Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Đà Nẵng, ngày 17-18/08/2017
DOI: 10.15625/vap.2017.00097

RÚT GỌN THUỘC TÍNH SỬ DỤNG ĐỘ LỢI THÔNG TIN ĐỂ TĂNG CƯỜNG HIỆU NĂNG CỦA CÁC HỆ THỐNG PHÁT HIỆN XÂM NHẬP MẠNG

Hoàng Ngọc Thanh[1, 3], Trần Văn Lăng[2, *]

[1]Trường Đại học Lạc Hồng

[2]Viện Cơ học và Tin học ứng dụng, VAST

[3]Khoa Công nghệ thông tin, Trường Đại học Bà Rịa - Vũng Tàu

thanhhn@bvu.edu.vn, langtv@vast.vn

TÓM TẮT: *Chức năng chính của hệ thống phát hiện xâm nhập mạng (Intrusion Detection System: IDS) là để bảo vệ hệ thống, phân tích và dự báo hành vi truy cập mạng của người sử dụng. Những hành vi này được xem xét là bình thường hoặc một cuộc tấn công. Các phương pháp máy học được sử dụng trong các IDS nhờ khả năng học hỏi từ các mẫu dữ liệu trong quá khứ để nhận ra các mẫu tấn công mới. Các phương pháp này tuy hiệu quả nhưng lại có chi phí tính toán tương đối cao. Trong khi đó, khối lượng và tốc độ của dữ liệu mạng phát triển ngày càng nhanh, các vấn đề chi phí máy tính cần phải được giải quyết. Bài viết này đề cập đến việc sử dụng độ lợi thông tin để rút gọn các thuộc tính của tập dữ liệu cần phân tích. Nhờ đó, giúp xây dựng các IDS với chi phí thấp hơn nhưng hiệu năng cao hơn. Kết quả thử nghiệm trên tập dữ liệu NSL-KDD99 sử dụng đánh giá chéo 5-fold đã minh chứng: với tập các thuộc tính tối ưu phù hợp với kiểu phân lớp cũng như phương pháp máy học, độ chính xác phân lớp của các IDS đã được cải thiện với thời gian tính toán ít hơn.*

Từ khóa: *Máy học, Độ lợi thông tin, An ninh mạng, Rút gọn thuộc tính.*

I. GIỚI THIỆU

Do những tiến bộ công nghệ gần đây, các dịch vụ dựa trên mạng ngày càng đóng vai trò quan trọng trong xã hội hiện đại. Kẻ xâm nhập không ngừng tìm kiếm các lỗ hổng của hệ thống máy tính để truy cập trái phép vào nhân của hệ thống. Tuy nhiên, các IDS hiện tại vẫn chưa đủ linh hoạt, khả năng mở rộng không cao, cũng như không đủ mạnh để đối phó với các cuộc tấn công như vậy.

Trước đây, các phương pháp dựa trên luật đã chiếm ưu thế. Những phương pháp này tìm ra sự xâm nhập bằng cách so sánh các đặc tính của dữ liệu cần phân tích với các dấu hiệu tấn công đã biết. Khi lưu lượng mạng phát triển nhanh chóng, việc cập nhật các dấu hiệu tấn công ngày càng trở nên khó khăn, tẻ nhạt và tốn nhiều thời gian. Kể từ đó, các phương pháp máy học đã được giới thiệu để giải quyết vấn đề phát hiện xâm nhập. Máy học đề cập đến các thuật toán máy tính có khả năng học hỏi từ các mẫu dữ liệu trong quá khứ để nhận ra các mẫu tấn công mới. Dựa trên máy học, các IDS đã hoạt động tốt hơn trong nhiều báo cáo cũng như thực tế triển khai. Tuy nhiên, tài sản "không có mô hình" của các phương pháp như vậy gây ra chi phí tính toán tương đối cao. Hơn nữa, khối lượng và tốc độ của dữ liệu mạng phát triển ngày càng nhanh, các vấn đề chi phí máy tính cần phải được giải quyết [1].

Một trong những giải pháp quan trọng nhằm giảm chi phí tính toán là rút gọn số thuộc tính của dữ liệu cần phân tích. Có nhiều tiếp cận khác nhau về vấn đề này đã được các học giả trình bày [2, 3, 4]. Tuy nhiên, các thuộc tính được lựa chọn tối ưu không chỉ phụ thuộc vào kiểu phân lớp mà còn phụ thuộc vào phương pháp máy học, đến nay chưa có một nghiên cứu nào đánh giá đầy đủ các thuộc tính nào là phù hợp nhất ứng với từng kiểu phân lớp, cũng như phương pháp máy học được sử dụng trong các IDS.

Nội dung bài báo đề xuất sử dụng độ lợi thông tin (Information Gain) để xếp hạng độ quan trọng của các thuộc tính trong tập dữ liệu cần phân tích. Sau đó, sử dụng hai thuật toán Backward Elimination Ranking (BER) và Forward Selection Ranking (FSR) để loại bỏ các thuộc tính không cần thiết. Từ đó, tìm ra các tập thuộc tính tối ưu ứng với từng kiểu phân lớp cũng như phương pháp máy học.

Việc rút gọn số thuộc tính của dữ liệu giúp cải thiện hiệu năng của các IDS dựa trên máy học, cụ thể là giảm thời gian huấn luyện và kiểm tra, đồng thời tăng độ chính xác phân lớp.

II. TẬP DỮ LIỆU

Trước khi các bộ phân lớp được đưa vào sử dụng để phát hiện xâm nhập mạng, các bộ phân lớp phải trải qua quá trình huấn luyện và kiểm tra, việc huấn luyện và kiểm tra được thực hiện trên tập dữ liệu đã được gán nhãn trước. Theo thống kê [5], tập dữ liệu được sử dụng phổ biến nhất trong các thí nghiệm cho đến nay là KDD99, được tạo ra bằng cách xử lý phần dữ liệu TCPDUMP lấy được trong 7 tuần từ hệ thống phát hiện xâm nhập DARPA 1998. KDD99 gồm các tập dữ liệu huấn luyện và kiểm tra. Tập dữ liệu huấn luyện có 4.898.431 bản ghi, mỗi bản ghi có 41 thuộc tính (loại giao thức, dịch vụ và cờ) và được dán nhãn là bình thường hoặc một cuộc tấn công một cách chính xác với một kiểu tấn công cụ thể [6]. Số thứ tự và tên các thuộc tính được mô tả chi tiết ở Bảng 1.

Bảng 1. Thông tin chi tiết 41 thuộc tính của tập dữ liệu huấn luyện và kiểm tra trong KDD99.

1	duration	11	num_failed_logins	21	is_host_login	31	srv_diff_host_rate
2	protocol_type	12	logged_in	22	is_guest_login	32	dst_host_count
3	service	13	num_compromised	23	count	33	dst_host_srv_count
4	flag	14	root_shell	24	srv_count	34	dst_host_same_srv_rate
5	src_bytes	15	su_attempted	25	serror_rate	35	dst_host_diff_srv_rate
6	dst_bytes	16	num_root	26	srv_serror_rate	36	dst_host_same_src_port_rate
7	land	17	num_file_creations	27	rerror_rate	37	dst_host_srv_diff_host_rate
8	wrong_fragment	18	num_shells	28	srv_rerror_rate	38	dst_host_serror_rate
9	urgent	19	num_access_files	29	same_srv_rate	39	dst_host_srv_serror_rate
10	hot	20	num_outbound_cmds	30	diff_srv_rate	40	dst_host_rerror_rate
						41	dst_host_srv_rerror_rate

Tập dữ liệu huấn luyện chứa 22 kiểu tấn công và thêm 17 kiểu trong tập dữ liệu kiểm tra, được phân thành 4 nhóm:

(1) Denial of Service (DoS), gồm các kiểu tấn công như: neptune, smurf, pod, teardrop,… Ở đó, kẻ tấn công làm cho các tài nguyên tính toán hoặc bộ nhớ quá tải để xử lý các yêu cầu hợp lệ, hoặc từ chối người dùng hợp lệ truy cập máy.

(2) Remote to Local (R2L), gồm các kiểu tấn công như: guess-passwd, ftp-write, imap, phf,… Ở đó, kẻ tấn công tuy không có tài khoản nhưng có khả năng gửi các gói tin đến một máy qua mạng, sẽ khai thác một số lỗ hổng để đạt được quyền truy cập cục bộ như là người sử dụng của máy đó.

(3) User to Root (U2R), gồm các kiểu tấn công như: buffer-overflow, load-module, perl, rootkit,… Ở đó, kẻ tấn công bắt đầu với một quyền truy cập bình thường và sau đó khai thác một số lỗ hổng để đạt được quyền truy cập root trên hệ thống.

(4) Probe, gồm các kiểu tấn công như: port-sweep, ip-sweep, nmap,… Ở đó, kẻ tấn công nỗ lực thu thập thông tin về mạng máy tính nhằm phá vỡ khả năng kiểm soát an ninh của nó.

Năm 2009, Tavallaee và các đồng nghiệp [6] đã tiến hành phân tích thống kê bộ dữ liệu KDD99. Các tác giả tìm thấy một số lượng lớn các bản ghi dư thừa, 78% trong tập dữ liệu huấn luyện và 75% trong tập dữ liệu kiểm tra. Số lượng bản ghi trùng lặp này có thể ngăn chặn các thuật toán máy học với các bản ghi không xuất hiện thường xuyên như các cuộc tấn công U2R. Các tác giả cũng lưu ý rằng các bản ghi trùng lặp trong tập dữ liệu KDD99 cũng sẽ làm cho kết quả đánh giá bị sai lệch, bởi các thuật toán sẽ phát hiện tốt hơn với các bản ghi xuất hiện thường xuyên. Tavallaee và các đồng nghiệp [6] đã tạo bộ dữ liệu NSL-KDD từ tập dữ liệu KDD99 để giải quyết các vấn đề đã đề cập ở trên, bằng cách loại bỏ các bản ghi dư thừa. Tập dữ liệu huấn luyện của NSL-KDD gồm 125.973 bản ghi và tập dữ liệu kiểm tra gồm 22.544 bản ghi, ít hơn nhiều so với tập dữ liệu KDD99. Các tác giả cho rằng kích thước của tập dữ liệu NSL-KDD là hợp lý, có thể được sử dụng như tập dữ liệu hoàn chỉnh mà không cần phải lấy mẫu ngẫu nhiên. Điều này cho phép xem xét một cách nhất quán và có thể so sánh các công trình nghiên cứu khác nhau.

Thông tin chi tiết về mỗi kiểu tấn công trong tập dữ liệu NSL-KDD được mô tả trong Bảng 2.

Bảng 2. Thông tin chi tiết tập dữ liệu huấn luyện trong NSL-KDD.

Phân lớp tấn công	Tên tấn công	Số bản ghi	Tỷ lệ %
Normal		67.343	53,45%
Probe	ipsweep, mscan, nmap, portsweep, saint, satan	11.656	9,26%
DoS	apache2, back, land, mailbomb, neptune, pod, processtable, smurf, teardrop, udpstorm	45.927	36,46%
U2R	buffer_overflow, httptunnel, loadmodule, perl, ps, rootkit, sqlattack, xterm	52	0,04%
R2L	ftp_write, guess_passwd, imap, multihop, named, phf, sendmail, snmpgetattack, snmpguess, spy, warezclient, warezmaster, worm, xlock, xsnoop	995	0,79%
Tổng cộng		125.973	100%

III. GIẢI PHÁP

Để tìm ra tập các thuộc tính tối ưu phù hợp nhất với kiểu phân lớp cũng như phương pháp máy học. Trước tiên, tùy kiểu phân lớp, các thuộc tính sẽ được sắp thứ tự (giảm dần) dựa vào độ lợi thông tin. Sau đó, một thuật toán lựa chọn thuộc tính được áp dụng để lựa chọn các thuộc tính tối ưu phù hợp nhất ứng với từng phương pháp máy học. Phần tiếp sau trình bày sơ lược về độ lợi thông tin, các mô hình máy học, các tiêu chí đánh giá, cũng như các thuật toán lựa chọn thuộc tính được sử dụng trong thực nghiệm.

A. Độ lợi thông tin

Là độ đo thông tin được đề xuất sử dụng để xếp hạng độ quan trọng của các thuộc tính trong tập dữ liệu cần phân tích. Giả thiết:

S: Tập dữ liệu huấn luyện.

S_i: Lớp của tập các lớp C_i ($i=1,\dots,m$).

a_j: Giá trị thuộc tính A ($j=1,\dots,v$).

Chỉ số thông tin (Information) cho sự phân lớp:

$$I(S_1, S_2, \dots, S_m) = -\sum_{i=1}^{m} \frac{S_i}{S} \log_2 \left(\frac{S_i}{S}\right).$$

Giả sử thuộc tính A được chọn để huấn luyện, $A=\{S'_1, S'_2,\dots, S'_v\}$.

Khi đó, chỉ số thông tin mong muốn (Entropy) cho sự phân lớp của A được tính theo công thức:

$$Ent(A) = \sum_{j=1}^{v} \frac{S'_j}{S} \left(- \sum_{i=1}^{m} \frac{S'_{ij}}{S'_j} \log_2 \frac{S'_{ij}}{S'_j} \right).$$

Trong đó, S'_{ij} là các trường hợp phân lớp của S'.

Độ lợi thông tin (Information Gain) có được trên thuộc tính A được tính như sau:

$$Gain(A) = I(S_1, S_2, \dots, S_m) - Ent(A).$$

B. Các mô hình máy học

Phần này trình bày tóm tắt một số mô hình máy học chính [7] được sử dụng trong thực nghiệm để tìm ra tập các thuộc tính tối thiểu phù hợp nhất ứng với từng kiểu phân lớp:

(1) K láng giềng gần nhất (k-NN): là một trong những phương pháp truyền thống phi tham số và đơn giản nhất để phân lớp dữ liệu. Nó tính khoảng cách xấp xỉ giữa các điểm khác nhau dựa trên các dữ liệu đầu vào và sau đó chỉ định điểm không được dán nhãn vào lớp của k láng giềng gần nhất của nó. Trong quá trình phân lớp, k là một tham số quan trọng và các giá trị khác nhau k sẽ tạo ra các kết quả khác nhau. Nếu k lớn đáng kể, những láng giềng được sử dụng để dự đoán sẽ làm cho thời gian phân lớp lớn và ảnh hưởng đến tính chính xác của dự báo.

(2) Máy vectơ hỗ trợ (SVM): Là một giải thuật máy học dựa trên lý thuyết học thống kê do Vapnik (1998) đề xuất. Bài toán cơ bản của SVM là bài toán phân lớp loại 2 lớp: Cho trước n điểm trong không gian d chiều (mỗi điểm thuộc vào một lớp ký hiệu là +1 hoặc -1, mục đích của giải thuật SVM là tìm một siêu phẳng (hyperplane) phân hoạch tối ưu cho phép chia các điểm này thành hai phần sao cho các điểm cùng một lớp nằm về một phía với siêu phẳng này.

(3) Mạng nơron nhân tạo (ANN): Là mô hình xử lý thông tin mô phỏng hoạt động của hệ thống thần kinh sinh vật (Haykin, 1999), bao gồm số lượng lớn các nơ ron được gắn kết để xử lý thông tin. Mạng nơron nhiều lớp (MLP) là cấu trúc mạng nơron được sử dụng rộng rãi trong bài toán phân lớp. MLP gồm một lớp đầu, là một tập hợp các nút đầu vào; một hoặc nhiều lớp ẩn của các nút tính toán và một lớp đầu ra của các nút tính toán. Mỗi kết nối giữa các nơron được gắn với một trọng số được điều chỉnh trong suốt quá trình huấn luyện. Ngoài ra, một thuật toán lan truyền ngược cũng được sử dụng để đào tạo một MLP.

(4) Cây quyết định (DT): Với những ưu điểm của mình, DT được đánh giá là một công cụ mạnh, phổ biến và đặc biệt thích hợp cho khai khoáng dữ liệu nói chung và phân lớp dữ liệu nói riêng. Ngoài những ưu điểm như: xây dựng tương đối nhanh, đơn giản. Việc phân lớp dựa trên DT đạt được sự tương tự, đôi khi là chính xác hơn so với các phương pháp phân lớp khác.

C. Tiêu chí đánh giá

Nếu **FP** là số mẫu bị phân lớp sai là dương tính; **TP** là số mẫu được phân lớp đúng là dương tính; **FN** là số mẫu bị phân lớp sai là âm tính; **TN** là số mẫu được phân lớp đúng là âm tính. Việc đánh giá hiệu năng của các IDS được thực hiện qua việc đo và so sánh các chỉ số [8]:

- Accuracy = (TP + TN) / (TP + FP + TN + FN);
- Sensitivity = Recall = TPR = TP / (TP + FN);
- Specificity = TNR = TN / (TN + FP);
- Efficiency = (Sensitivity + Specificity) / 2;
- Độ chính xác cảnh báo: Precise = P = TP / (TP + FP);
- Thời gian huấn luyện và kiểm tra.

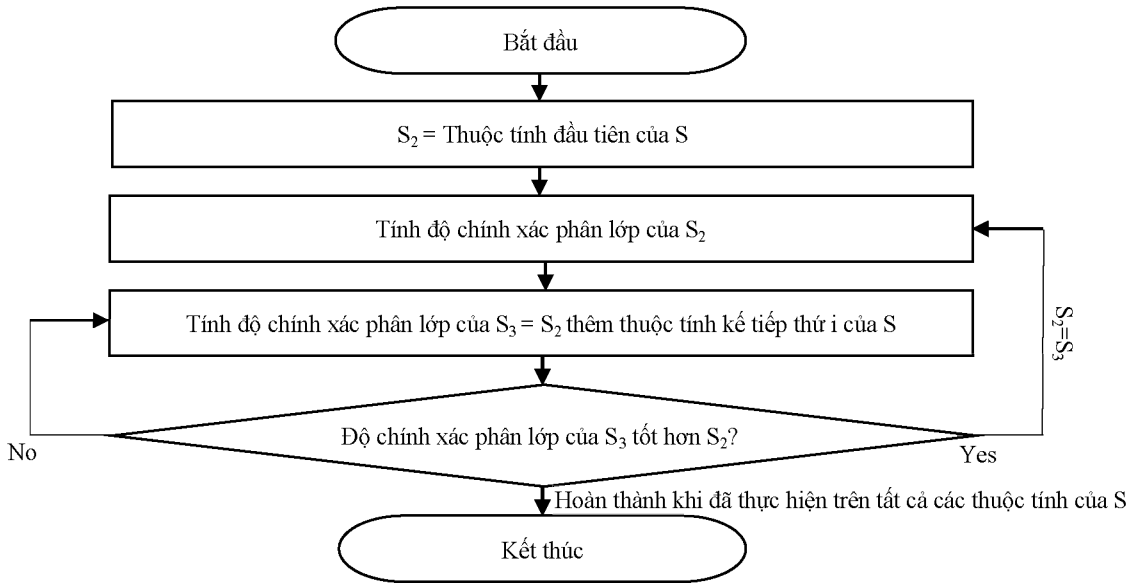
Có nhiều kỹ thuật đánh giá độ chính xác dự báo như: đánh giá chéo K-fold, Holdout, Re-substitution và Leave-one-out [9]. Trong đó, đánh giá chéo K-fold được xem là hiệu quả, phù hợp với các IDS. Theo đó, các bản ghi được phân ngẫu nhiên thành *k* tập con; một tập con được chỉ định là tập dữ liệu kiểm tra và các tập con còn lại được xử lý như tập dữ liệu huấn luyện. Sau đó, quá trình đánh giá chéo lặp lại *k* lần, cũng như độ chính xác phân lớp có thể được kiểm tra thông qua các độ chính xác phân lớp trung bình từ *k* lần đánh giá. Đánh giá chéo K-fold đặc biệt phù hợp với nguồn dữ liệu huấn luyện lớn, trái với đánh giá Leave-one-out, tốn nhiều thời gian để thực hiện.

D. Thuật toán chọn lựa thuộc tính

Có hai thuật toán lựa chọn thuộc tính được đề xuất thực hiện. Thuật toán đầu tiên, xuất phát từ tập các thuộc tính rỗng, sau đó các thuộc tính sẽ lần lượt được chọn để bổ sung nếu việc bổ sung thuộc tính đó giúp cải thiện độ chính xác phân lớp của hệ thống, các thuộc tính có độ lợi thông tin lớn hơn sẽ được chọn để bổ sung trước.

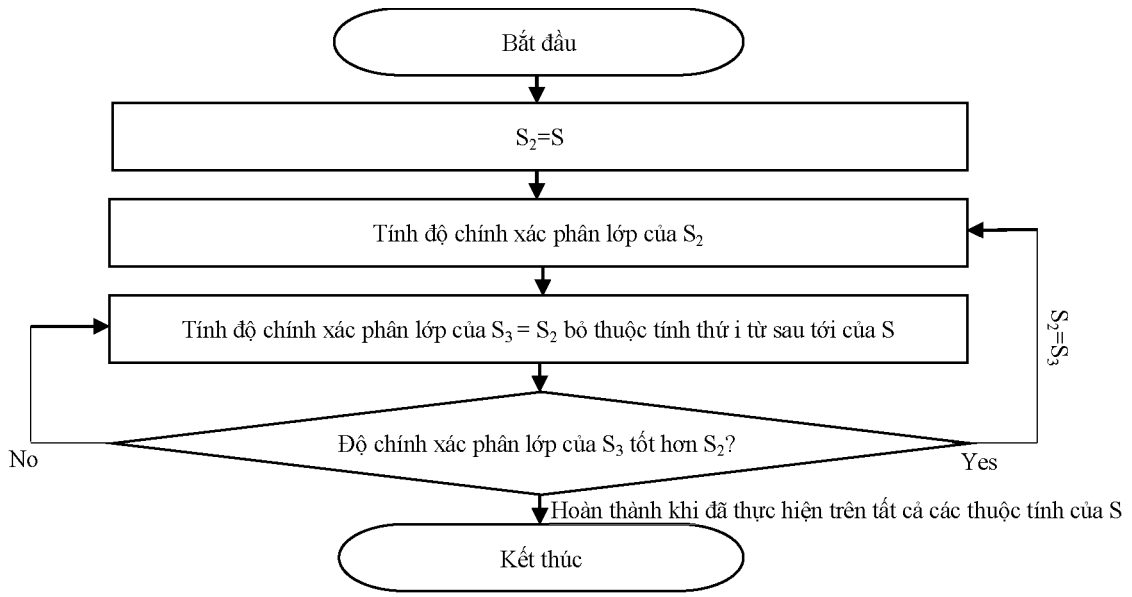
Thuật toán thứ hai, xuất phát từ tập đầy đủ 41 thuộc tính, sau đó các thuộc tính sẽ lần lượt được chọn để loại bỏ nếu việc loại bỏ thuộc tính đó giúp cải thiện độ chính xác phân lớp của hệ thống, các thuộc tính có độ lợi thông tin nhỏ hơn sẽ được chọn để loại bỏ trước. Lưu đồ giải thuật của mỗi thuật toán được trình bày như sau:

(1) Thuật toán Forward Selection Ranking (FSR):



Hình 1. Thuật toán lựa chọn thuộc tính FSR

(2) Thuật toán Backward Elimination Ranking (BER):



Hình 2. Thuật toán Backward Elimination Ra

IV. KẾT QUẢ THÍ NGHIỆM

Các chương trình, thuật toán trong thí nghiệm sử dụng ngôn ngữ lập trình C#, dựa trên thư viện, khung làm việc máy học Accord.NET (http://accord-framework.net) và cơ sở dữ liệu SQL Server 2014.

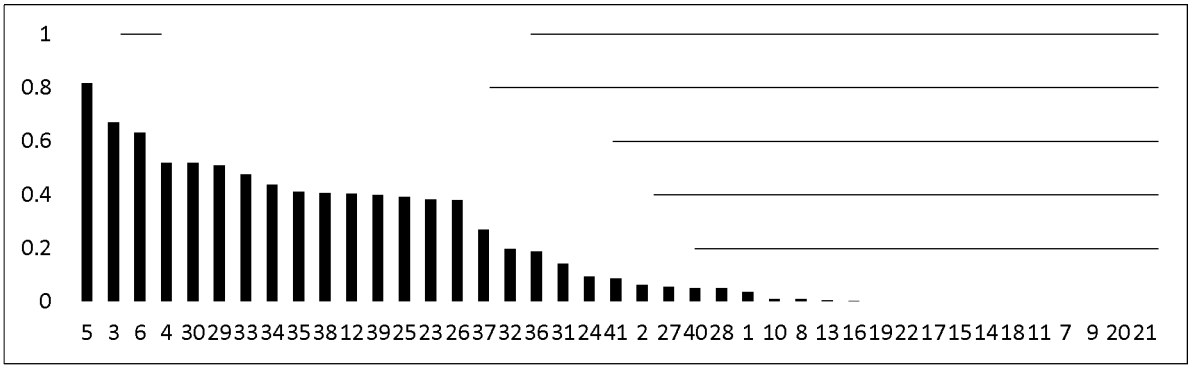
Thông tin chi tiết về các tập dữ liệu dùng trong thí nghiệm, số mẫu tin cụ thể ứng với mỗi kiểu tấn công trong mỗi tập dữ liệu được thống kê ở Bảng 3.

Bảng 3. Thông tin chi tiết 2 tập dữ liệu được sử dụng trong thí nghiệm.

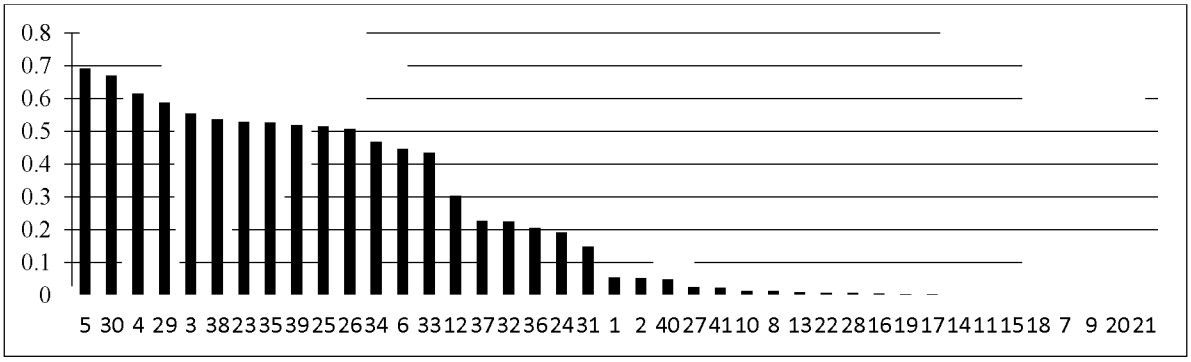
TT	Tên tập dữ liệu	Số mẫu tin ứng với từng kiểu tấn công					Tổng số mẫu tin
		Normal	DoS	Probe	U2R	R2L	
1	NSL-KDD	67.343	45.927	11.656	52	995	125.973
2	Probe-U2R-R2L	0	0	41.102	52	1.126	42.280

Trong đó, tập dữ liệu NSL-KDD được sử dụng cho các phân lớp Normal và DoS, tập dữ liệu Probe-U2R-R2L gồm tất cả các mẫu tin của các kiểu tấn công Probe, U2R và R2L rút trích từ tập dữ liệu KDD99, được sử dụng cho các phân lớp còn lại: Probe, U2R và R2L. Đó là do tỷ lệ mẫu tin của các kiểu tấn công Probe, U2R và R2L ở tập dữ liệu NSL-KDD ít, không đảm bảo độ chính xác phân lớp khi đánh giá hiệu quả của thuật toán.

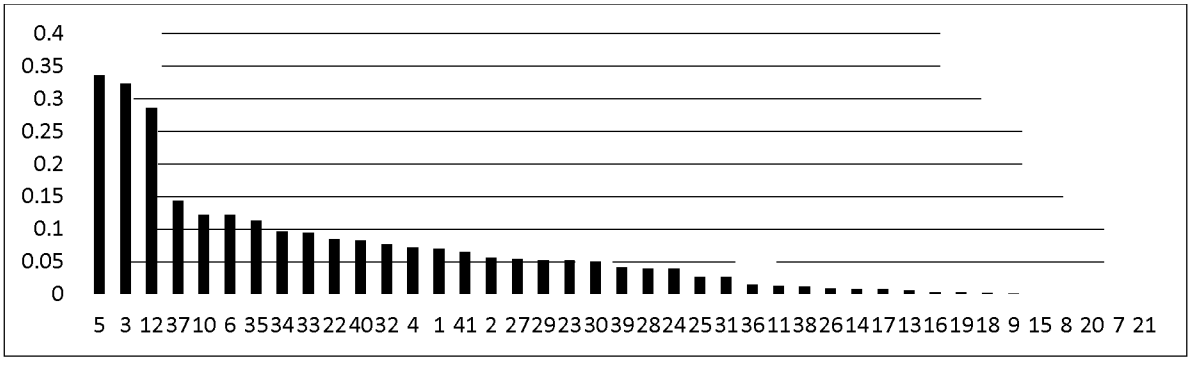
Sau đó, tùy kiểu phân lớp là Normal, DoS, Probe, U2R hoặc R2L, ta tiến hành tính toán độ lợi thông tin của từng thuộc tính. Kết quả tính toán và sắp xếp độ lợi thông tin của các thuộc tính khi phân lớp Normal được trình bày ở Hình 3. Tương tự cho các phân lớp DoS, Probe, U2R và R2L cũng được trình bày lần lượt ở các Hình 4, Hình 5, Hình 6 và Hình 7.



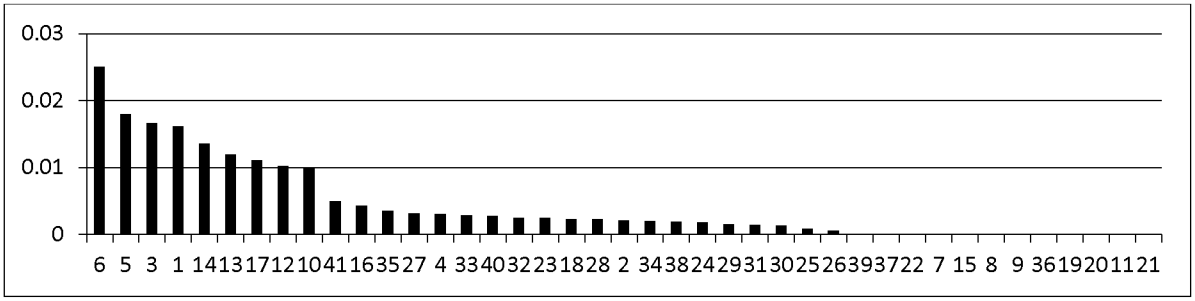
Hình 3. Độ lợi thông tin của các thuộc tính khi phân lớp Normal.



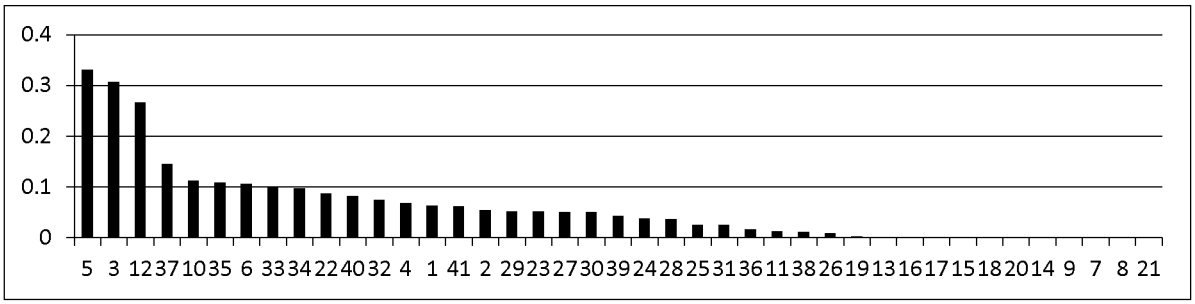
Hình 4. Độ lợi thông tin của các thuộc tính khi phân lớp DoS.



Hình 5. Độ lợi thông tin của các thuộc tính khi phân lớp Probe.



Hình 6. Độ lợi thông tin của các thuộc tính khi phân lớp U2R.



Hình 7. Độ lợi thông tin của các thuộc tính khi phân lớp R2L.

Sau đó, hai thuật toán lựa chọn thuộc tính BER và FSR được áp dụng để lựa chọn các thuộc tính tối ưu phù hợp nhất ứng với từng phương pháp máy học. Kết quả độ chính xác (Accuracy) và độ nhạy (Recall) phân lớp sử dụng đánh giá chéo 5-fold tốt nhất với mỗi kiểu phân lớp, mỗi thuật toán lựa chọn thuộc tính, cũng như mỗi mô hình máy học được trình bày ở Bảng 4, Bảng 5, Bảng 6 và Bảng 7. Theo đó, cột BER thể hiện độ chính xác (hoặc độ nhạy) phân lớp khi dùng thuật toán BER, cột FSR thể hiện độ chính xác (hoặc độ nhạy) phân lớp khi dùng thuật toán FSR và cột FULL thể hiện độ chính xác (hoặc độ nhạy) phân lớp khi không rút gọn thuộc tính.

Ở đây, đánh giá chéo k-fold với k=5 được chọn, vì nếu k quá lớn, tập huấn luyện sẽ lớn hơn nhiều so với tập kiểm tra, và kết quả đánh giá sẽ không phản ánh đúng bản chất của phương pháp máy học, đặc biệt là với các tập dữ liệu lớn. Đó cũng là lý do đánh giá chéo 5-fold được nhiều học giả lựa chọn [10].

Bảng 4. Độ chính xác phân lớp (Accuracy) với các kiểu phân lớp Normal và DoS.

TT	Bộ phân lớp	Phân lớp Normal			Phân lớp DoS		
		BER	FSR	FULL	BER	FSR	FULL
1	Naive Bayes	92.21%	93.31%	89.56%	97.83%	96.85%	82.92%
2	SVM	94.63%	93.62%	94.11%	97.55%	96.81%	97.48%
3	Cây quyết định (C4.5)	**99.74%**	99.63%	99.71%	**99.98%**	99.97%	99.97%
4	Mạng nơron	99.30%	98.96%	99.11%	99.90%	99.73%	99.85%
5	Hồi quy logistic	95.52%	94.95%	95.31%	97.97%	97.40%	97.95%
6	Hồi quy logistic đa thức	95.66%	94.78%	95.47%	98.68%	98.28%	98.36%
7	K láng giềng gần nhất	99.64%	99.68%	99.61%	99.91%	99.94%	99.88%

Bảng 5. Độ nhạy phân lớp (Recall) với các kiểu phân lớp Normal và DoS.

TT	Bộ phân lớp	Phân lớp Normal			Phân lớp DoS		
		BER	FSR	FULL	BER	FSR	FULL
1	Naive Bayes	92.59%	95.75%	88.41%	95.70%	94.77%	97.89%
2	SVM	95.95%	95.50%	95.90%	94.94%	93.62%	94.69%
3	Cây quyết định (C4.5)	**99.77%**	99.66%	99.73%	**99.97%**	99.96%	99.97%
4	Mạng nơron	99.35%	98.99%	99.18%	99.83%	99.56%	99.71%
5	Hồi quy logistic	96.71%	95.94%	96.41%	95.82%	95.50%	95.77%
6	Hồi quy logistic đa thức	96.28%	95.89%	96.41%	97.31%	96.85%	96.91%
7	K láng giềng gần nhất	99.66%	99.76%	99.66%	99.91%	99.94%	99.86%

Bảng 6. Độ chính xác phân lớp (Accuracy) với các kiểu phân lớp Probe, U2R và R2L.

TT	Bộ phân lớp	Phân lớp Probe			Phân lớp U2R			Phân lớp R2L		
		BER	FSR	FULL	BER	FSR	FULL	BER	FSR	FULL
1	Naive Bayes	99.58%	99.50%	99.56%	98.64%	89.69%	88.37%	99.49%	99.35%	99.36%
2	SVM	99.30%	99.06%	99.14%	99.75%	99.75%	99.74%	99.05%	98.81%	98.96%
3	Cây quyết định (C4.5)	**99.95%**	99.90%	99.86%	99.87%	99.92%	99.87%	**99.91%**	99.87%	99.83%
4	Mạng nơron	99.91%	99.87%	99.84%	99.86%	99.85%	99.84%	99.86%	99.83%	99.76%
5	Hồi quy logistic	99.30%	99.17%	99.27%	99.81%	99.79%	99.80%	99.22%	99.02%	99.17%
6	Hồi quy logistic đa thức	99.61%	99.17%	99.54%	99.81%	99.81%	99.80%	99.52%	99.19%	99.52%
7	K láng giềng gần nhất	99.92%	99.89%	99.90%	99.87%	**99.93%**	99.85%	99.87%	99.85%	99.81%

Bảng 7. Độ nhạy phân lớp (Recall) với các kiểu phân lớp Probe, U2R và R2L.

TT	Bộ phân lớp	Phân lớp Probe			Phân lớp U2R			Phân lớp R2L		
		BER	FSR	FULL	BER	FSR	FULL	BER	FSR	FULL
1	Naive Bayes	99.84%	99.75%	99.84%	59.52%	**93.38%**	82.98%	98.01%	96.19%	96.54%
2	SVM	99.79%	99.67%	99.72%	37.50%	40.31%	30.22%	91.42%	91.23%	90.65%
3	Cây quyết định (C4.5)	**99.98%**	99.96%	99.94%	76.26%	82.94%	78.19%	99.40%	99.11%	98.69%
4	Mạng nơron	99.95%	99.97%	99.93%	64.47%	56.74%	60.19%	99.20%	**99.43%**	98.09%
5	Hồi quy logistic	99.54%	99.39%	99.52%	48.10%	49.20%	51.91%	96.22%	95.80%	96.04%
6	Hồi quy logistic đa thức	99.83%	99.39%	99.80%	47.86%	47.66%	50.24%	96.29%	95.90%	96.14%
7	K láng giềng gần nhất	99.94%	99.97%	99.95%	66.79%	83.79%	67.34%	99.00%	98.69%	98.52%

Các độ chính xác (hoặc độ nhạy) phân lớp được tô đậm là kết quả tốt nhất tương ứng với từng kiểu phân lớp. Từ đó giúp ta xác định được phương pháp máy học, cũng như các thuộc tính được lựa chọn để độ chính xác phân lớp tương ứng với từng kiểu phân lớp là tốt nhất, Bảng 8 trình bày chi tiết kết quả đạt được đó. Theo đó, cột Thời gian huấn luyện thể hiện thời gian huấn luyện bộ phân lớp khi rút gọn thuộc tính và cột Thời gian tiết kiệm thể hiện % thời gian tiết kiệm được so với trường hợp huấn luyện bộ phân lớp khi không rút gọn thuộc tính. Với thuật toán K láng giềng gần nhất chúng tôi chọn sử dụng k=5, đây là kết quả được chọn sau một số các thí nghiệm với các giá trị khác nhau của k. Ở Bảng 7, dễ dàng nhận thấy ở các phân lớp U2R và R2L, độ nhạy của các phương pháp máy học dựa trên K láng giềng gần nhất và cây quyết định thấp hơn so với Naive Bayes và Mạng nơron, nhưng vẫn được chọn do số mẫu tin bị phân lớp sai là dương tính ở các phương pháp máy học Naive Bayes và Mạng nơron cao hơn.

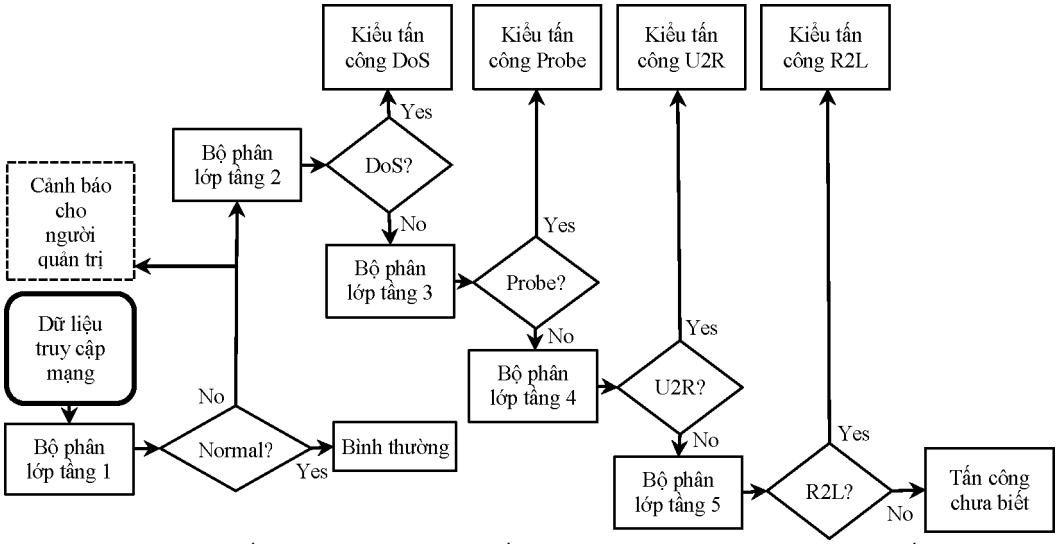
Bảng 8. Các thuộc tính và phương pháp máy học được lựa chọn phù hợp nhất với từng kiểu phân lớp.

Kiểu phân lớp	Phương pháp máy học	Các thuộc tính được lựa chọn	Accuracy (%)	Recall (%)	Thời gian huấn luyện (giây)	Thời gian tiết kiệm
Normal	Cây quyết định (C4.5)	11, 15, 17, 22, 19, 16, 13, 8, 1, 28, 40, 2, 41, 24, 36, 32, 37, 23, 39, 12, 38, 34, 33, 30, 4, 6, 3, 5	99.74%	99.77%	105	26%
DoS		13, 8, 41, 40, 2, 31, 32, 25, 39, 35, 38, 3, 29, 4, 30, 5	99.98%	99.97%	25	63%
Probe		36, 25, 23, 2, 1, 35, 6, 12, 3, 5	99.95%	99.98%	1	83%
U2R	K láng giềng gần nhất (k=5)	6, 5, 3, 14, 12, 41, 4, 28, 38, 11	99.93%	83.79%	2	82%
R2L	Cây quyết định (C4.5)	13, 19, 11, 36, 39, 23, 41, 32, 22, 6, 35, 10, 37, 12, 3, 5	99.91%	99.40%	2	60%

Với kết quả đạt được, ta có thể xây dựng một bộ phân lớp lai đa tầng như mô tả ở Hình 8 dựa trên mô hình phân đa lớp truyền thống One-Versus-Rest [11], với các tập thuộc tính được lựa chọn tối ưu trước khi phân lớp ở mỗi tầng.

Theo đó, dữ liệu truy cập mạng được đưa vào tầng 1, ở đó các thuộc tính phù hợp sẽ được chọn lựa và phân lớp là bình thường hoặc một cuộc tấn công, nếu truy cập là một cuộc tấn công, hệ thống sẽ cảnh báo cho người quản trị, đồng thời dữ liệu sẽ được chuyển sang tầng 2, ở đó các thuộc tính phù hợp lại được chọn lựa và phân lớp để xác định đó có phải là kiểu tấn công DoS hay không? Nếu không, dữ liệu sẽ được chuyển sang các tầng kế tiếp, các thuộc tính phù hợp lại được chọn lựa và phân lớp để xác định chính xác kiểu tấn công cụ thể, trường hợp không xác định được, thì đó là kiểu tấn công mới chưa được biết đến.

Kết quả thí nghiệm, độ chính xác dự báo tổng thể của bộ phân lớp lai đa tầng có rút gọn thuộc tính đạt 99.74% khi phân lớp các truy cập bình thường và 99.77% khi phân lớp các kiểu tấn công, tốt hơn so với việc không rút gọn thuộc tính có tỷ lệ tương ứng là 99.71% và 99.57%. Hơn thế nữa, về thời gian huấn luyện và kiểm tra, bộ phân lớp lai đa tầng có rút gọn thuộc tính giảm chỉ còn xấp xỉ 34% so với trường hợp không rút gọn thuộc tính.



Hình 8. Kiến trúc bộ phân lớp lai đa tầng với các tập thuộc tính được lựa chọn tối ưu.

V. KẾT LUẬN

Từ kết quả thí nghiệm, ta nhận thấy: do tính chất đặc thù dữ liệu của mỗi kiểu tấn công cũng như phương pháp máy học, phương pháp rút gọn thuộc tính sử dụng độ lợi thông tin kết hợp với các thuật toán rút gọn thuộc tính phù hợp cho ra các tập thuộc tính tối ưu phù hợp nhất. Qua đó, cải thiện độ chính xác dự báo tổng thể của bộ phân lớp lai đa tầng trong khi giảm thời gian huấn luyện và kiểm tra của toàn hệ thống, điều đó đồng nghĩa với việc giảm chi phí tính toán của các IDS, phù hợp với thực tế là khối lượng và tốc độ của dữ liệu mạng đang ngày càng lớn hơn. Đồng thời, kết quả thí nghiệm cũng đặt ra các vấn đề cần được tiếp tục nghiên cứu, đặc biệt là các nội dung:

(1) Việc nghiên cứu sử dụng các độ đo thông tin khác như: tỷ suất lợi ích (Gain Ratio), thuộc tính tương quan (Correlation Attribute),… để rút gọn thuộc tính, có thể sẽ đem lại hiệu năng cao hơn khi phát triển các IDS.

(2) Năng lực xử lý dữ liệu cũng như tính toán của hệ thống máy đóng vai trò quan trọng trong việc khai thác thuật toán cũng như phương pháp máy học. Từ đó nâng cao hiệu quả xử lý, tiếp cận theo hướng trí tuệ nhân tạo.

TÀI LIỆU THAM KHẢO

1. Al-Jarrah O. Y., Siddiqui A., et al. - Machine-Learning-Based Feature Selection Techniques for Large-Scale Network Intrusion Detection. In Distributed Computing Systems Workshops, 2014 IEEE 34th International Conference on, IEEE, 2014, 177-181.
2. Calix R. A., Sankaran R. - Feature Ranking and Support Vector Machines Classification Analysis of the NSL-KDD Intrusion Detection Corpus. Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, 2013, 292-295.
3. Moradi Koupaie H., Ibrahim S., Hosseinkhani J. - Outlier detection in stream data by machine learning and feature selection methods. International Journal of Advanced Computer Science and Information Technology (IJACSIT), 2014, **2** 17-24.
4. Patel S., Sondhi J. - A Review of Intrusion Detection Technique using Various Technique of Machine Learning and Feature Optimization Technique. International Journal of Computer Applications, 2014, **93**(14) 43-47.
5. Aburomma A. A., Reaz M. B. I. - Evolution of Intrusion Detection Systems Based on Machine Learning Methods. Australian Journal of Basic and Applied Sciences, **7**(7) 799-813.
6. Tavallaee, Mahbod; Bagheri, Ebrahim; Lu, Wei; Ghorbani, Ali A. - A detailed analysis of the KDD CUP 99 data set. 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009, pp.1-6.
7. Gaidhane R., Vaidya C., Raghuwanshi M. - Survey: Learning Techniques for Intrusion Detection System (IDS), International Journal of Advance Foundation and Research in Computer (IJAFRC), 2014, **1**(2) 21-28.
8. Marina Sokolova, Guy Lapalme - A systematic analysis of performance measures for classification tasks. Information Processing and Management 45, 2009, 427-437.
9. Li W., Liu Z. - A method of SVM with Normalization in Intrusion Detection. Procedia Environmental Sciences 11, 2011, Part A(0) 256-262.
10. Aburomma A. A., Reaz M. B. I. - Evolution of Intrusion Detection Systems Based on Machine Learning Methods. Australian Journal of Basic and Applied Sciences, 2013, 7(7) 799-813.
11. Neha Mehra, Surendra Gupta - Survey on multiclass classification methods. International Journal of Computer Science and Information Technologies, Vol. 4 (4), 2013, 572-576.

FEATURE SELECTION BASED ON INFORMATION GAIN TO IMPROVE PERFORMANCE OF NETWORK INTRUSION DETECTION SYSTEMS

Hoang Ngoc Thanh, Tran Van Lang

ABSTRACT: *The main function of the Intrusion Detection System (IDS) is to protect the system, analyze and predict the network access behavior of users. These behaviors are considered to be normal or an attack. Machine learning methods are used in IDS because of the ability to learn from past patterns in order to identify new patterns of attack. These methods are effective but have relatively high computational costs. Moreover, as the volume and velocity of network data grows rapidly, such computing cost issues must be resolved. This article refers to using Information Gain to reduce features of the dataset to be analyzed. Thanks to that, it helps to build IDS at a lower computational cost but with higher performance. The test results on the NSL-KDD99 data set use a 5-fold cross-validation assay: with a set of optimization attributes that match the layering as well as the machine learning method, the accuracy of the IDS has been improved with less computational time.*