MỘT GIẢI PHÁP PHÁT HIỆN XÂM NHẬP TRÁI PHÉP DỰA TRÊN PHƯƠNG PHÁP HỌC SÂU

Vũ Đình Thu*, Trịnh Khắc Linh, Trần Đức Sự

Tóm tắt: *Hệ thống phát hiện xâm nhập (Instruction Detection System - IDS) là một hệ thống được dùng để phát hiện các tấn công, xâm nhập mạng trái phép. Việc cảnh báo các tấn công chủ yếu dựa trên các mẫu sẵn có do vậy không thể cảnh báo được cuộc tấn công với các mẫu chưa biết. Bài báo này trình bày một hướng tiếp cận ứng dụng học sâu để phát hiện hành vi bất thường đối với hệ thống mạng được bảo vệ. Các thực nghiệm được thực hiện trên tập dữ liệu KDD cup 99 cho thấy mạng học sâu hiệu quả đối với phát hiện hành vi bất thường.*

Từ khóa: Máy học; Deep learning; Xâm nhập; Mã độc; Bất thường, KDD.

1. MỞ ĐẦU

Hệ thống phát hiện xâm nhập (Instruction Detection System - IDS) là một hệ phân tích, phát hiện các tấn công mạng, mã độc cho hệ thống mạng CNTT. IDS cũng có thể phân biệt các tấn công từ bên trong hay tấn công từ bên ngoài. IDS phát hiện tấn công dựa trên các mẫu tấn công đã biết (giống như cách các phần mềm diệt virus dựa vào các dấu hiệu đặc biệt để phát hiện và diệt virus. Việc dựa phát hiện dựa trên các mẫu đã biết có hạn chế đó là sẽ không phát hiện được những loại tấn công mới xuất hiện. Để phát hiện các loại tấn công mới phát hiện cần phải thực hiện phân tích các hành vi bất thường. Việc phát hiện tấn công mạng dựa trên phân tích các hành vi bất thường rất quan trọng trong việc phát hiện các loại tấn công có chủ đích sử dụng các loại mã độc mới với các kỹ thuật rất tinh vi.

Đã có có nhiều nghiên cứu liên quan đến phát hiện xâm nhập bất thường trong mạng máy tính. Về cơ bản, các hướng tiếp cận chính cho phát hiện xâm nhập bất thường là dựa vào đối sánh mẫu bằng cách định nghĩa các tập luật để làm mẫu so sánh đối chiếu với các dữ liệu mạng. Gần đây, đã có nhiều nghiên cứu phát hiện xâm nhập mạng bất thường dựa trên phương pháp học máy. Nghiên cứu của S. Chung, và K. Kim [11] đã xây dựng và kiểm thử mô hình phát hiện xâm nhập bằng cách áp dụng một tổ hợp nhiều thuật toán học máy như support vector machine (SVM), decision tree, phân lớp Naive Bayesian. Đồng thời cũng có nghiên cứu sử dụng phân cụm K-mean để phát hiện các lưu lượng độc hại. Nghiên cứu của Shin [12] sử dụng dụng K-mean phân cụm không phân cấp để tìm ra sự tương đồng và sau đó tìm ra các tham số để phát hiện tấn công DdoS và tấn công của sâu mạng Witty trong cùng thời gian. Nghiên cứu của Hatim [13] xây dựng mô hình học phát hiện tấn công mạng bằng cách lai thuật toán K-mean với SVM.

Gần đây đã có một số nghiên cứu áp dụng học sâu cho phát hiện xâm nhập bất thường, đây là hướng tiếp cận nâng cao so với các phương pháp học máy truyền thống. Nhà nghiên cứu Ni [14] đã sử dụng mạng DBNs (Deep belief networks) với tập dữ liệu KDD Cup 99 và cho kết quả độ chính xác cao hơn 6% so với SVM. Một nghiên cứu khác của S. Jo, H. Sung và B. Ahn [15] đã so sánh giữa FANN (Forward additive neural network) với SVM và đã chỉ ra FANN có độ chính xác cao hơn, độ phát hiện bất thường tốt hơn SVM.

Trong bài báo này sẽ trình bày việc áp dụng phương pháp học sâu sử dụng mô hình mạng DNN (Deep neural networks) cho việc học phân lớp các hành vi bất thường với tập dữ liệu sử dụng là KDD Cup 99.

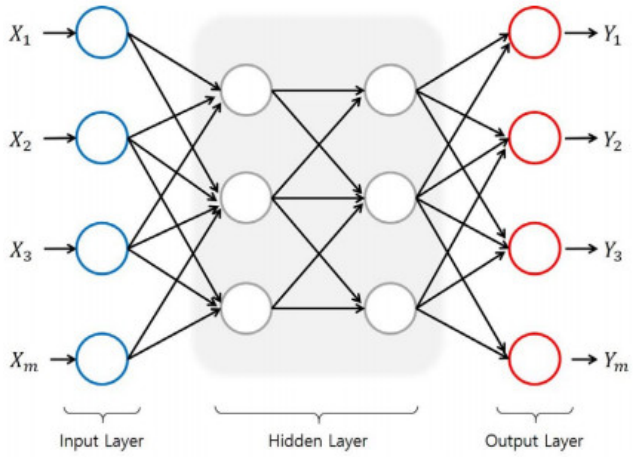
2. PHÂN LỚP CÁC HÀNH VI BẤT THƯỜNG SỬ DỤNG MẠNG DNN

2.1. Giới thiệu về học sâu

Học sâu là một phạm trù nhỏ của máy học, học sâu tập trung giải quyết các vấn đề liên quan đến mạng thần kinh nhân tạo nhằm nâng cấp các công nghệ như nhận diện giọng nói, thị giác máy tính và xử lý ngôn ngữ tự nhiên. Học sâu đang trở thành một trong những lĩnh vực đang thu hút được sự quan tâm trong khoa học máy tính. Chỉ trong vài năm, học sâu đã thúc đẩy tiến bộ trong đa dạng các lĩnh vực như nhận thức sự vật (object perception), dịch tự động (machine translation), nhận diện giọng nói,… những vấn đề từng rất khó khăn với các nhà nghiên cứu trí tuệ nhân tạo...

Học sâu là một lớp của các thuật toán máy học mà:

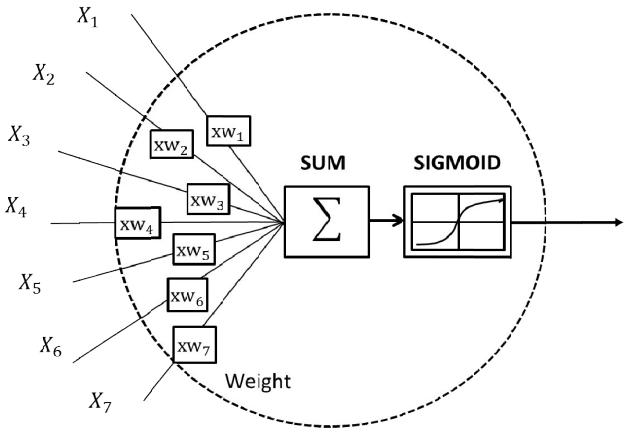
- Sử dụng một tầng nhiều lớp các đơn vị xử lý phi tuyến để trích tách đặc điểm và chuyển đổi. Mỗi lớp kế tiếp dùng đầu ra từ lớp trước làm đầu vào. Các thuật toán này có thể được giám sát hoặc không cần giám sát và các ứng dụng bao gồm các mô hình phân tích (không có giám sát) và phân loại (giám sát).
- Dựa trên học (không có giám sát) của nhiều cấp các đặc điểm hoặc đại diện của dữ liệu. Các tính năng cao cấp bắt nguồn từ các tính năng thấp cấp hơn để tạo thành một đại diện thứ bậc.
- Học nhiều cấp độ đại diện tương ứng với các mức độ trừu tượng khác nhau các mức độ hình thành một hệ thống phân cấp của các khái niệm.



***Hình 1**. Mô hình mạng DNN (Deep Neural Network).*

Các mô hình mạng học sâu gồm có DNN, DBNs (Deep belief networks), CNN (Convolutional neural network), RNN (Recurrent neural network)...Đối với mạng DNN thì cấu trúc của mạng mô phỏng hoạt động của tế bào thần kinh trong tự nhiên được minh họa trong hình 2, trong đó các tín hiệu kích hoạt $(x_0, x_1, ...)$ được gửi tới neural và được điều chỉnh nhân bởi các trọng số kết nối $(w_0, w_1, ...)$. Tổng các tín hiệu này tiếp tục được điều chỉnh bởi hệ số bias – thể hiện ngưỡng lọc nội

tại của tế bào. Cuối cùng, tín hiệu đầu ra của neural được biến đổi bởi hàm kích hoạt (activation function).

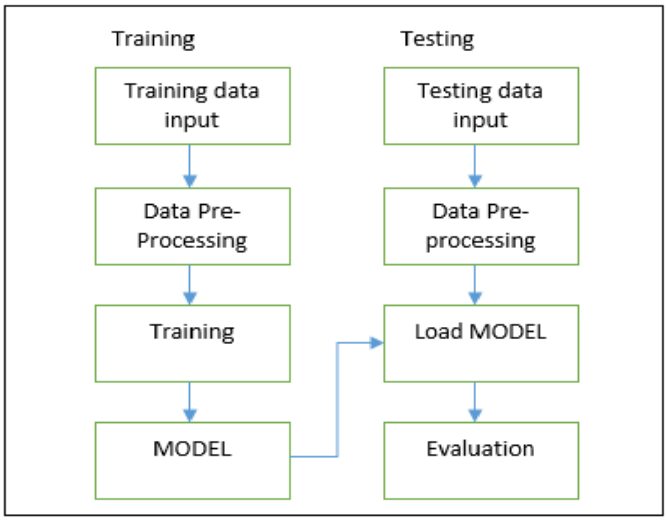


***Hình 2**. Nút hoạt động mạng DNN.*

Các neural được chia thành các lớp (layer), các lớp được sắp xếp theo thứ tự tuyến tính. Các neural trong cùng một lớp không được kết nối với nhau. Một neural thuộc lớp trước liên kết tới các neural thuộc lớp liền sau. Như vậy tín hiệu được truyền từ lớp đầu vào đến lớp đầu ra theo một hướng. Việc các neural giữa 2 lớp liên tiếp được kết nối như thế nào tùy theo bài toán cụ thể và topo mạng neural thường được lựa chọn dựa trên góc nhìn chủ quan của mô hình được đề xuất cho bài toán đó.

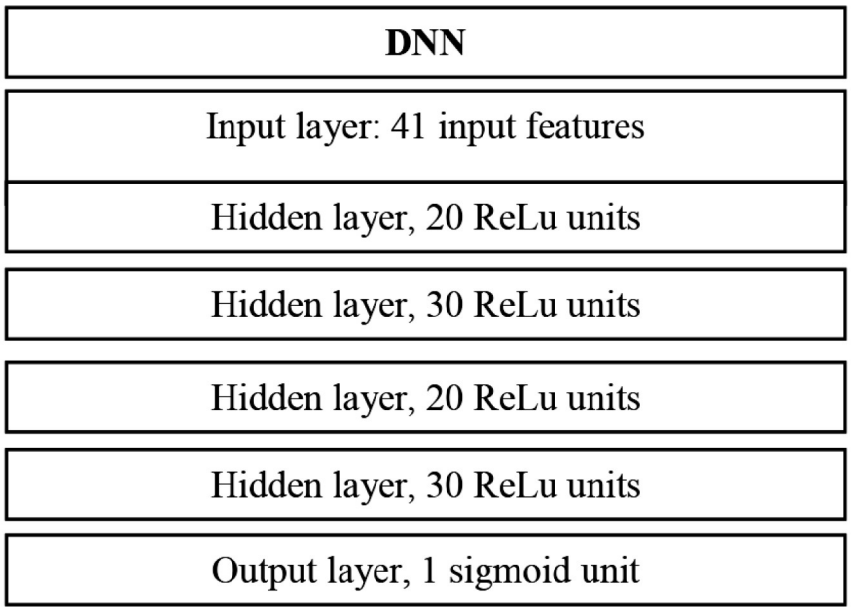
2.2. Phân lớp hành vi bất thường sử dụng mạng DNN

Trong bài báo này sẽ trình bày việc áp dụng mô hình mạng DNN cho việc học phân lớp theo quy trình ở hình 3. Ở bước huấn luyện, dữ liệu huấn luyện sẽ được xử lý trước khi huấn luyện. Các tham số của mô hình đã huấn luyện được lưu lại. Ở bước kiểm thử, dữ liệu kiểm thử được tiền xử lý, và tải các tham số của mô hình đã huấn luyện để kiểm thử trên tập dữ liệu và cho kết quả đánh giá.



***Hình 3**. Quy trình phát hiện xâm nhập bất thường sử dụng học máy.*

Trong quy trình trên, MODEL ở đây là mạng DNN được áp dụng với các tham số như sau: 4 lớp ẩn (hidden layers) và 100 node ẩn (hidden units), hàm kích hoạt là hàm ReLU cho các lớp ẩn[2]. Đồng thời sử dụng tối ưu Adam Optimizer[3] cho lan truyền ngược.



***Hình 4**. Các tham số sử dụng.*

Bộ dữ liệu sử dụng: Trong phần thử nghiệm này sử dụng tập dữ liệu KDD Cup 1999[5] được xây dựng từ năm 1998 của tổ chức DARPA (cục quốc phòng Mỹ và quản lý bởi Trung tâm thí nghiệm MIT Lincoln) và thường xuyên được cập nhật (KDD Cup newdata). Tập dữ liệu bao gồm một kiểu dữ liệu bình thường (normal) và 22 kiểu tấn công khác nhau được phân loại thành 4 lớp: từ chối dịch vụ (DoS), trinh sát hệ thống (Probe), chiếm quyền hệ thống (U2L) và khai thác điểm yếu (R2L). Chi tiết thông tin về bộ dữ liệu KDD Cup 99 được mô tả trong tài liệu của Aggarval, P., Sharma [16].

3. THỬ NGHIỆM, ĐÁNH GIÁ KẾT QUẢ

3.1 Xử lý dữ liệu

Dựa vào tập dữ liệu KDD99, lựa chọn các thuộc tính cơ bản từ các gói tin kết nối đến của một giao thức TCP, chẳng hạn như khoảng thời gian kết nối, kiểu giao thức, số lượng byte dữ liệu, các cờ để chỉ ra tình trạng lỗi kết nối hoặc bình thường, các các hoạt động tạo tập tin và một số hoạt động cố gắng truy cập vào hệ thống.

Trong tập dữ liệu KDD Cup 1999 thực hiện chọn 10% trong số dữ liệu này để làm thực nghiệm. Trong 10% bộ dữ liệu đào tạo của KDD 99 có ba giao thức khác biệt là TCP, UDP và ICMP, các nghiên cứu cho thấy rằng các giao thức này đều có liên quan đến bất kỳ cuộc tấn công mạng nào. Dữ liệu được xử lý biến đổi thành dữ liệu gồm có 41 thuộc tính như trong bảng 1 dưới đây.

***Bảng 1.** Bảng mô tả các thuộc tính trong tập dữ liệu KDD Cup 1999.*

TT	Tên thuộc tính	Mô tả	Kiểu thuộc tính
1	Duration	Khoảng thời gian (số giây) của kết nối.	Liên tục
2	protocol_type	Kiểu giao thức (TCP, UDP, ICMP).	Rời rạc
3	Service	Các dịch vụ trên mạng.	Rời rạc
4	Flag	Tình trạng bình thường hay lỗi kết nối.	Rời rạc
5	src_bytes	Số lượng byte dữ liệu từ nguồn tới đích.	Liên tục
6	dst_bytes	số lượng byte dữ liệu từ đích đến nguồn.	Liên tục
7	Land	1 nếu kết nối đến máy chủ, 0 ngược lại.	Rời rạc
8	wrong_fragment	Số sai trong phân mảnh.	Liên tục
9	Urgent	Số lượng gói tin khẩn cấp.	Liên tục
10	Hot	Số lượng "nóng" các chỉ số.	Liên tục
11	num_failed_logins	Số lần đăng nhập thất bại.	Liên tục
12	logged_in	1 nếu thành công, 0 nếu thất bại.	Rời rạc
13	num_compromised	Số điều kiện thoả hiệp.	Liên tục
14	root_shell	1 nếu gốc đạt được, 0 ngược lại.	Rời rạc
15	su_attempted	1 nếu là quyền root, 0 ngược lại.	Rời rạc
15	num_root	Số root truy cập.	Liên tục
17	num_file_creations	Số lượng tạo tập tin.	Liên tục
18	num_shells	Số lượng cảnh báo.	Liên tục
19	num_access_files	Số hoạt động trên các tập tin kiểm soát truy cập.	Liên tục
20	num_outbound_cmd	Số các lệnh gửi đi trong một phiên ftp.	Liên tục
21	Is_host_login	1 nếu đăng nhập vào thuộc danh sách nóng, 0 ngược lại.	Rời rạc
22	Is_guest_login	1 đăng nhập là một khách, 0 ngược lại.	Rời rạc
23	Count	Số lượng kết nối cùng một máy chủ cùng 2 giây.	Liên tục
24	srv_count	Số lượng kết nối cùng một dịch vụ trong 2 giây.	Liên tục
25	serror_rate	% các kết nối "SYN" lỗi.	Liên tục
26	srv_serror_rate	% các kết nối "SYN" lỗi.	Liên tục
27	rerror_rate	% của các kết nối "REJ" lỗi.	Liên tục
28	srv_serror_rate	% của các kết nối "REJ" lỗi.	Liên tục

29	same_srv_rate	% kết nối các dịch vụ tương tự.	Liên tục
30	diff_srv_rate	% các kết nối đến các dịch vụ khác nhau.	Liên tục
31	srv_diff_host_rate	% Các kết nối đến các máy chủ khác nhau.	Liên tục
32	dst_host_count	Số lượng kết nối đến máy chủ nguồn.	Liên tục
33	dst_host_srv_count	Số lượng kết nối từ nguồn đến đích.	Liên tục
34	dst_host_same_srv_rate	% kết nối máy chủ đích đến nguồn các dịch vụ tương tự	Liên tục
35	dst_host_diff_srv_rate	% máy chủ kết nối từ đích đến nguồn qua các dịch vụ khác nhau.	Liên tục
36	dst_host_same_srv_port _rate	% kết nối máy chủ đích đến nguồn các dịch vụ tương tự qua cổng.	Liên tục
37	dst_host_srv_diff_host_r ate	% máy chủ kết nối từ đích đến nguồn qua các dịch vụ khác nhau.	Liên tục
38	dst_host_serror_rate	% của các kết nối máy chủ đích "SYN" lỗi	Liên tục
39	dst_host_srv_serror_rate	% của các kết nối máy chủ đích đến nguồn "SYN" lỗi.	Liên tục
40	dst_host_rerror_rate	% của các kết nối máy chủ đích "REJ" lỗi	Liên tục
41	dst_host_srv_rerror_rate	% của các kết nối máy chủ đích đến nguồn "REJ" lỗi.	Liên tục

3.2. Công cụ cài đặt thử nghiệm

Trong phần cài đặt thử nghiệm, bài báo sử dụng thư viện Tensorflow để đặc tả các tham số của mạng DNN và thực hiện các thử nghiệm khác nhau.

3.3. Kết quả thực nghiệm

Thực nghiệm 1: Thực nghiệm này được thực hiện với các bộ dữ liệu huấn luyện kích thước khác nhau, cùng sử dụng số bước huấn luyện là số bước huấn luyện: 200 bước.

Bảng 2. *Kết quả thực nghiệm 1.*

Training data	**Accuracy**	**Actual label mean**	**Predictions mean**	**Loss**	**Precision**	**Recall**
10%	0.979887	0.231789	0.283409	0.959292	0.944581	0.970144
30%	0.971431	0.527020	0.52702	0.527044	0.97636	0.96926
60%	0.988054	0.759712	0.765936	0.172178	0.990597	0.993707
90%	0.987373	0.823635	0.82613	0.441808	0.99158	0.993102
100%	0.990855	0.803091	0.808048	0.183481	0.99881	0.989792

Kết quả trên cho thấy, với tập dữ liệu huấn luyện càng nhiều, độ chính xác thu được càng cao.

Thực nghiệm 2: Thực nghiệm với các bước huấn luyện mạng khác nhau

Bảng 3. *Kết quả thực nghiệm 2.*

Steps	Accuracy	Actual label mean	Predictions mean	Loss	Precision	Recall
10	0.966054	0.803091	0.793278	0.979437	0.969294	0.989063
50	0.985246	0.803091	0.806325	0.738506	0.989256	0.992406
100	0.983908	0.895682	0.896582	0.739842	0.992421	0.999427
150	0.992664	0.803091	0.817657	0.468709	0.996986	0.99387
200	0.990855	0.803091	0.804048	0.183481	0.99881	0.989792

Kết quả cho thấy khi tăng số bước huấn luyện thì giá trị loss (độ lỗi) giảm đi tương ứng, độ chính xác cũng tăng.

Thực nghiệm 3: Thực nghiệm so sánh với một số phương pháp khác sử dụng tập dữ liệu "10% KDD".

Bảng 4. *Kết quả thực nghiệm 3.*

Method	Accuracy
Decision Tree ID3[9]	0.9386
Support vector machines[8]	0.9345
Navie Bayes[10]	0.983125
Deep Neural Networks	0.97989

Kết quả thực nghiệm 3 cho thấy khi so sánh với các phương pháp học máy khác, thì trong thử nghiệm này phương pháp học sâu sử dụng mạng DNN cho độ chính xác cao hơn hầu hết các phương pháp, và chỉ thấp hơn không đáng kể so với phương pháp Navie Bayes.

4. KẾT LUẬN

Bài báo đã trình bày về vấn đề phát hiện xâm nhập trái phép và áp dụng một mô hình mạng học sâu để thử nghiệm đánh giá sự hiệu quả. Qua thử nghiệm đã cho kết quả tốt với mô hình thử nghiệm so với các phương pháp khác, điều đó cho thấy việc ứng dụng mạng học sâu sẽ mang lại hiệu quả tốt cho phát hiện xâm nhập bất thường và hoàn toàn có thể áp dụng trong thực tế. Để tăng cường độ chính xác cho việc phát hiện xâm nhập trái phép, cần tiến hành thực hiện trên các mô hình mạng học sâu với các tham số thử nghiệm khác nhau để lựa chọn ra bộ tham số phù hợp cho kết quả tốt nhất. Ngoài sử dụng mô hình mạng DNN thì có thể sử dụng các mô hình mạng khác như DBNs, CNN, RNN, để áp dụng trong bài toán phát hiện các hành vi bất thường, đây là các hướng nghiên cứu rất khả thi và phù hợp không chỉ riêng cho bài toán phát hiện các hành vi bất thường mà còn trong cả các lĩnh vực khác.

TÀI LIỆU THAM KHẢO

[1]. Jin Kim, Nara Shin, Seung Yeon Jo & Sang Hyun Kim, *"Method of Intrusion Detection using Deep Neural Network"*, Big Data and Smart Computing (BigComp). 2017 IEEE International Conference on 13-16 Feb. 2017.

[2]. G. Dahl, T. Sainath & G. Hinton, *"Improving deep neural networks for LVCSR using rectified linear units and dropout"*, 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8609-8613, 2013.

[3]. D. Kingma & J. Ba Adam, "*A method for stochastic optimization"*, arXiv preprint arXiv:1412.6980 2014.

[4]. N. Gao, L. Gao, Q. Gao, & H. Wang An, "*Intrusion Detection Model Based on Deep Belief Networks"*, Advanced Cloud and Big Data (CBD), 2014 Second International Conference on, pp. 247-252, 2014.

[5]. Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, "*A detailed analysis of the KDD CUP 99 data set"*. Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on 8-10 July 2009.

[6]. Rumelhart, David E, Hinton, Geoffrey E.; Williams, Ronald J. *"Learning representations by back-propagating errors"*. Nature. 323 (6088): 533–536. Bibcode:1986Natur.323..533R. doi:10.1038/323533a0

[7]. Tahmasebi, Pejman, Hezarkhani, Ardeshir (21 January 2011). *"Application of a Modular Feedforward Neural Network for Grade Estimation"*. Natural Resources Research. 20 (1): 25–,32. doi:10.1007/s11053-011-9135-3

[8]. V. Vapnik, "*The Nature of Statistical Learning Theory"*, Springer Verlag, 1995.

[9]. Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81–106

[10]. Rish, Irina (2001), "*An empirical study of the naive Bayes classifier*". IJCAI Workshop on Empirical Methods in AI.

[11]. S. Chung, & K. Kim, "*A Heuristic Approach to Enhance the performance of Intrusion Detection System using Machine Learning Algorithms"*, Proceedings of the Korea Institutes of Information Security and Cryptology Conference (CISC-W'15), 2015.

[12]. D. Shin, K. Choi, S. Chune & H. Choi, "*Malicious Traffic Detection Using K-means"*, The Journal of Korean Institute of Communications and Information Sciences, 41(2), pp. 277-284. 2016.

[13]. M. Tahir, W. Hassan, A. Md Said, N. Zakaria, N. Katuk, N. Kabir, M. Omar, O. hazali & N. Yahya, "*Hybrid machine learning technique for intrusion detection system"*, 5th International Conference on Computing and Informatics (ICOCI), 2015.

[14]. N. Gao, L. Gao, Q. Gao, & H. Wang, "*An Intrusion Detection Model Based on Deep Belief Networks"*, Advanced Cloud and Big Data (CBD), 2014 Second International Conference on, pp. 247-252, 2014.

[15]. S. Jo, H. Sung, & B. Ahn, "*A Comparative Study on the Performance of SVM and an Artificial Neural Network in Intrusion Detection"*, Journal of the Korea Academia-Industrial cooperation Society, 17(2), pp. 703-711, 2016.

[16]. Aggarval, P., Sharma, S.K., "*Analysis of KDD dataset attributes—class wise for intrusion detection"*. In: 3rd International Conference on Recent Trend in Computing 2015 (ICRTC-2015).

ABSTRACT

A METHOD FOR INTRUSION DETECTION BASED ON DEEP LEARNING

The Intrusion Detection System (IDS) is a system used to detect attacks and unauthorized network intrusion. The warning of attacks is primarily based on the available patterns so it is not possible to warn the attack with unknown patterns. This paper presents a deep learning approach to detecting unusual behavior for protected networks. Experiments performed on the KDD cup 99 data set shows that deep learning is effective for detecting abnormal behavior

Keywords: Machine Learning; Deep Learning; Instrusion; Malware; Abnormal; KDD.

Nhận bài ngày 21 tháng 01 năm 2019
Hoàn thiện ngày 18 tháng 3 năm 2019
Chấp nhận đăng ngày 25 tháng 3 năm 2019

Địa chỉ: Trung tâm Công nghệ thông tin và giám sát an ninh mạng – Ban Cơ yếu Chính phủ.
* Email: vudinhthu@gmail.com.