Anti-Phishing Working Group (APWG), số lượng các vụ tấn công phishing trên toàn cầu đã tăng mạnh, với hơn 1,2 triệu vụ được ghi nhận trong năm 2024, tăng 15% so với năm trước (APWG Reports). Những cuộc tấn công này không chỉ ảnh hưởng đến cá nhân mà còn gây thiệt hại lớn cho các tổ chức, với tổng thiệt hại tài chính ước tính lên đến hàng tỷ USD mỗi năm (FBI, 2023).

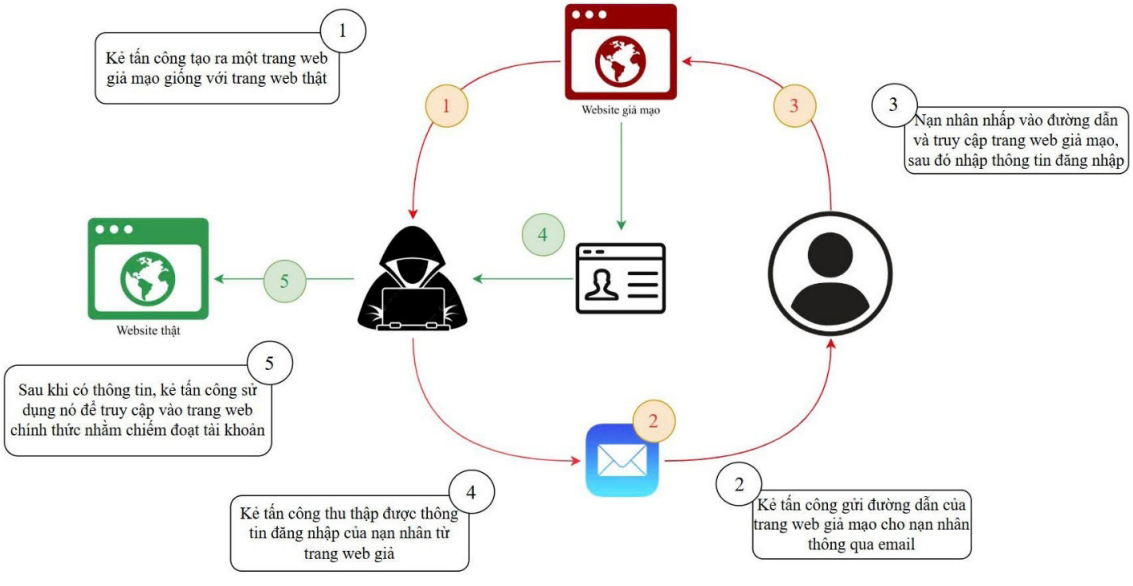
[1] Trường Đại học Mở Hà Nội
[2] Học viên Cao học, Trường Đại học Mở Hà Nội
[3] Sinh viên, Trường Đại học Mở Hà Nội

Các kỹ thuật tấn công URL Phishing ngày càng trở nên phức tạp, bao gồm việc sử dụng URL rút gọn, kỹ thuật che giấu (obfuscation), và thậm chí cả trí tuệ nhân tạo để tạo ra các URL giả mạo khó bị phát hiện bởi các phương pháp truyền thống (Alabdan, 2020). Các phương pháp phát hiện URL Phishing hiện nay, như sử dụng danh sách đen (blacklist), phân tích cú pháp (heuristic analysis), hay học máy (machine learning), đã đạt được một số thành công nhất định. Tuy nhiên, chúng vẫn gặp khó khăn trước sự thay đổi nhanh chóng của các mẫu URL Phishing (Sahoo, Liu, & Hoi, 2017). Đặc biệt, các cuộc tấn công phishing nhắm mục tiêu (spear phishing) và các kỹ thuật tấn công dựa trên tâm lý xã hội (social engineering) đang gia tăng, đòi hỏi những giải pháp phát hiện hiệu quả hơn (Chiew, Yong, & Tan, 2018).

Trong bối cảnh đó, việc nghiên cứu và phát triển các phương pháp phát hiện URL Phishing tiên tiến, đặc biệt là những phương pháp dựa trên kiến trúc transformer và phân tích dữ liệu thời gian thực, trở nên cấp thiết. Bài báo này tập trung phân tích các kỹ thuật phát hiện URL phishing, đồng thời đề xuất hướng tiếp cận mới nhằm nâng cao hiệu quả phát hiện và giảm thiểu rủi ro từ các cuộc tấn công này.

II. Cơ sở lý thuyết

Phishing là một hình thức tấn công lừa đảo nhằm đánh cắp thông tin cá nhân hoặc thông tin nhạy cảm của người dùng thông qua các phương tiện điện tử như email, mạng xã hội hoặc tin nhắn SMS. Một trong những hình thức phổ biến nhất là thông qua các liên kết giả mạo (*phishing URLs*), nơi kẻ tấn công thiết kế các địa chỉ URL có cấu trúc tương tự với các website hợp pháp nhằm đánh lừa người dùng truy cập và cung cấp dữ liệu cá nhân (APWG, 2024).



Hình 1. Cách thức lừa đảo bằng URL Phishing

Trong những năm gần đây, phát hiện URL phishing đã thu hút sự quan tâm rộng rãi từ cộng đồng nghiên cứu. Nổi bật có thể kể đến nghiên cứu của Opara và cộng sự (2024) đề xuất WebPhish - một mô hình học sâu đầu-cuối kết hợp embedding từ nội dung thô của URL và mã HTML, xử lý bằng các lớp CNN, nhằm khai thác đồng thời đặc trưng ngữ nghĩa, cấu trúc và ngữ pháp của website, đạt độ chính xác

98,10%. Trong khi đó, Ghalechyan và cộng sự (2024) tập trung vào hiệu suất mô hình trong môi trường thực tế, bằng cách ứng dụng mạng nơ-ron xác suất (PNN) và mô hình truyền thống trên tập dữ liệu kết hợp từ các nguồn mở (PhishTank, OpenPhish) và dữ liệu thực tế từ EasyDMARC, thu được độ chính xác trung bình 97,00%. Bên cạnh đó, nghiên cứu của Remmide và cộng sự (2022) ứng dụng mạng TCN (Temporal Convolutional Network), một biến thể của CNN có khả năng mô hình hóa chuỗi dài hạn, cùng với embedding Word2Vec và GloVe, để xử lý chuỗi URL. Kết quả thực nghiệm trên ba bộ dữ liệu khác nhau cho thấy mô hình đạt độ chính xác lên đến 98,95% vượt trội so với các mô hình truyền thống như KNN hoặc Logistic Regression, và chứng minh tính hiệu quả của TCN trong nhiệm vụ phân loại URL Phishing. Sánchez-Paniagua và cộng sự (2022) đề xuất bộ dữ liệu PILU-90K với các URL login hợp lệ nhằm phản ánh tình huống thực tế hơn so với các nghiên cứu chỉ sử dụng trang chủ. Họ chứng minh rằng các mô hình huấn luyện trên dữ liệu homepage sẽ giảm độ chính xác khi áp dụng lên URL login, và mô hình TF-IDF kết hợp Logistic Regression đạt kết quả 96,5%. Trong khi đó, Taha và cộng sự (2024) thực hiện so sánh các thuật toán học máy cổ điển trên tập dữ liệu Kaggle, cho thấy Random Forest vượt trội với độ chính xác 96,89% theo sau là XGBoost và Decision Tree. Nhìn chung, các nghiên cứu đã khẳng định tiềm năng vượt trội của các mô hình học sâu, đặc biệt là những mô hình tự động trích chọn đặc trưng từ dữ liệu thô đối với bài toán phát hiện các URL Phishing.

Bảng 1. Một số nghiên cứu trước đây

Tác giả	Phương pháp chính	Đặc trưng	Mô hình chính
Opara và cộng sự (2024)	CNN deep learning với embedding	Raw URL + HTML	CNN (deep, end-to-end)
Ghalechya và cộng sự (2024)	Fine-tuned BERT + Probabilistic NN	Raw URL	BERT-basedNN
Remmide và cộng sự (2022)	Mạng TCN (Temporal Convolutional Network)+Embedding	Chuỗi URL và embedding từ Word2Vec/GloVe	2 lớp TCN + pooling + dense layer
Sánchez-Paniagua và cộng sự (2022)	ML & DL: TF-IDF + Logistic Regression, CNN	URL login thực tế: TF-IDF N-gram và 38 đặc trưng NLP	Logistic Regression, CNN
Taha và cộng sự (2024)	So sánh các mô hình học máy cổ điển	Đặc trưng rút trích từ URL	Random Forest, DecisionTree,XGBoost, LR, AdaBoost

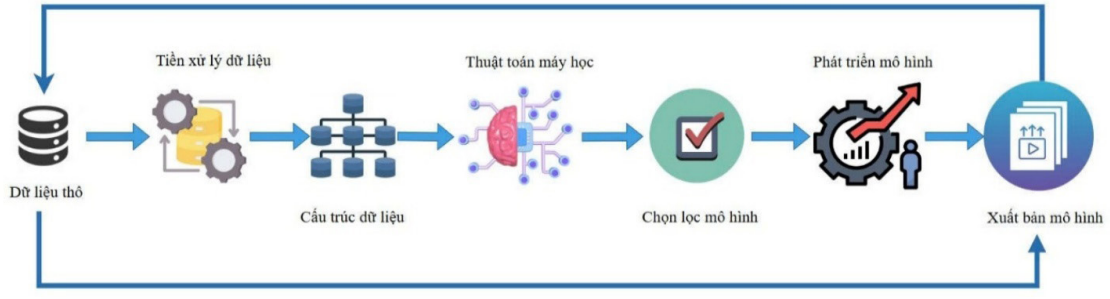
III. Phương pháp nghiên cứu

3.1. Các thuật toán học máy

Trong ba nhóm kỹ thuật chính của học máy gồm: có giám sát, không giám sát và học tăng cường, chúng tôi lựa chọn phương pháp máy học có giám sát do bài toán thuộc nhóm phân loại nhị phân. Phương pháp này sử dụng tập dữ liệu gồm cặp đầu vào - đầu ra huấn luyện mô hình, tối ưu qua hàm mất mát nhằm giảm sai số (Vũ, 2016). Để đảm bảo độ tin cậy, tính chính xác và tránh quá khớp, chúng tôi áp dụng các kỹ thuật như kiểm tra chéo, cân bằng dữ liệu, giảm nhiễu, và đánh giá mô hình bằng ma trận nhầm lẫn.

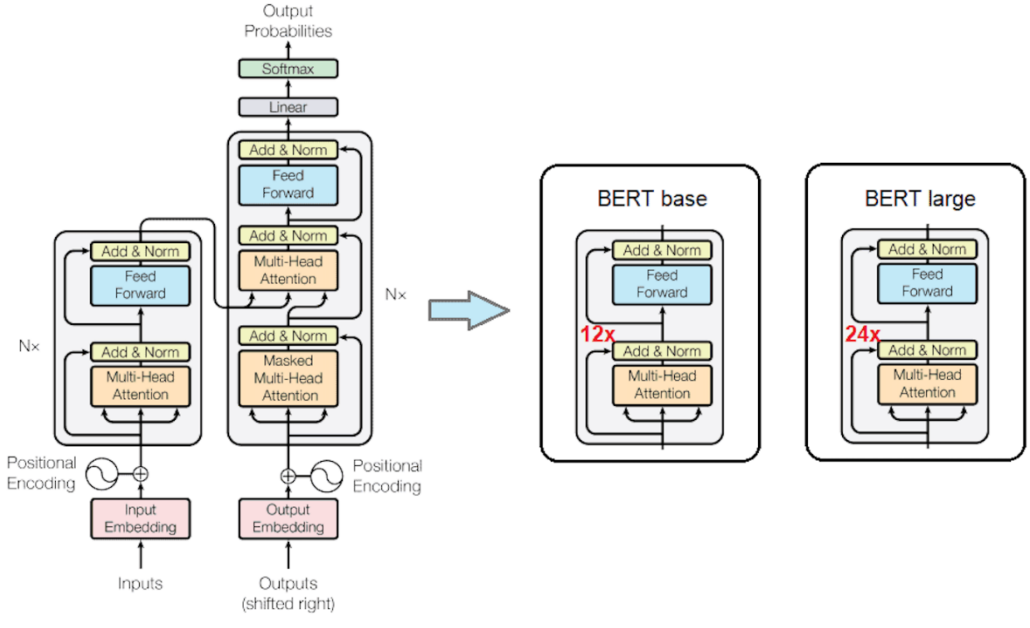
Kỹ thuật máy học có giám sát được nhóm tác giả sử dụng trong nghiên cứu này

là: XGBoost, Random Forest. XGBoost là một thuật toán tăng cường dần (gradient boosting) tối ưu hóa hiệu suất bằng cách kết hợp nhiều cây quyết định yếu, nổi bật với tốc độ nhanh và khả năng tránh quá khớp. Random Forest là mô hình học tổ hợp sử dụng nhiều cây quyết định độc lập để tăng độ chính xác và khả năng tổng quát, đồng thời giảm thiểu rủi ro từ nhiễu dữ liệu.



Hình 2. Tổng quan về quy trình máy học

Bên cạnh học máy có giám sát, nhóm nghiên cứu lựa chọn một mô hình dựa trên kiến trúc transformer, cụ thể là mô hình BERT để có sự so sánh tổng quan về phương pháp học có giám sát truyền thống với mô hình sử dụng kiến trúc transformer hiện đại. BERT (Bidirectional Encoder Representations from Transformers) là một mô hình xử lý ngôn ngữ tự nhiên (NLP) dựa trên kiến trúc transformer, được Google AI giới thiệu vào năm 2018. BERT sử dụng cơ chế attention hai chiều để học biểu diễn ngữ cảnh của từ trong câu, giúp cải thiện đáng kể độ chính xác trong nhiều tác vụ NLP như phân loại văn bản, hỏi đáp, và tìm kiếm thông tin. Mô hình được huấn luyện trước bằng hai nhiệm vụ chính: Masked Language Model (MLM) và Next Sentence Prediction (NSP), giúp nó hiểu rõ hơn về ngữ cảnh của ngôn ngữ (Devlin, Chang, Lee, & Toutanova, 2019).



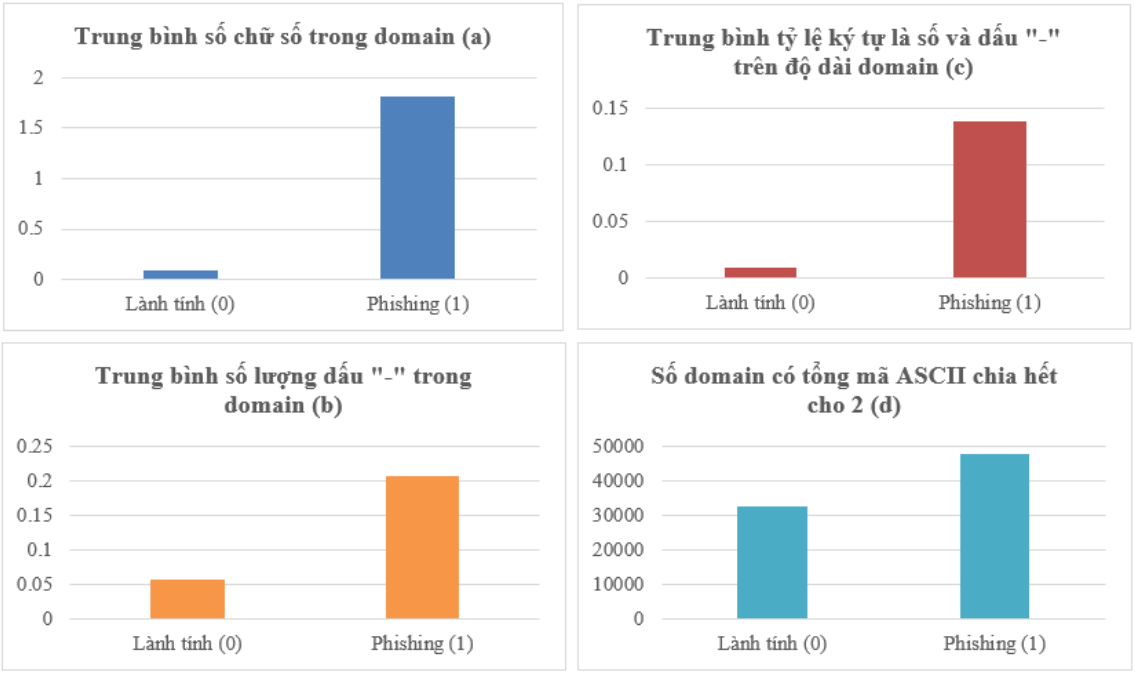
Hình 3. Kiến trúc mô hình BERT được phát triển từ kiến trúc Transformer (Smith, 2024)

3.2. Lựa chọn đặc trưng

Trong nghiên cứu này, đặc trưng được chia thành 2 nhóm: đặc trưng URL và đặc trưng Domain. Nhóm đặc trưng của URL nhóm tác giả sử dụng 20 đặc trưng từ nghiên cứu trước của nhóm (Vũ, Trần, Đỗ, Hoàng, & Ngô, 2022). Các đặc trưng đó như sau:

- urlLength: Độ dài của URL (số ký tự).
- specialCharsCount: Số lượng ký tự đặc biệt (như !@#$%^&*(),.?":{}|<>_=+-).
- hasKeyword: Kiểm tra xem URL có chứa từ khóa phishing phổ biến *(như login, secure, update, verify, account, password, bank, paypal, signin)* hay không (1 có, 0 không).
- hasSpecialChar: Kiểm tra xem URL có chứa ký tự đặc biệt hay không (1 có, 0 không).
- hexCharsCount: Số lượng ký tự hexa (dạng %XX) trong URL.
- digitsCount: Số lượng chữ số trong URL.
- dotCount: Số lượng dấu chấm (.) trong URL.
- slashRatio: Tỉ lệ số dấu gạch chéo (/) so với độ dài URL.
- uppercaseCount: Số lượng chữ cái in hoa trong URL.
- vowelsCount: Số lượng nguyên âm (*a, e, i, o, u,* cả chữ hoa và chữ thường) trong URL.
- consonantsCount: Số lượng phụ âm trong URL.
- domainToUrlRatio: Tỉ lệ độ dài domain so với độ dài toàn bộ URL.
- hasHttpWww: Kiểm tra xem URL có chứa http, https, hoặc www hay không (1 nếu có, 0 nếu không).
- hasIp: Kiểm tra xem URL có chứa địa chỉ IP (IPv4 hoặc IPv6) hay không (1 nếu có, 0 nếu không).
- hasExe: Kiểm tra xem URL có chứa đuôi .exe hay không (1 nếu có, 0 nếu không).
- hasPort: Kiểm tra xem URL có chứa cổng (port) hay không (1 nếu có, 0 nếu không).
- hasBackslash: Kiểm tra xem URL có chứa dấu \ hay không (1 nếu có, 0 nếu không).
- hasRedirect: Kiểm tra xem URL có chứa tham số chuyển hướng (như redirect, url, link, goto, forward) hay không (1 nếu có, 0 nếu không).
- hasRef: Kiểm tra xem URL có chứa tham số ref=, cdm=, referrer=, reference=, hoặc aff= hay không (1 nếu có, 0 nếu không).
- maxSub30: Kiểm tra xem chuỗi con lớn nhất trong URL (phân tách bởi /?&=:#) có độ dài > 30 ký tự hay không (1 nếu > 30, 0 nếu không).

Chúng tôi đề xuất đặc trưng phản ánh các đặc điểm quan trọng có thể phân biệt giữa các domain trong URL lành tính và URL phishing, dựa trên các đặc trưng ngôn ngữ, cấu trúc, thống kê và trực quan hóa dữ liệu được minh họa tại Hình 4. Đặc trưng về ngữ nghĩa, chúng tôi sử dụng bộ từ điển của NLTK. (Vũ, Trần, Đỗ, Hoàng, & Ngô, 2022).

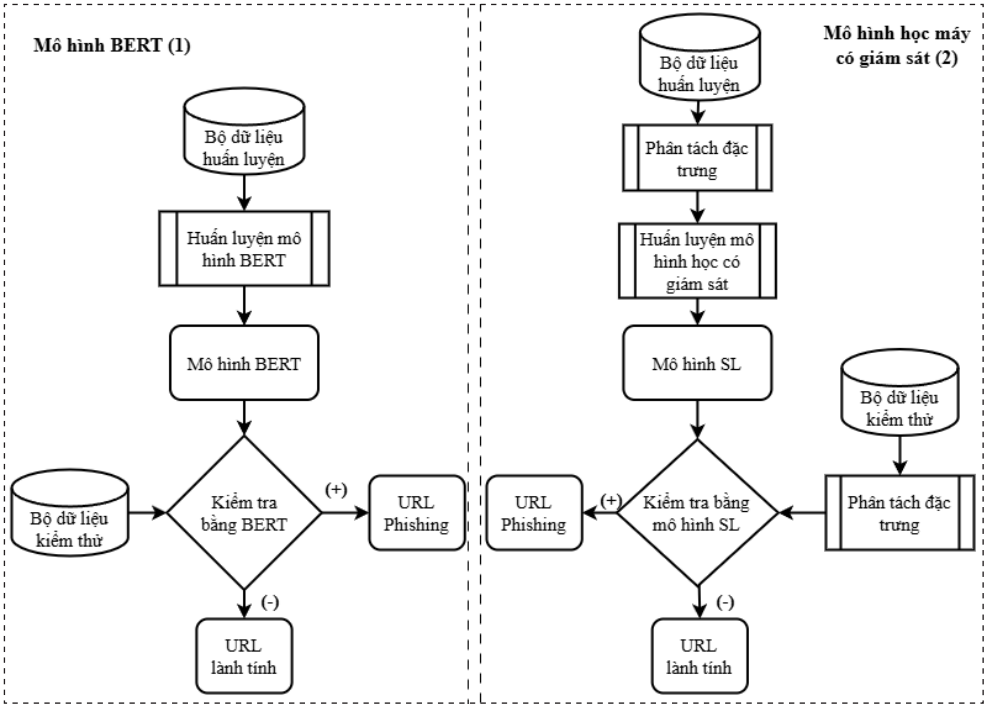


Hình 4. Thống kê một số đặc trưng của domain

Các đặc trưng về domain như sau:

- length: Độ dài của domain (số ký tự).
- ltdFreq: Tần suất domain xuất hiện trong danh sách tên miền lành tính (1 nếu có trong legitDomain1m.txt, 0 nếu không).
- numberCount: Số lượng chữ số trong domain.
- hyphenCount: Số lượng dấu - trong domain.
- numHypRatio: Tỉ lệ (số chữ số + số dấu -) trên độ dài domain.
- even01: Kiểm tra xem tổng ASCII của hai ký tự đầu tiên của domain có chia hết cho 2 không (1 nếu có, 0 nếu không).
- entropy: Entropy ký tự của domain, dựa trên phân phối nguyên âm, phụ âm, chữ số, và ký tự khác.
- wordCount: Số lượng từ có nghĩa trong domain.
- wordEntropy: Entropy của ký tự thuộc các từ có nghĩa và không thuộc từ trong domain.
- freqCommonWord: Tổng giá trị tần suất của các từ phổ biến trong domain.
- wordCombination: Kiểm tra xem có cặp từ kết hợp hợp lệ trong domain hay không (1 nếu có, 0 nếu không).
- maxWordLen: Độ dài của từ dài nhất trong domain.
- minWordLen: Độ dài của từ ngắn nhất trong domain.
- hasNoun: Kiểm tra xem domain có chứa danh từ hay không (1 có, 0 không).
- hasVerb: Kiểm tra xem domain có chứa động từ hay không (1 có, 0 không).
- hasAdjective: Kiểm tra xem domain có chứa tính từ hay không (1 có, 0 không).

3.3. Mô hình phát hiện



Hình 5. Quy trình huấn luyện và kiểm thử URL Phishing đề xuất

Hình 5 mô tả quy trình huấn luyện và kiểm thử đề xuất, hai mô hình phát hiện hoạt động một cách độc lập nhằm so sánh hiệu quả giữa phương pháp học sâu sử dụng BERT (1) và phương pháp học máy truyền thống dựa trên đặc trưng được trích xuất thủ công (2).

Với phương pháp dựa trên mô hình BERT, toàn bộ chuỗi URL từ bộ dữ liệu huấn luyện được đưa trực tiếp vào quá trình huấn luyện mô hình ngôn ngữ sâu, trong đó BERT có khả năng tự động học các đặc trưng ngữ nghĩa và cấu trúc từ dữ liệu đầu vào thông qua các token được phân tách. Sau khi hoàn tất huấn luyện, mô hình dựa trên BERT được sử dụng để kiểm tra các URL trong bộ dữ liệu kiểm thử và đưa ra quyết định phân loại URL là phishing hay lành tính.

Ngược lại, mô hình học máy có giám sát không trực tiếp học từ chuỗi URL, mà yêu cầu phân tách và trích xuất đặc trưng thủ công từ URL như chiều dài, số lượng ký tự đặc biệt, số lượng dấu chấm, v..v. Những đặc trưng được xây dựng thành vectơ đặc trưng sử dụng để huấn luyện thông qua các thuật toán học máy như Random Forest hoặc XGBoost. Trong pha kiểm thử, các URL cũng cần được trích xuất đặc trưng tương tự trước khi đưa vào mô hình để phân loại..

3.4. Phương pháp đánh giá

Nhóm tác giả sử dụng ma trận nhầm lẫn để đánh giá mô hình:

- Tỷ lệ dương tính giả:

$$FPR = \frac{FP}{FP + TN} \quad (1)$$

- Tỷ lệ âm tính giả:

$$FNR = \frac{FN}{FN + TP} \quad (2)$$

- Độ đo:

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (3)$$

• Độ chính xác toàn cục:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

trong đó, TP là dương tính, TN là âm tính, FP là dương tính giả (nhầm lẫn) và FN là âm tính giả (bỏ sót).

• Tỷ lệ kiểm thử:

$$DR = \frac{NCorrect}{NTest} \quad (5)$$

trong đó, NCorrect là số mẫu chẩn đoán chính xác, NTest là tổng số mẫu kiểm thử.

IV. Kết quả và thảo luận

4.1. Tập dữ liệu huấn luyện và kiểm thử

Bộ dữ liệu chúng tôi sử dụng trong bài nghiên cứu này là Malicious URLs dataset (Siddhartha, 2021). Bộ dữ liệu này không chỉ có URL Phishing mà còn có URL Malware và URL Defacement; được tổng hợp từ những tập dữ liệu đã được các nghiên cứu trước đây sử dụng như ISCX-URL-2016, Malware Domain Blacklist, Phishtank, PhishStorm, cùng các repo GitHub chuyên về URL an toàn, đảm bảo tính tin cậy và đa dạng của dữ liệu.

Dữ liệu sử dụng trong bài toán được chia làm 2 phần: dữ liệu huấn luyện và dữ liệu kiểm thử. Tập dữ liệu huấn luyện gồm: 75.000 URL lành tính gán nhãn "0"; 25.000 URL phishing, 25.000 URL malware, 25.000 URL defacement tổng cộng gồm 75.000 URL độc hại gán nhãn "1"; bài toán được đưa về phân loại nhị phân. Tập dữ liệu dùng để kiểm thử gồm 5.000 URL độc hại mỗi loại và không nằm trong tập đã huấn luyện. Trong pha kiểm thử, chúng tôi triển khai tổ chức kiểm thử theo từng họ URL để phân tích và đánh giá.

Bảng 2. Dữ liệu huấn luyện và kiểm thử

Tập dữ liệu	Bình thường	Phishing	Malware	Defacement	Tổng
Huấn luyện	75.000	25.000	25.000	25.000	150.000
Kiểm thử		5.000	5.000	5.000	15.000

4.2. Kết quả và đánh giá

Kết quả tại Bảng 3 và Bảng 4 cho thấy khi bổ sung 16 đặc trưng domain làm tăng đáng kể hiệu suất của mô hình. Cụ thể, ACC tăng 2,64% và 4,01% khi sử dụng Random Forest và XGBoost. Tỷ lệ âm tính giả giảm 3,13% và 4,37% và dương tính giả giảm 2,14% và 3,63% lần lượt khi sử dụng Random Forest và XGBoost.

Bảng 3. Hiệu suất mô hình sử các đặc trưng URL (Vũ & cộng sự, 2022)

Thuật toán sử dụng	ACC	F1-score	FNR	FPR
Random Forest	93,69%	93,57%	8,38%	4,23%
XGBoost	93,70%	93,69%	6,69%	5,90%

Bảng 4. Hiệu suất mô hình sau khi thêm 16 đặc trưng domain

Thuật toán sử dụng	ACC	F1-score	FNR	FPR
Random Forest	96,33%	96,28%	5,24%	2,09%
XGBoost	97,71%	97,71%	2,32%	2,27%

Trong nghiên cứu này, chúng tôi sử dụng 2 phương pháp để lựa chọn một phương pháp mà ở đó hiệu suất là tốt nhất. Phương pháp học máy có giám sát sử dụng: Random Forest và XGBoost; Phương pháp dựa trên kiến trúc transformer sử dụng mô hình BERT.

Bảng 5. Hiệu suất huấn luyện

Thuật toán sử dụng	ACC	F1-score	FNR	FPR
Random Forest	96,33%	96,28%	5,24%	2,09%
XGBoost	97,71%	97,71%	2,32%	2,27%
BERT	99,05%	99,05%	1,40%	0,29%

Kết quả trong Bảng 5 cho thấy mô hình BERT đạt hiệu suất vượt trội so với hai thuật toán học máy có giám sát, bao gồm Random Forest và XGBoost. Cụ thể, mô hình BERT đạt độ chính xác là 99,05%, tỷ lệ âm tính giả và dương tính giả cũng thấp nhất lần lượt là 1,40% và 0,29%. Điều này cho thấy mô hình BERT không chỉ có độ chính xác cao mà còn giảm thiểu rủi ro bỏ sót URL Phishing và hạn chế phát hiện nhầm URL lành tính thành URL Phishing. Vì vậy chúng tôi lựa chọn mô hình BERT cho việc phán đoán URL Phishing.

Bảng 6. So sánh với các nghiên cứu trước đây

Các nghiên cứu	ACC	F1-score
Opara và cộng sự (2024)	98,10%	
Taha và cộng sự (2024)	96,89%	
Ghalechya và cộng sự (2024)	97,00%	88,41%
Remmide và cộng sự (2022)	98,95%	98,00%
Sanchez-Paniagua và cộng sự (2022)	96,50%	96,51%
Đề xuất của chúng tôi	99,05%	99,05%

Bảng 6 so sánh hiệu suất mô hình đề xuất của chúng tôi với các nghiên cứu trước đây, mô hình của chúng tôi đạt kết quả khá khả quan. So với các phương pháp truyền thống như Random Forest hay XGBoost, mô hình BERT không chỉ vượt trội về mặt độ chính xác mà còn thể hiện sự ổn định trong phân loại qua F1-score cao, cho thấy khả năng cân bằng tốt giữa precision và recall.

Bảng 7. Tỉ lệ phát hiện (DR) của các họ với từng mô hình

Thuật toán sử dụng	Phishing	Malware	Defacement	Tất cả
Random Forest	95,20%	98,26%	90,92%	95,87%
XGBoost	96,42%	98,42%	96,58%	97,66%
BERT	99,08%	99,92%	99,80%	99,45%

Bảng 7 thống kê kết quả phát hiện thử nghiệm trên 3 nhóm URL Phishing, cho thấy mô hình BERT đạt tỷ lệ phát hiện cao nhất ở cả ba loại: Phishing đạt 99,08%, Malware 99,92%, Defacement 99,80% và tổng thể đạt 99,45%. So với một số nghiên cứu trước, mô hình BERT thể hiện khả năng phân loại vượt trội và ổn định giữa các nhóm URL độc hại. Điều này khẳng định tính hiệu quả và tính tổng quát cao của mô hình BERT trong nhiệm vụ phát hiện URL Phishing đa dạng.

V. Kết luận

Thông qua nghiên cứu này, nhóm tác giả đã xây dựng và đánh giá một mô hình phát hiện URL Phishing dựa trên hai hướng tiếp cận: học máy có giám sát và dựa trên kiến trúc transformer. Qua thực nghiệm trên bộ dữ liệu lớn và đa dạng, kết quả thu được cho thấy mô hình BERT có hiệu suất cao, đạt 99,05%. Tỷ lệ phát hiện đạt 99,45% trên bộ dữ liệu thử nghiệm. Hướng tiếp cận này giúp loại bỏ quá trình tìm kiếm và trích xuất các đặc trưng của URL Phishing. Mô hình BERT dựa trên kiến trúc transformer giúp mô hình có khả năng phát hiện linh hoạt và hiệu quả các mẫu URL Phishing, kể cả các biến thể chưa từng xuất hiện trong dữ liệu huấn luyện.

Trong tương lai, chúng tôi định hướng mở rộng nghiên cứu theo hai hướng: (1) sử dụng các mô hình học sâu tiên tiến nhằm triển khai trên các thiết bị đầu cuối hoặc nền tảng giới hạn tài nguyên, (2) sử dụng các tập dữ liệu mới, lớn hơn, đa dạng hơn nhằm phát hiện URL Phishing.

Tài liệu tham khảo

[1]. Alabdan, R. (2020). Phishing Attacks Survey: Types, Vectors, and Technical Approaches. *Future Internet, 12*(10), 168. doi:doi: 10.3390/fi12100168

[2]. APWG. (2024). *APWG Reports*. (Phishing Activity Trends Report: 2024 Annual Report) Retrieved 2025, from https://docs.apwg.org/reports/apwg_trends_report_q3_2024.pdf

[3]. Chiew, K. L., Yong, K. S., & Tan, C. L. (2018). A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications, 106*, 1-20. doi:doi: 10.1016/j.eswa.2018.03.050

[4]. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proc. 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. (NAACL-HLT)*, 4171-4186.

[5]. FBI. (2023). *FBI IC3 Annual Report*. (2023 Internet Crime Report) Retrieved 2025, from https://www.ic3.gov/Media/PDF/AnnualReport/2023_IC3Report.pdf

[6]. Ghalechya, H., Israyelyan, E., Arakelyan, A., Hovhannisyan, G., & Davtyan, A. (2024). Phishing URL detection with neural networks: an empirical study. *Scientific Reports, 14*(25134). doi:DOI: 10.1038/s41598-024-74725-6

[7]. Opara, C., Chen, Y., & Wei, B. (2024). Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics. *Expert Systems with Applications, 236*. doi:https://doi.org/10.1016/j.eswa.2023.121183

[8]. Remmide, M. A., Boumahdi, F., Boustia, N., Feknous, C. L., & Della, R. (2022). Detection of Phishing URLs Using Temporal Convolutional Network. *Procedia Computer Science, 212*(https://doi.org/10.1016/j.procs.2022.10.209), 74-82.

[9]. Sahoo, D., Liu, C., & Hoi, S. C. (2017). (Malicious URL Detection using Machine Learning: A Survey) Retrieved 2025, from https://arxiv.org/abs/1701.07179

[10]. Sánchez-Paniagua, M., Fernandez, E. F., Alegre, E., Al-Nabki, W., & Gonzalez-Castro, V. (2022). Phishing URL Detection: A Real-Case Scenario Through Login URLs. *IEEE Access, 10*(DOI: 10.1109/ACCESS.2022.3168681), 42949 - 42960.

[11]. Siddhartha, M. (2021). *Kaggle*. (Malicious URLs dataset) Retrieved 2025, from https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset

[12]. Smith, B. (2024). *Towards Data Science*. (A Complete Guide to BERT with Code) Retrieved 2025, from https://towardsdatascience.com/a-complete-guide-to-bert-with-code-9f87602e4a11/

[13]. Taha, M. A., Jabar, H. D., & Mohammed, W. K. (2024). A Machine Learning Algorithms for Detecting Phishing Websites: A Comparative Study. *Iraqi Journal for Computer Science and Mathematics, 5*(DOI: 10.52866/ijcsm.2024.05.03.015), 275-286.

[14]. Vũ, X. H., Trần, T. D., Đỗ, T. U., Hoàng, V. T., & Ngô, M. P. (2022). Phát hiện Email URL lừa đảo sử dụng học máy có giám sát. *Tạp chí Khoa học - Trường Đại học Mở Hà Nội*, 44-53. Retrieved from https://www.researchgate.net/publication/368541645_PHAT_HIEN_EMAIL_URL_LUA_DAO_SU_DUNG_HOC_MAY_CO_GIAM_SAT_DETECT_EMAIL_URLS_PHISHING_USING_SUPERVISED_MACHINE_LEARNING

[15]. Vũ, H. T. (2016). Phân nhóm các thuật toán Machine Learning. In *Machine Learning.* https://machinelearningcoban.com/2016/12/27/categories/.

PHISHING URL DETECTION BASED ON THE BERT MODEL

Vu Xuan Hanh[4], Do Duy Trinh[4], Ngo Van Son[5], Nguyen Anh Tuan[6]

Abstract: *In the context of the growing and increasingly complex cyber-attacks, particularly cyber-fraud schemes, the development of attack detection models has become an urgent necessity. This paper proposes a Phishing URL detection method based on a transformer architecture and compares it with supervised machine learning-based detection methods using feature extraction. The authors extracted 36 features and categorized them into two main groups: URL features and Domain features. Algorithms such as Random Forest, XGBoost, and the BERT model were trained, tested, and evaluated on a diverse dataset, including Phishing URLs, Malware URLs, and Defacement data. The results show that the BERT model achieved an accuracy of 99.05% and a high detection rate of 99.45%, demonstrating its stability and proving the effectiveness of the transformer-based approach.*

Keywords: *Phishing URL, Phishing URL detection, transformer architecture, BERT, XGBoost, Machine Learning, Random Forest*

---

[4] Hanoi Open University

[5] Postgraduate student, Hanoi Open University

[6] Student, Hanoi Open University