Phân loại luồng dữ liệu dựa trên học chuyển giao đa nguồn trong hệ thống mạng SDN phân tán

Hoàng Nam Thắng, Nguyễn Trần Lê Tuấn, Dương Công Sơn, Tống Văn Vạn, Trần Hải Anh

***Tóm tắt*— Ngày nay, phân loại luồng dữ liệu mạng đóng một vai trò quan trọng trong nhiều lĩnh vực như quản trị mạng, bảo mật, an toàn thông tin. Dữ liệu mạng mã hoá đang dần phổ biến, đơn cử như các dữ liệu của các nhà cung cấp dịch vụ lớn như Google, Facebook. Với các dữ liệu mạng bị mã hoá thì các phương pháp truyền thống như phân loại theo cổng (Port), theo kiểm tra gói sâu (DPI – Deep Packet Inspection) đã không còn hiệu quả, thay vào đó các phương pháp dựa vào học máy sẽ hiệu quả hơn. Gần đây các nghiên cứu về mạng định nghĩa bằng phần mềm SDN (Software Defined Networking) phân tán đã giúp giải quyết vấn đề nhất quán dữ liệu giữa các miền SDN. Từ đó vấn đề phân loại luồng dữ liệu trong mạng SDN phân tán có thể quy về việc phân loại như trên một miền, tuy nhiên trong trường hợp có một miền mạng SDN mới kết nối vào mạng SDN phân tán với số lượng ít các dữ liệu có thể dẫn tới kết quả phân loại không được tốt. Với lý do đó nhóm tác giả đã đề xuất giải thuật MMSTrAdaBoost, một giải thuật chuyển giao tri thức từ các miền mạng đã có nhiều dữ liệu sang miền mạng mới thành lập dựa trên giải thuật MSTradaBoost (TrAdaBoost đa nguồn). Kết quả phân loại dữ liệu mã hóa khi sử dụng giải pháp đề xuất để phân loại 3 loại dịch vụ E-commerce, Interactive data, Video on-demand trên miền mạng mới xuất hiện với chỉ số macro-F1 đều đạt trên 88%.**

***Abstract*— Network traffic classification (TC) is a critical task in network management, security and information security. As network encryption becomes more popular, TC-based machine learning methods have shown great performance, compared to other TC approaches such as port-based or payload inspection. Besides, recent studies on Software-defined networking (SDN) architecture have addressed the data consistency problem in distributed SDN. This means that the TC problem in distributed SDN with multiple domains can now be considered as one domain. Nevertheless, when a new SDN domain is added to the distributed system, the lack of network data on this domain is inevitable. This can make it difficult to train a good TC model for the new domain due to the absence of a training dataset. To address the problem of insufficient training data in a new SDN domain, this paper proposes a algorithm, called MMSTrAdaBoost (modified multiple source TrAdaBoost), a transfer learning method that utilizes knowledge already learned from exsisting SDN domains to improve the performance of the TC model in a new domain. Specifically, our proposal is based on a multisource TradaBoost algorithm that takes advantage of useful data from various source domains. The experimental results show that the TC model in a new domain based on our proposal achieves about 88% macro-F1, when detecting three popular network services: E-commerce, Interactive data, and Video on-demand.**

Từ khóa— phân loại luồng dữ liệu mạng; mạng định nghĩa bằng phần mềm; học chuyển tiếp; TrAdaBoost; TrAdaBoost đa nguồn.

Keywords— traffic classification; SDN; transfer learning; TrAdaBoost; multisource TrAdaBoost.

I. ĐẶT VẤN ĐỀ

Phân loại luồng dữ liệu mạng là một chủ đề quan trọng, được quan tâm trong hệ thống mạng hiện đại ngày nay [1], có thể giúp nhà quản trị mạng theo dõi và kiểm soát lưu lượng mạng, đảm bảo sự ưu tiên cho các dịch vụ quan trọng.

Bài báo được gửi báo cáo trước đó tại Hội thảo quốc gia VNICT 2023, sau đó gửi Tạp chí vào ngày 25/9/2023. Bài báo được nhận xét bởi phản biện thứ nhất vào ngày 26/9/2023 và được chấp nhận đăng vào ngày 02/10/2023. Bài báo được nhận xét bởi phản biện thứ hai vào ngày 03/10/2023 và được chấp nhận đăng vào ngày 05/10/2023.

Ngoài ra, phân loại luồng cũng có thể hỗ trợ trong việc phát hiện và ngăn chặn các hoạt động độc hại như tấn công mạng. Trong số các phương pháp tiếp cận bài toán phân loại luồng dữ liệu mạng, kỹ thuật phân loại dựa theo cổng, phân loại dựa theo nội dung của gói tin và phân loại dựa trên thống kê là các phương pháp phổ biến nhất [2].

Phương pháp phân loại dựa theo cổng sẽ xác định các ứng dụng thông qua các cổng đã được mặc định trước. Tuy nhiên, phương pháp này có mặt hạn chế là độ chính xác không cao khi mà các ứng dụng ngày nay đã sử dụng các cổng động hoặc che giấu giá trị này bởi các kỹ thuật mã hóa hoặc đóng gói. Trong khi đó, phương pháp phân loại dựa theo nội dung của gói tin tập trung vào việc tìm kiếm các mẫu nhận dạng của các ứng dụng mạng trong từng gói tin. Phương pháp này có thể nhận dạng được các ứng dụng sử dụng các số cổng động hoặc bị che giấu. Tuy nhiên, nó có nhược điểm là phức tạp, tốn thời gian và không hiệu quả khi các gói tin bị mã hóa hoặc nén [3]. Đối với phương pháp phân loại theo thống kê, kỹ thuật này xác định các luồng dữ liệu thuộc các ứng dụng khác nhau sẽ có đặc trưng riêng biệt như: độ trễ giữa các gói tin trong một luồng, thời gian sống của luồng dữ liệu. Sau đó, các đặc trưng này sẽ được đưa vào các thuật toán học máy (ML) để huấn luyện và dự đoán nhãn của các luồng dữ liệu. Với dữ liệu được mã hóa, nhiều nghiên cứu đã cho thấy phương pháp học sâu (DL) có kết quả phân loại luồng vượt trội so với các phương pháp truyền thống trên [2-8].

SDN được xuất hiện trong thời gian gần đây, là một kiến trúc mạng sáng tạo, đại diện cho xu hướng phát triển của mạng trong tương lai [9]. Trong nhiều nghiên cứu mạng về SDN, nhiều nhóm nghiên cứu đã quan tâm đến hệ thống mạng SDN phân tán, với mỗi miền mạng được quản lý bởi một hoặc nhiều bộ điều khiển SDN [10-12]. Phân loại luồng dữ liệu mạng trong mạng SDN phân tán có thể quy về việc xem xét như một miền mạng khi sử dụng kết quả nghiên cứu [13] của nhóm tác giả, tuy nhiên trong tình huống các miền mạng mới xuất hiện với thách thức về thiếu hụt dữ liệu có thể dẫn tới kết quả huấn luyện mô hình trên miền mới này sẽ không được tốt.

Để giải quyết mặt hạn chế trên, bài báo này tập trung vào việc đề xuất phương pháp sử dụng tri thức của các mô hình đã học được từ các miền mạng trước đó, để cải tiến mô hình phân loại luồng dữ liệu trên miền mạng mới chưa có đầy đủ dữ liệu. Giải pháp của nhóm tác giả được dựa trên thuật toán MMSTrAdaBoost, là một phương pháp học chuyển giao dựa trên trọng số, nhằm khai thác nhiều nguồn dữ liệu khác nhau để cải thiện hiệu quả của mô hình phân loại trên tập dữ liệu mục tiêu [14]. Ở trong phạm vi của bài báo này, tập dữ liệu mục tiêu là tập dữ liệu của một miền mạng SDN mới xuất hiện, tuy nhiên đang gặp vấn đề thiếu hụt dữ liệu để huấn luyện mô hình phân loại luồng. Giải thuật cải tiến cũng áp dụng một cơ chế chọn lọc để loại bỏ những nguồn dữ liệu không liên quan hoặc gây nhiễu cho việc chuyển giao tri thức từ các miền mạng trước đó sang một miền mạng mới.

Phần còn lại của bài báo được tổ chức như sau. Chương II khảo sát các nghiên cứu áp dụng kỹ thuật DL và học chuyển giao trong phân loại luồng. Chương III đề cập cơ sở lý thuyết về thuật toán TrAdaBoost cho các bài toán học chuyển giao. Chương IV đề cập tới phương pháp đề xuất của nhóm tác giả, Chương V thực nghiệm và phân tích hiệu năng của phương pháp đề xuất. Chương VI tóm gọn kết quả đạt được trong bài báo.

II . CÁC NGHIÊN CỨU LIÊN QUAN

Trong phần này, nhóm tác giả tiến hành khảo sát một số nghiên cứu đã ứng dụng kỹ thuật DL cho bài toán phân loại luồng mạng bị mã hóa. Phần lớn các nghiên cứu đều cho thấy các kỹ thuật DL có kết quả phân loại vượt trội so với các phương pháp truyền thống.

Mahdi [2] đề xuất cơ chế "Deep Packet" để phân loại luồng bị mã hóa dựa trên thuật toán DL. Kiến trúc của Deep Packet được dựa trên sự kết hợp của cả hai mạng là mạng nơ-ron tái

tạo (autoencoder) và mạng nơ-ron tích chập. Deep Packet có hai mục tiêu chính. Thứ nhất, có thể phân loại được lưu lượng dựa theo nhóm dịch vụ của chúng như: truyền File, Chat, gửi Email. Thứ hai, có thể phân loại được lưu lượng dựa theo kiểu ứng dụng của chúng như: Youtube, Netflix, Skype. Điểm nổi bật của phương pháp này là có thể nhận dạng được lưu lượng kể cả khi bị mã hóa VPN hoặc lưu lượng không bị mã hóa. Kết quả thực nghiệm cho thấy Deep Packet đạt 97,3% của marco-F1 đối với phân loại kiểu ứng dụng, và đạt 99,5% của marco-F1 đối với phân loại nhóm dịch vụ.

Wang [5] đề xuất đề xuất cơ chế "Datanet" để phân loại luồng bị mã hóa trên hệ thống mạng của nhà thông minh. Điểm nổi bật của nhóm tác giả là phát triển cơ chế Datanet trên nền tảng SDN, sau đó tích hợp SDN vào thiết bị cảm biến trung gian nhằm quản lí và phân bố tài nguyên mạng hiệu quả, đảm bảo chất lượng dịch vụ cho các thiết bị nhà thông minh. Kiến trúc của "Datanet" được dựa trên mạng nơ-ron tích chập và mạng nơ-ron đa tầng truyền thẳng và được huấn luyện với bộ dữ liệu hơn 200.000 luồng dữ liệu được mã hóa từ 15 ứng dụng khác nhau với đa dạng giao thức như HTTPS, SSH và SSL. Phương pháp đề xuất của nhóm tác giả cũng đạt tỷ lệ xấp xỉ 98% cho các chỉ số đánh giá phân loại như độ chuẩn xác, độ phủ và marco-F1.

Một số nhóm cũng đã sử dụng phương pháp chuyển giao tri thức từ miền nguồn sang miền mục tiêu, để phân loại luồng dữ liệu mạng. Guanglu Sun và các cộng sự [15] đã dùng dữ liệu của đại học Cambridge thu thập với 12 lớp ứng dụng như Mail, Game. Nhóm tác giả đã sử dụng thuật toán TrAdaBoost để huấn luyện mô hình với việc sử dụng mô hình phân lớp yếu là Maximum Entropy và phân loại cho nhiều lớp. Kết quả phân loại đạt độ chính xác 98,7%. Ngoài ra, Zahra Taghiyarrenani và cộng sự [16] đã đề xuất một giải thuật cải tiến của Domain Adaptation để phân loại luồng mạng với dữ liệu từ Brasil. Kết quả phân loại chính xác trên 90%. Bên cạnh sự thành công trong các công bố trên, việc áp dụng giải thuật TrAdaBoost cho bài toàn phân loại luồng dữ liệu mã hóa vào các hệ thống SDN phân tán - đa miền vẫn còn rất hạn chế.

III. Cơ sở lý thuyết

Trong phần này, bài báo sẽ trình bày về ý tưởng cốt lõi của thuật toán TrAdaBoost đơn nguồn và đa nguồn cho bài toán phân loại nhị phân, cũng như cải tiến của TrAdaBoost đơn nguồn cho bài toán phân loại đa lớp.

TrAdaBoost là một kỹ thuật để giải quyết các bài toán học chuyển giao [17]. Mục tiêu của TrAdaBoost là cải thiện khả năng học của một mô hình trên một bài toán đích bằng cách giảm thiểu độ lệch phân phối của dữ liệu giữa bài toán nguồn và bài toán mục tiêu. TrAdaBoost giải quyết vấn đề này bằng cách sử dụng một cơ chế chọn lọc và đánh độ ảnh hưởng cao cho những mẫu dữ liệu có ích từ bài toán nguồn.

A. TrAdaBoost đơn nguồn và đa nguồn

Theo [17], thuật toán TrAdaBoost giả định rằng không gian chứa các véc-tơ đặc trưng của miền nguồn và mục tiêu là như nhau, tức là $\mathcal{X}_S = \mathcal{X}_\mathcal{T}$, nhưng có sự khác nhau về phân phối của dữ liệu, tức là $\mathrm{P}(\mathrm{X_S}) \neq \mathrm{P}(\mathrm{X_T})$. Mấu chốt của thuật toán TradaBoost là nằm ở cơ chế cập nhật trọng số. TrAdaBoost sẽ giảm trọng số, hay giảm độ ảnh hưởng của những mẫu dữ liệu thuộc miền nguồn mà bị phân loại sai. Quá trình này còn gọi là lọc nhiễu (do các mẫu dữ liệu này thường không có ích cho huấn luyện mô hình). Tuy nhiên, đối với những mẫu bị phân loại sai mà thuộc miền mục tiêu, TrAdaBoost sẽ tăng giá trị trọng số của chúng. Đây còn gọi là quá trình tăng cường, tức là khuyến khích mô hình cố gắng học lại những mẫu này trong tương lai.

Theo [14], một trong những vấn đề trong việc chuyển giao tri thức từ một miền nguồn sang một miền mục tiêu là độ tương đồng của dữ liệu giữa hai miền. Nếu hai miền hoàn toàn không có độ tương đồng với nhau, hay phân phối dữ liệu của hai miền quá lệnh, điều này có thể dẫn tới hiện tượng chuyển giao tri thức kém hiệu quả, và giảm khả năng học của mô hình

trên miền mục tiêu. Thuật toán TrAdaBoost đa nguồn giải quyết vấn đề này bằng cách tận dụng tri thức từ nhiều nguồn khác nhau, với hy vọng khai thác được tri thức có ích từ những miền nguồn có độ tương đồng cao với miền mục tiêu. TrAdaBoost đa nguồn vẫn giữ nguyên cơ chế cập nhật trọng số cho các mẫu dữ liệu tương tự như thuật toán TrAdaBoost đơn nguồn ở phần III-A. Tuy nhiên, ở mỗi vòng lặp, TrAdaBoost đa nguồn cần tính toán thêm ba thành phần sau đây. Thứ nhất, cho *N* miền nguồn khác nhau, bộ phân loại thứ i, tại miền nguồn D_{S_i} được huấn luyện riêng biệt trên bộ dữ liệu $X_{S_i} \cup X_T$, với X_T là bộ dữ liệu của miền mục tiêu D_T, và $1 \leq i \leq N$. Thứ hai, tính độ lỗi của từng bộ phân loại thứ i trên X_T, và lựa chọn chọn bộ phân loại có độ lỗi tối thiểu. Cuối cùng, cập nhật trọng số cho các mẫu thuộc miền nguồn mà có bộ phân loại với độ lỗi tối thiểu, và các mẫu thuộc miền mục tiêu là X_T. Khi $N = 1$, thuật toán TrAdaBoost đa nguồn tương tự với TrAdaBoost đơn nguồn.

B. Phân loại đa lớp với TrAdaBoost đơn nguồn

Ở phần III-A đã làm rõ về thuật thuật toán TrAdaBoost đơn nguồn và cải tiến của nó là TrAdaBoost đa nguồn. Tuy nhiên thì hai biến thể trên của TrAdaBoost mới chỉ được thiết kế cho mô hình phân lớp nhị phân. Theo [18], nhóm tác giả đã cải tiến thuật toán TrAdaBoost đơn nguồn cho vấn đề phân loại đa lớp. Ban đầu, ta có cơ chế cập nhật trọng số của TrAdaBoost đơn nguồn như sau.

$$w_i^{t+1} = \begin{cases} w_i^t \cdot \beta^{I(h_t(x_i)\neq y(x_i))}, 1 \leq i \leq m \\ w_i^t \cdot \beta_t^{I(h_t(x_i)\neq y(x_i))}, m+1 \leq i \leq m+n \end{cases} \quad (1)$$

Trong đó, *I(.)* là hàm chỉ nhận hai giá trị là 0 hoặc 1, $i \in [1, m]$ là chỉ các mẫu dữ liệu thuộc miền nguồn và $i \in [m+1, m+n]$ là chỉ các mẫu dữ liệu thuộc miền mục tiêu. w_i^t là véc-tơ trọng số của mẫu thứ i tại vòng lặp thứ t, $h_t(x_i)$ là hàm dự đoán và $y(x_i)$ là nhãn thực tế của mẫu x_i. Ta nhận thấy rằng, với mỗi vòng lặp, các mẫu thuộc miền nguồn sẽ được nhân thêm với hằng số cố định $\beta = 1/\left(1 + \sqrt{2lnm/N}\right)$, trong đó N là số vòng lặp tối đa. Tuy nhiên, đối với các mẫu thuộc miền mục tiêu, chúng sẽ được nhân thêm với hằng số thay đổi $\beta_t = (1 - \varepsilon_t)/\varepsilon_t$, với ε_t là độ lỗi của mô hình khi phân loại trên tập dữ liệu của miền mục tiêu.

Nghiên cứu [15] đề xuất một cơ chế cập nhật trọng số mới được thiết kế cho bài toán phân loại đa lớp.

$$w_i^{t+1} = \begin{cases} w_i^t \cdot K(1-\varepsilon_t) \cdot e^{\alpha \cdot I(h_t(x_i)\neq y(x_i))}, \\ 1 \leq i \leq m \\ w_i^t \cdot e^{\alpha \cdot I(h_t(x_i)\neq y(x_i))}, \\ m+1 \leq i \leq n+m \end{cases} \quad (1)$$

Trong đó, K là tổng số các lớp, α_t là độ tin cậy phân loại của mô hình trên tập dữ liệu miền mục tiêu, với $\alpha_t = log(1-\varepsilon_t)/log(K-1)$. Ngoài ra, $\alpha = log(1/(1+\sqrt{2ln/N})$ phản ảnh độ tin cậy phân loại của mô hình trên tập dữ liệu miền nguồn.

IV. Phương pháp đề xuất

Trong phần này, nhóm tác giả sẽ làm rõ phương pháp đề xuất để giải quyết vấn đề chuyển giao tri thức giữa các miền mạng SDN, nhằm cải thiện bộ phân loại luồng trên miền mạng SDN mới gặp vấn đề thiếu hụt dữ liệu. Phương pháp đề xuất này tận dụng tri thức từ nhiều nguồn khác nhau và có khả năng phân loại luồng cho nhiều dịch vụ mạng. Bài toán có thể thể được mô hình hoá như sau:

Đầu vào:

- Tập $S = \{S_1, \ldots, S_N\}$ là tập chứa N miền mạng SDN nguồn, với $1 \leq k \leq N$.

- Mỗi miền nguồn S_k sẽ có một bộ dữ liệu D_{S_k} cùng với một mô hình phân loại luồng H_k tương ứng.

- Tập $C = \{C_1, \ldots C_{N_c}\}$, với C_v là dịch vụ thứ v tại mỗi miền mạng SDN, $1 \leq v \leq N_c$.

- Giả sử một miền mạng SDN mới (miền mục tiêu) tham gia vào hệ thống phân tán, gọi là miền thứ T, có bộ dữ liệu D_T.

Đầu ra:

- Mô hình phân loại luồng trên miền mạng mục tiêu: $H_T(x_u) = C_v$. Trong đó, $\{x_u \in D | D = D_{S_1} \cup \ldots \cup D_{S_N} \cup D_T\}$, và $C_v \in C$. Ở đây, x_u được coi là ma trận luồng u có kích thước $\mathcal{R}^{\mathcal{N}_\mathcal{P} \times \mathcal{N}_\mathcal{B}}$, và D là tập chứa tất cả các bản ghi trên toàn bộ miền mạng SDN.

Ràng buộc:

- Bộ D_T của miền mạng mục tiêu thứ T gặp vấn đề thiếu hụt dữ liệu.

- Huấn luyện mô hình H_T dựa trên bộ dữ liệu từ N miền mạng nguồn $S_1, \ldots, S_N$.

Ta ký hiệu cặp $\left(w_u^{S_k}, x_u^{S_k}\right)$ lần lượt là trọng số và ma trận đặc trưng của luồng thứ u, tại miền mạng nguồn S_k, với $1 \le u \le \left|D_{S_k}\right|$ và $1 \le k \le N$. Trong đó, $\left|D_{S_k}\right|$ là số lượng bản ghi thuộc miền S_k và ma trận đặc trưng $x_u \in R^{N_p \times N_B}$, với N_p là số lượng bản ghi được sử dụng trong một luồng dữ liệu và N_B là số lượng bytes trong một bản ghi mà nhóm tác giả thu thập. Một giải thuật MMSTrAdaBoost được đề xuất có Mã giả 1. Ở dòng thứ 9 trong Mã giả 1, theo [14] ban đầu các tác giả đề xuất công thức tính độ lỗi được thiết kế cho vấn đề phân lớp nhị phân.

Nhóm tác giả thực hiện thay đổi hàm lỗi này để phù hợp cho vấn đề phân loại với đa lớp. Trong đó ở công thức (5), cặp $\left(w_j^T, x_j^T\right)$ lần lượt là trọng số và ma trận đặc trưng của luồng thứ j, tại miền mục tiêu thứ T, với $1 \le j \le |D_T|$. Ngoài ra, y_j^T là kiểu dịch vụ (nhãn thực tế) của luồng thứ j, và $H_k(.)$ là kiểu dịch vụ được dự đoán bởi mô hình. Hàm $y_j^T \cdot \log(.)$ thực tế là công thức cross-entropy, được dùng để đánh giá độ sai lệch giữa nhãn thực tế và kết quả dự đoán của mô hình phân lớp [19]. Ở dòng thứ 13 trong 1, α_t là độ tin cậy của H_*^t khi phân loại trên bộ dữ liệu miền mục tiêu D_T tại vòng lặp thứ t và m là tổng số mẫu trên toàn bộ miền nguồn. Ở công thức (3), (4) sẽ được hiệu chỉnh lại để phù hợp cho việc phân loại đa lớp dịch vụ với hàm mũ là công thức cross-entropy. Công thức (3) cập nhật trọng số cho các bản ghi thuộc N miền nguồn, với $|C|$ là số lượng dịch vụ cần phân loại. Công thức (4) cập nhật trọng số cho các mẫu thuộc miền mục tiêu.

$$w_u^{S_k} = w_u^{S_k} . |C| . (1 - \epsilon^*) . e^{\alpha . \left(y_u^{S_k} . log\left(H_t^*\left(\mathbf{x}_u^{S_k}\right)\right)\right)}, \quad 1 \le u \le |D_{S_k}|, 1 \le k \le N \tag{3}$$

$$w_v^T = w_v^T . e^{\alpha_t . (y_v^T . \log\,(H_t^*(\mathbf{x}_v^T)))}, 1 \le v \le |D_T| \tag{4}$$

Thuật toán 1. Giải thuật MMSTrAdaBoost

Đầu vào: Tập các dữ liệu nguồn $D = D_{S_1} \cup \ldots \cup D_{S_N}$, bộ dữ liệu D_T của miền mục tiêu.

- M: Số lượng vòng lặp.

Đầu ra: Mô hình phân loại lưu lượng trên miền mạng mục tiêu: H_T.

- Khởi tạo $(\mathbf{w}^{S_1}, \ldots \mathbf{w}^{S_N}, \mathbf{w}^T)$, trong đó $\mathbf{w}^{S_k} = (w_1^{S_k}, \ldots, w_{|D_{S_k}|}^{S_k})$ là véc-tơ trọng số của miền mạng nguồn thứ k, và $\mathbf{w}^T = (w_1^T, \ldots, w_{|D_T|}^T)$ là véc-tơ trọng số của miền mạng mục tiêu.

- Khởi tạo $\alpha = \log\,(1/(1 + \sqrt{2lnm/M}))$.

For $t = 1$ to M **do**

For $k = 1$ to N **do**

- Huấn luyện mô hình phân loại H_k, dựa trên bộ dữ liệu $D_{S_k} \cup D_T$.

- Tính toán độ lỗi của từng mô hình H_k trên bộ dữ liệu của miền mục tiêu là D_T:

$$\varepsilon_k^t = -\sum_j \frac{w_j^T\left(y_j^T \cdot \log\left(H_k\left(x_j^T\right)\right)\right)}{\sum_j w_j^T} \tag{5}$$

end for

- Chọn cặp (H_k^t, ϵ_k^t) sao cho có ϵ_k^t là nhỏ nhất, và gán là (H_*^t, ϵ_*^t).

- Tính $\alpha_t = \frac{\log(1-\epsilon^*)}{\epsilon^*} + \log(|C|-1)$

- Tiến hành cập nhật các véc-tơ trọng số theo công thức (3), (4).

end for

$$H_T(x) = \underset{C_v \in C}{argmax} \sum_{t=1}^{M} \alpha_t \cdot I(H_t^*(x) = C_v \in V) \quad (6)$$

Công thức (6) kết hợp kết quả từ các bộ phân loại H_t^* và độ tin cậy của nó được lựa chọn tại mỗi vòng lặp, để tạo thành một bộ phân loại luồng mạng H_T tối ưu tại miền mục tiêu. Trong đó, *I (.)* là hàm chỉ, nhận giá trị 1 nếu mô hình dự đoán vào kiểu dịch vụ $C_v \in C$ và 0 nếu ngược lại. Cuối cùng, với mỗi luồng dữ liệu, nhm tác giả sử dụng ma trận dữ liệu đặc trưng và phân loại dịch vụ cho chúng theo công thức (6).

V. ĐÁNH GIÁ THỰC NGHIỆM

Trong phần này, nhóm tác giả sẽ tiến hành đánh giá các giải thuật MMSTrAdaBoost với các giải thuật phân loại lưu lượng mạng dựa trên các DL khác nhau.

A. Mô tả bộ dữ liệu

Bộ dữ liệu ban đầu bao gồm các giá trị byte được mã hóa trong gói tin, được bắt thông qua công cụ Selenium WebDriver trên trình duyệt Google Chrome. Dữ liệu này được lưu dưới dạng pcap (packet capture) là một định dạng tập tin dùng để lưu trữ dữ liệu gói tin được bắt trong các mạng máy tính. Các gói tin này được bắt và lưu trữ bởi các phần mềm bắt gói tin như Wireshark. Do bộ dữ liệu này được bắt ở tầng liên kết dữ liệu, nó sẽ bao gồm cả header của tầng liên kết dữ liệu. Header chứa một số thông tin liên quan đến lớp vật lý và đóng vai trò quan trọng trong việc chuyển tiếp các khung trong mạng. Tuy nhiên, thông tin này không quan trọng đối với việc phân loại luồng mạng [5]. Do đó, trong quá trình xử lý, header của tầng liên kết dữ liệu được loại bỏ. Sau đó, gói dữ liệu trong bộ dữ liệu sẽ được chuyển đổi từ bit sang byte để giảm kích thước đầu vào. Để tăng tốc độ huấn luyện, tất cả các byte của gói tin được chuẩn hóa bằng cách chia cho 255 (giá trị tối đa cho một byte). Sau khi lọc thành công, nhóm tác giả đã thu thập được bộ dữ liệu như Bảng 1.

BẢNG 1. THỐNG KÊ BỘ DỮ LIỆU CÁC DỊCH VỤ SỬ DỤNG

Miền SDN	Tên ứng dụng	Số lượng luồng	Số lượng bản ghi	Tổng
Miền 1	Thegioididong	E-commerce	509	8.915
	Tiki	E-commerce	1.205	
	YouTube	Video on-demand	4.645	
	GG Chat	Interactive data	2.556	
Miền 2	Amazon	E-commerce	834	6.247
	Ebay	E-commerce	1.836	
	Facebook	Video on-demand	1.379	
	GG VoIP	Interactive data	2.198	
Miền 3	Alibaba	E-commerce	1.401	7.839
	Shopee	E-commerce	2.581	
	Tiktok	Video on-demand	288	
	File Transfer	Interactive data	3.569	

B. Cấu hình thực nghiệm

1. Phân bố dữ liệu trên các miền mạng

Tập dữ liệu sau khi được thu thập được gom nhóm thành 3 lớp dịch vụ: Interactive data, E-commerce và Video on-demand trong Bảng 1.

Hệ thống mạng SDN phân tán được xem xét có 3 miền trong đó 2 miền mục nguồn D_{S_1} và D_{S_2} với tương ứng 8.915 và 6.247 bản

ghi. Bài báo quy ước bản ghi đại diện cho một luồng dữ liệu, tương đương với một luồng các gói tin liên tiếp. Miền mục tiêu D_T được cấu hình giảm dần kích thước của tập dữ liệu huấn luyện (với một tỷ lệ là XT%) trong tổng số 7.839 bản ghi trong miền đó, ngoài ra 20% dữ liệu (391 bản ghi) được trích ra từ miền D_T để làm tập kiểm thử mô hình. Việc giảm kích thước dữ liệu của miền mục tiêu trên các trường hợp thực nghiệm để đánh giá xem thuật toán đề xuất MMSTrAdaBoost có thể chuyển giao tri thức hiệu quả tới miền mục tiêu có dữ liệu ít ở mức độ nào.

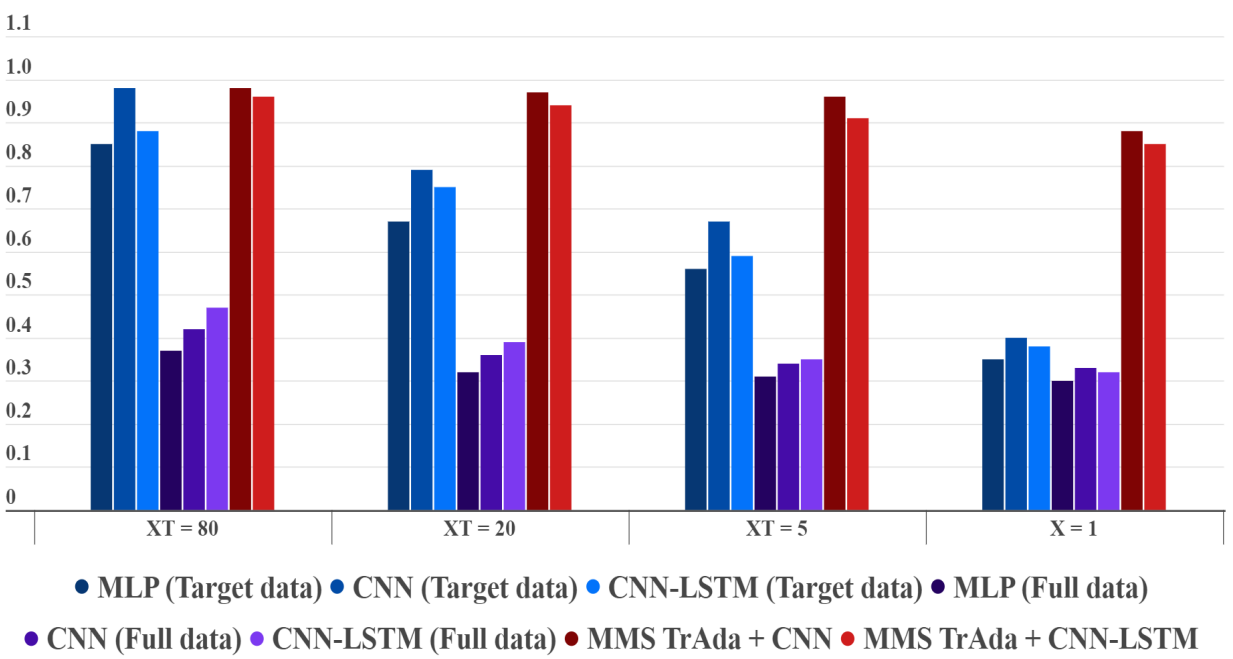
2. Cấu hình bộ phân loại

Các mô hình phân loại luồng được huấn luyện trên máy tính với CPU Intel® Core™ i7 6700HQ Processor, 16GB RAM, với 2560 nhân CUDA.

Theo Mã giả 1, mỗi vòng lặp ta cần chọn được một mô hình phân loại tối ưu cho bộ dữ liệu nguồn và đích. Trong phần này, mô hình mạng nơ-rơn đa tầng (MLP), mạng tích chập (CNN) và mô hình lai giữa mạng tích chập và mạng hồi quy (CNN-LSTM) được sử dụng làm bộ phân loại luồng mạng mã hóa. Cấu hình các siêu tham số của mạng MLP gồm 2 lớp ẩn với số nơ-ron lần lượt là 64, 32; mỗi lớp này đều sử dụng hàm phi tuyến ReLU. Lớp cuối cùng bao gồm 3 nơ-ron đại diện cho 3 kiểu loại dịch vụ, và sử dụng hàm kích hoạt Softmax.

Cấu hình các siêu tham số của mạng CNN như sau: Phần đầu của mạng CNN bao gồm 3 tầng tích chập dùng để trích xuất đặc trưng. Tầng tích chập thứ nhất sử dụng 16 bộ lọc với kích cỡ 32×32, tầng thứ hai sử dụng 32 bộ lọc kích cỡ 16 × 16 và tầng cuối sử dụng 64 bộ lọc với kích cỡ 8 × 8. Sau mỗi tầng tích chập sẽ được sử dụng thêm tầng giảm số chiều có kích cỡ 2×2. Mạng CNN được thiết kế sao cho nếu kích thước của đầu ra của mỗi tầng hiện tại bị giảm một nửa, số bộ lọc của tầng tiếp theo sẽ gấp đôi. Phần cuối của mạng CNN sẽ sử dụng thêm một mạng nơ-ron truyền thẳng đơn giản để làm nhiệm vụ dự đoán kiểu đầu ra. Mạng nơ-ron truyền thẳng bao gồm 1 lớp ẩn với 256 nơ-ron và 1 lớp đầu ra với 3 nơ-ron.

Cấu hình các siêu tham số của mạng CNN-LSTM như sau: Mạng này sử dụng 2 tầng tích chập với kích cỡ lần lượt là 32×32×16 và 16×16×32. Sau mỗi tầng tích chập cũng sẽ được



Hình 1. Đánh giá chỉ số F1-score của các thuật toán, với từng miền dữ liệu khác nhau

sử dụng tầng giảm số chiều, với kích cỡ 2 × 2. Đầu ra của tầng tích chập cuối cùng sẽ được đưa vào mạng LSTM để học mối quan hệ giữa các chuỗi bản ghi liên tiếp của 1 luồng dữ liệu. Mạng LSTM sẽ được cấu hình với 16 ô đơn vị ẩn (được gọi là LSTM unit). Tầng cuối cùng chính là mạng nơ-ron truyền thẳng với lần lượt 32 nơ-ron và 3 nơ-ron ở đầu ra.

Dữ liệu được chia thành 3 tập con: 70% tập huấn luyện (training set) để huấn luyện mô hình, 15% tập kiểm soát (validation set) để giám sát sự quá khớp (overfitting) của mô hình và 15% tập đánh giá (test set) dùng để đánh giá kết quả mô hình sau khi huấn luyện.

C. Kết quả và phân tích thực nghiệm

Kết quả đánh giá trong nghên cứu được đánh giá từ hai quan điểm khác nhau. Đầu tiên, xem xét sự đánh đổi giữa độ phức tạp và độ chính xác của mô hình khi thay đổi kích thước của ma trận đặc trưng PBM. Thứ hai, đánh giá chất lượng của các bộ phân loại khi dữ liệu huấn luyện giảm dần.

Hình 1 so sánh giải thuật MMSTrAdaBoost với các giải thuật DL với kích thước dữ liệu khác nhau dựa trên chỉ số marco-F1. Tại XT = 80%, các giải thuật đề xuất và mô hình DL như CNN, MLP và CNNLSTM với dữ liệu trên miền DT đều cho chất lượng khá tốt lần lượt là 0.98, 0.85 và 0.88. Trong khi đó, việc sử dụng thêm các mẫu dữ liệu trong các miền nguồn làm giảm giá trị marco-F1 của các mô hình này xuống thấp (dưới 0.50). Khi XT = 20%, lượng dữ liệu huấn luyện bị giảm mạnh, các giải thuật DL chứng kiến sự giảm mạnh về giá trị của marco-F1, đặc biệt CNN giảm từ 0.98 xuống 0.79 tại XT = 20%. Các giải thuật MMSTrAdaBoost sử dụng các tri thức của các miền nguồn, do dó giải thuật đề xuất bị ảnh hưởng không đáng kể khi lượng dữ liệu huấn luyện giảm. Tại XT = 5% và 1%, các mô hình DL không còn hiệu quả, với giá trị của marco-F1 < 0.67 và 0.5. Tuy nhiên, giải thuật đề xuất vẫn chứng kiến chỉ số marco-F1 lớn hơn 0,85, đặc biệt MMSTrAdaBoost + CNN có giá trị marco-F1 là 0.88 khi dữ liệu huấn luyện trong miền mục tiêu chỉ khoảng 71 mẫu.

VI. KẾT LUẬN

Bài báo này đã đề xuất phương pháp phân loại luồng dữ liệu mạng tại miền mạng mới hình thành trong mạng SDN phân tán. Phương pháp đề xuất được cải tiến từ giải thuật MMSTrAdaBoost. Thực nghiệm trên bộ dữ liệu mạng mã hóa được thu thập từ 12 ứng dụng mạng và được chia thành 3 nhóm E-commerce, Interactive data và Video on-demand. Kết quả đạt được trong tình huống ít dữ liệu của miền mục tiêu nhất cũng đã đạt hơn 88% trong khi với giải thuật khác chỉ là dưới 67%. Tuy nhiên điểm yếu giải thuật đề xuất của nhóm tác giả là điều kiện hội tụ và thời gian xử lý việc lựa chọn tri thức trong miền phù hợp. Trong tương lai, nhóm tác giả sẽ sử dụng mô hình học tăng cường với việc xác định phần thưởng của việc chọn tri thức trong từng miền tại thời điểm trước để đưa ra quyết định chọn tri thức trong các miền có sẵn lần học tiếp theo. Điều này giúp cải thiện quá trình lựa chọn tri thức và tối ưu hóa hiệu suất của phương pháp.

LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi Trường Đại học Xây dựng Hà Nội (HUCE) trong đề tài mã số 46-2023/KHXD.

TÀI LIỆU THAM KHẢO

[1]. J. Yan and J. Yuan, "A survey of traffic classification in software defined networks," in 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN). IEEE, 2018, pp. 200–206.

[2]. M. Lotfollahi, M. Jafari Siavoshani, R. Shirali Hossein Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," Soft Computing, vol. 24, no. 3, pp. 1999–2012, 2020.

[3]. K.-C. Chiu, C.-C. Liu, and L.-D. Chou, "Capc: Packet-based network service classifier with convolutional autoencoder," IEEE Access, vol. 8, pp. 218 081–218 094, 2020.

[4]. V. Tong, H. A. Tran, S. Souihi, and A. Mellouk, "A novel quic traffic classifier based on convolutional neural networks," in 2018 IEEE Global Communications Conference (GLOBECOM). IEEE, 2018, pp. 1–6.

[5]. P. Wang, F. Ye, X. Chen, and Y. Qian, "Datanet: Deep learning based encrypted network traffic classification in sdn home gateway," IEEE Access, vol. 6, pp. 55 380–55 391, 2018.

[6]. Tuan, T. A., Cuong, N. N., Anh, N. V., & Long, H. V. . (2023). Proposing the application of a deep learning model to detect the malicious IP address of botnet in the computer network. Journal of Science and Technology on Information Security, 3(17), 43-52. https://doi.org/10.54654/isj.v3i17.894.

[7]. Quy, T. N., Tung, N. T., Trung, D. Q., & Viet, D. H. (2022). Convolutional neural network based sidechannel attacks. Journal of Science and Technology on Information Security, 1(15), 26-37. https://doi.org/10.54654/isj.v1i15.834

[8]. Dung, N. T., Quân, N. V., & Hùng, N. V. (2023). Application of deep learning model in network reconnaissance attack detection. Journal of Science and Technology on Information Security, 2(16), 60-72. https://doi.org/10.54654/isj.v1i16.922

[9]. D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," Proceedings of the IEEE, vol. 103, no. 1, pp. 14–76, 2014.

[10]. N.-T. Hoang, V. Tong, H. A. Tran, C. S. Duong, and T. L. T. Nguyen, "Lstm-based server and route selection in distributed and heterogeneous sdn network," Journal of Computer Science and Cybernetics, vol. 39, no. 1, pp. 79– 99, 2023.

[11]. T. Moufakir, M. F. Zhani, A. Gherbi, and O. Bouachir, "Collaborative multi-domain routing in sdn environments," Journal of Network and Systems Management, vol. 30, pp. 1–23, 2022.

[12]. F. Bannour, S. Souihi, and A. Mellouk, "Distributed sdn control: Survey, taxonomy, and challenges," IEEE Communications Surveys & Tutorials, vol. 20, no. 1, pp. 333–354, 2017.

[13]. N.-T. Hoang, H.-N. Nguyen, H.-A. Tran, and S. Souihi, "A novel adaptive east–west interface for a heterogeneous and distributed sdn network," Electronics, vol. 11, no. 7, p. 975, 2022.

[14]. Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE, 2010, pp. 1855–1862.

[15]. "Network traffic classification based on transfer learning," Computers & Electrical Engineering, vol. 69, pp. 920–927, 2018.

[16]. Z. Taghiyarrenani and H. Farsi, "Domain adaptation with maximum margin criterion with application to network traffic classification," pp. 159–169, 2022.

[17]. W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in Proceedings of the 24th international conference on Machine learning, 2007, pp. 193–200.

[18]. H. He, K. Khoshelham, and C. Fraser, "A multiclass tradaboost transfer learning algorithm for the classification of mobile lidar data," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 166, pp. 118–127, 2020.

[19]. M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," arXiv preprint arXiv:2008.05756, 2020.

SƠ LƯỢC VỀ TÁC GIẢ

Hoàng Nam Thắng

Đơn vị công tác: Trường Đại học Xây Dựng Hà Nội.

Email: thanghn@huce.edu.vn

Quá trình đào tạo: Tốt nghiệp Công nghệ thông tin tại Đại học Bách khoa Hà Nội năm 2008; Thạc sĩ Khoa học Máy tính tại Học viện Kỹ thuật quân sự năm 2021. Hiện đang là Nghiên cứu sinh tại Đại học Bách khoa Hà Nội.

Hướng nghiên cứu hiện nay: Mạng định nghĩa bằng phần mềm; phân loại lưu lượng mạng; định tuyến hướng dịch vụ.

Nguyễn Trần Lê Tuấn

Đơn vị công tác: Trường Đại học Xây Dựng Hà Nội.

Email: tuan1553564@huce.edu.vn

Quá trình đào tạo: Tốt nghiệp kỹ sư ngành Khoa học máy tính tại Đại học Xây dựng Hà Nội năm 2023. Hiện đang theo học Thạc sĩ ngành Hệ thống thông minh và ứng dụng, trường Paris-Est Marne-la-Vallee, Đại học Gustave Eiffel, Pháp.

Hướng nghiên cứu hiện nay: Mạng định nghĩa bằng phần mềm; phân loại lưu lượng mạng; định tuyến hướng dịch vụ.

Dương Công Sơn

Đơn vị công tác: Đại học Xây Dựng Hà Nội

Email: son167464@huce.edu.vn

Quá trình đào tạo: Tốt nghiệp kỹ sư ngành Khoa học máy tính tại Đại học Xây Dựng Hà Nội năm 2023.

Hướng nghiên cứu hiện nay: Mạng định nghĩa bằng phần mềm; phân loại lưu lượng mạng; định tuyến hướng dịch vụ.

Tống Văn Vạn

Đơn vị công tác: Đại học Bách khoa Hà Nội

Email: vantv@soict.hust.edu.vn

Quá trình đào tạo: Tốt nghiệp hệ kỹ sư tài năng Công nghệ thông tin tại Đại học Bách khoa Hà Nội năm 2017; Tiến sĩ Công nghệ thông tin tại Đại học Paris-Est Creteil năm 2021.

Hướng nghiên cứu hiện nay: Mạng máy tính; Blockchain; an toàn không gian số.

Trần Hải Anh

Đơn vị công tác: Đại học Bách khoa Hà Nội

Email: anhth@soict.hust.edu.vn

Quá trình đào tạo: Tốt nghiệp kỹ sư Công nghệ thông tin tại Đại học Bách khoa Hà Nội năm 2008; Thạc sĩ Công nghệ thông tin tại trường Paris-Sud Orsay, Pháp, năm 2009; Tiến sĩ Công nghệ thông tin tại Đại học Paris-Est Creteil năm 2013.

Hướng nghiên cứu hiện nay: Mạng máy tính; các hệ thống phân tán; tin sinh học; an toàn không gian số.