International Journal on Advanced Science Engineering Information Technology

Vol.8 (2018) No. 4
ISSN: 2088-5334

A Study on Machine Learning Based Light Weight Authentication Vector

Do-Hyeon Choi [#], Jung-Oh Park [*]

[#] *Department of Computer Science, Soongsil University, No.401, Sadang-ro, Dongjack-gu Seoul, 07027, Korea*
E-mail:cdhgod0@ssu.ac.kr

**Department of Paideia, SungKyul University, Manan-gu, Anyang-city, Gyeonggi-Do, 14097, Korea*
E-mail: jopark02@sungkyul.ac.kr

***Abstract*— Artificial Intelligence area has been rapidly advanced around the global companies such as Google, Amazon, IBM and so on. In addition, it is anticipated to facilitate the innovation in a variety of industries in the future. AI provides us with convenience in our lives, on the other hand, the valuable information on the subjects that utilize this has the potential to be exposed at anytime and anywhere. In the next advancement of AI area, the technical developments of the new security are required other than the existing methods. Generation and validation methods of light-weight authentication vector are suggested in this study to be used in many areas as an expanded security function. Upon the results of the capacity analysis, it was verified that efficient and safe security function could be performed using the existing machine learning algorithm. Authentication vector is designed to insert the encrypted data as variable according to the change of time. The security function was performed by comparing coordinate distance values within the authentication vector, and the internal structure was verified to optimize the performance cost required for data reverse search.**

***Keywords*— artificial intelligence; machine learning; authentication vector; virtualization; next generation security.**

I. INTRODUCTION

According to International Data Corporation, the AI market is anticipated to reach about $47 billion (W53 trillion) by 2020 with annual growth of 55.1% [1]. Recently, technology oriented global companies such as Google and Amazon have been expanding the business area into AI by lots of research(prediction, education, etc.) supports and heavy investment. With such a trend that the study of AI and is active in various fields, games, and education [2],[3]. OWL Cyber Security analyzed the level of cyber security among 500 companies in American Fortune, lately. As a result, they announced top 5 companies to be exposed to DarkNet including Google, Amazon, Apple, Facebook and eBay [4]. DarkNet has been known as the black market to trade drugs, prescription medicines, confidential information and so on in the internet [5]. As such, critical data of individuals and the companies have been traded and utilized in many hacking crimes.

This study suggested the new concept of light-weight authentication vector, totally different from the existing ones. It was designed to be operated as the sub-module level in the existing web server environment, and utilized with machine learning algorithm for the internal data processing of authentication vector. This consists of related studies, light-weight authentication vector, capacity analysis and conclusion in chapter 2 to 5, respectively. In this chapter, vulnerability of security is analyzed on the platform environment based on machine learning algorithm and AI.

A. Machine Learning Algorithm

Machine learning means the algorithm or processing to perform the improved works upon extraction and learning the patterns from the data without prior program [6]. Generally, machine learning can be divided by Supervised Learning and Unsupervised Learning to predict the operation after pattern analysis with the real data [7]. As seen in Figure 1, the problems to learn and find the specific data can be solved with Supervised Learning in case all the data are labelled. On the contrary, the problems to categorize the specific data into similar data group can be solved with Unsupervised Learning in case the data are mixed [8]. To date, more studies on Supervised Learning have been conducted in the machine learning area while the studies and investment are anticipated to be increased more with Unsupervised Learning that can understand the data without labelling [9].

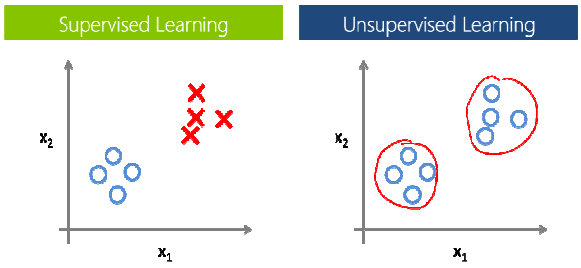

Fig. 1 Supervised Learning and Unsupervised Learning

To date, more studies on Supervised Learning have been conducted in the machine learning area while the studies and investment are anticipated to be increased more with Unsupervised Learning that can understand the data without labelling.

B. Vulnerability of Security Analyze

In this chapter, vulnerability of security is analyzed on the communication area of internal data in the virtualization technique. Virtualization is a core technique to utilize various resources virtually including applications, operation system, hardware and so on [10]. The hierarchy of virtualization security is highly related with hypervisor in the lower classes. Table 1 shows the status of hypervisor related technology such as Google, IBM, Amazon and so on. They support the representative hypervisors like Xen (Linux) and Hyper-V (Windows), and use mostly type 1 structure that is proper for processing of machine learning algorithm as the server based structure [11].

TABLE I
TECHNOLOGY STATUS OF HYPERVISOR

Flatform	**AI Service**	**Machine Learning Tool**	**Type**
Google Cloud	Google Cloud Machine Learning	TensorFlow	Type1
IBM Bluemix	Watson Analysis	IBM System ML	Type1
Amazon AWS	Azure Machine Learning	CNTK(Computational Network Tookit)	Type1
MS Azure	Amazon Machine Learning	DSSTNE(Deep Scalable Sparse Tensor Network Engine)	Type1

Since type 2 communicates the data through the host operating system, it has been known as its low efficiency on the communication, relatively [12]. Figure 2 shows the hypervisor structures of type 1, type 2 and their communication channels [13]. All the communication channels are through host operation system or hypervisor. Virtual Management Module (VMM) to perform the function of security classification exists inside or outside of the virtual machine.

Due to this, virtual machine and VMM can be exposed to the attacks from multiple channels. Also, the structures of the security classification internally developed in each platform vendors can be additional source of security vulnerability. There have been multiple studies on the attacks to the known vulnerable points including hardware Trojan attack, Distributed DoS (DDos) attack, internal client and VMM malicious code infection [14], [15], [16]. With respect to hypervisor security technology, little progress has been shown in Virtual Machine Introspection (VMI) or Agentless Technology [17], [18].

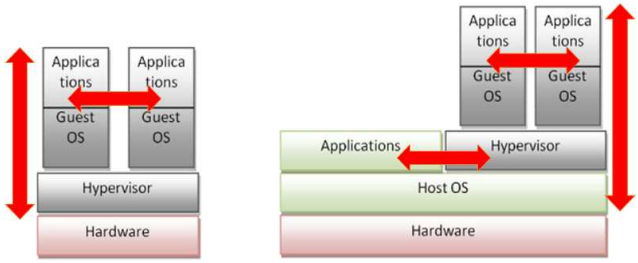

Fig. 2 Type1 and Type2 Hypervisor Structure of the Communicating Route

II. MATERIAL AND METHOD

Optimization items of machine learning algorithm are defined to be necessary in the design of light-weight authentication vector. Naive Bayes and K-Nearest Neighbor were used in this study as the machine learning algorithm. The below shows the optimization methods of machine leaning algorithm in this study.

① Naive Bayes: to define the data type in advance to be input to authentication vector to prepare primarily processed preliminary data.
② K-Nearest Neighbor: to extract only the shortest distance for the accuracy of distance value.

The followings show the requirements of data access control in machine learning algorithm.

① Data access: to define the collected data scope as the effective ones clearly.
② Distribute important data: Service provider and platform provider store the information of authentication vector, separately.
③ Machine learning data extraction range: to minimize the variables for the calculation capacity and the reliability of estimate results.

A. Generation of Light-Weight Authentication Vector and Selection of Initial Coordinate

Figure 3 shows the coordinate of initial authentication vector (Auth_V) and the selection method of matrix. Initially, the matrix of 60 × 60 (3600) is generated. Then, random number is input to each coordinate (x, y) to be shuffled. Initial coordinate (x=0, y=0) of authentication vector is used with time stamp based on the current time (second). For instance, 18 and 75 are selected upon separating 5 hours 12 minutes and 30 seconds (18750 seconds). Initial coordinate of 18 and 15 are used after dividing by 60, and reversed coordinate of 15 and 18 are newly selected. Then, coordinate of 15 and 9 is selected as the mid-point in the matrix. The selected mid-point coordinate is the position of real coded data input. After selection of initial authentication vector coordinate, the size of authentication vector is expanded to

input the data. The initial size of 60 × 60 is expanded to 3600 × 3600 (1 byte per each) at maximum. The coded data can be input up to the size of 12,960,000 bytes (12.96 Megabytes).

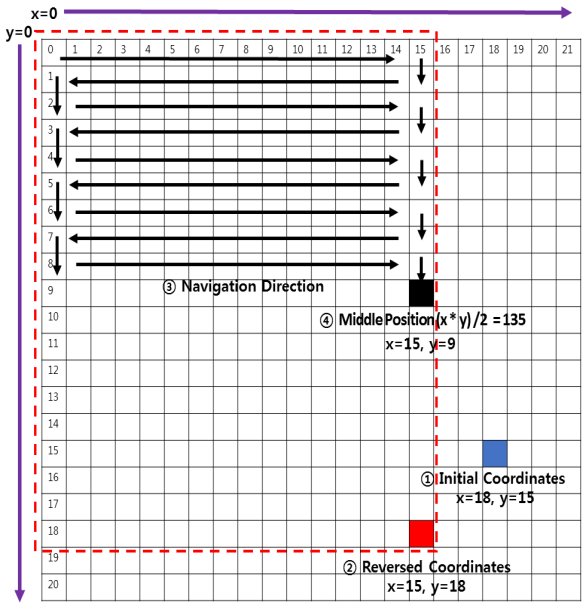

Fig. 3 Generation of Auth_V and Coordinates Selection

B. Data Coding and Selection of Input Coordinate

The user authentication information is connected and coded including ID, password, IP, operation system information, web-browser information and so on. Advanced Encryption Standard (AES) of National Institute of Standards and Technology (NIST) 256 was applied. Table 2 shows the items of authentication information (Auth_Info).

TABLE II
TECHNOLOGY STATUS OF HYPERVISOR

Type	Description
ID	-
PASSWORD	-
Uniq_NUM	Request Unique Number
SEQ_NUM	Sequence Number
IP	Internet Protocol Address
OS_INFO	Operating System Infomation
Web_Browser	Web Browser
Time Stamp	Login Time Information
Option 1	Temporary field 1
Option N	Temporary field N

As seen in Figure 4, coded data stating with 15, 9 (E_Auth) is input. In case that the size of input data is bigger than that of initial matrix 60 × 60, the data is input from the expanded authentication vector coordinate (15 × 60, 9 × 60) = 900, 540. The direction of input is the same with the one of initial searching, serially input upon rotating from the right to the left and from the top to the bottom.

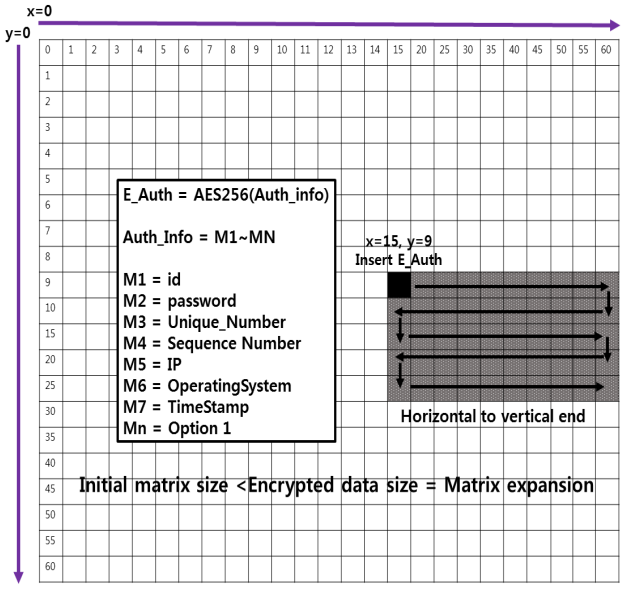

Fig. 4 E_Auth Insertion Process

New data is input continuously in authentication vector. Once the data are accumulated, input coordinate must be duplicated. Upon checking the size of previously input data, new data are input from the location of the last input. The essential information on input coordinate including midterm coordinates and duplicated coordinates is stored in the table of authentication vector. The table of authentication vector is used to test the coded data coordinate and extract them. As seen in Table 3, the table of authentication vector is stored in service provider during the maintenance of session.

TABLE III
AUTHENTICATION VECTOR TABLE

Session	Pattern Number	Center x, y	Request Number	Insert	State
S1	10	15, 9	1	1	T
S2	28	5, 9	3	2	T
S3	11	10, 10	2	0	F

The individual identifiers and pattern numbers are stored as the table information on the number input in the initial matrix, midterm position of coordinate, request number, duplication frequency and session status. Session status represents the status information of session connection. Authentication vector completing input of coded authentication information is transmitted to the platform provider on the web security standard protocol such as SSL/TLS and so on as a final. The actual authentication vector generated is stored in the platform provider (PP).

C. Validation on the Structure of Single Authentication Vector

If the coded data extraction process is repeated, the rate of data processing becomes inefficient. Validation on the structure of authentication vector provides with the methods to bypass the authentication process to be performed continuously. Figure 5 shows the conditional probabilities (CP1 and CP2) of authentication vector. Like formula (1), CP1 that pattern number (number of input coordinates) can

be appeared using Naive Bayes and CP2 that pattern number is duplicated in the matrix are calculated.

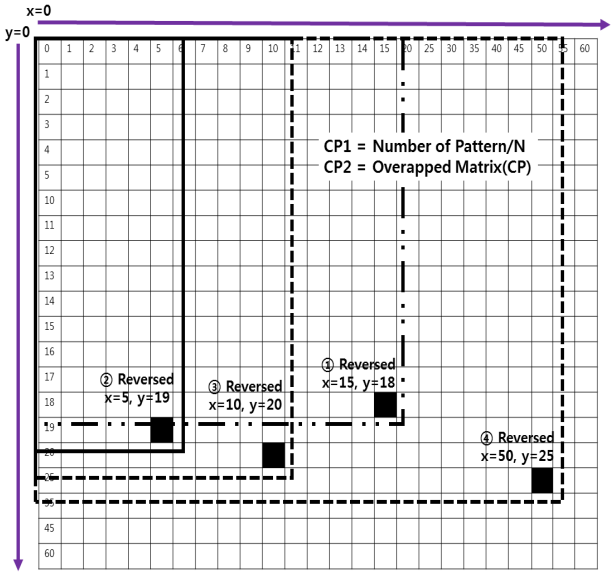

Fig. 5 Conditional Probability CP1 and CP2

$$CP1(A) = \frac{Count(PatternNumber)}{N}$$

$$CP2(A|B) = \frac{P(B|A)\,P1(A)}{P(B)} = \frac{P(A \cap B)}{P(B)} \quad (1)$$

For instance, CP1 of the matrix size 60, 60 (3600) input with 4 pattern numbers (4 authentication vectors) is 4 / 3600 = 0.001%. CP2 is ((1 / 4) × 0.001) / (4 / 3600) = 0.25% considering the variable I = 1 as the duplication number of the coordinates. This represents the probability with about 0.25% that one coordinate can be duplicated in four matrix coordinates in the whole authentication vector. The structure of normal authentication vector can be checked with number of coordinates and duplication frequency. The followings are the requirements for structural validation.

① Authentication number: In case normal authentication process is performed at least once, the structure can be validated.

② Size: The whole size of authentication vector as the size of the largest coordinate should be identical.

③ Number of duplications: Since the expansion (coordinate duplication at least once), the structure is validated.

The whole size can be calculated from the coordinate of the authentication information table (Center x, y). The size of initial authentication vector 60 × 60(3.6Kb) is very small. Hence, coordinate duplication must be occurred during the initial process of data input and authentication vector should be expanded.

D. Validation on the Distance of Multiple Authentication Vector

Authentication information can be occurred in multiple devices including personal computers, notebooks, smartphones, and so on. Since all the information types are different including static information (ID and password), dynamic information (IP, operation system, web browsers and so on), and serial information (serial number and requested number), new authentication vectors are generated whenever login process is performed. As in Figure 6, the distance is calculated using Euclid distance of K-Nearest Neighbor. Changes of each coordinate reflect the continuous changes of the distance by time. The method to calculate the coordinate is as in formula (2).

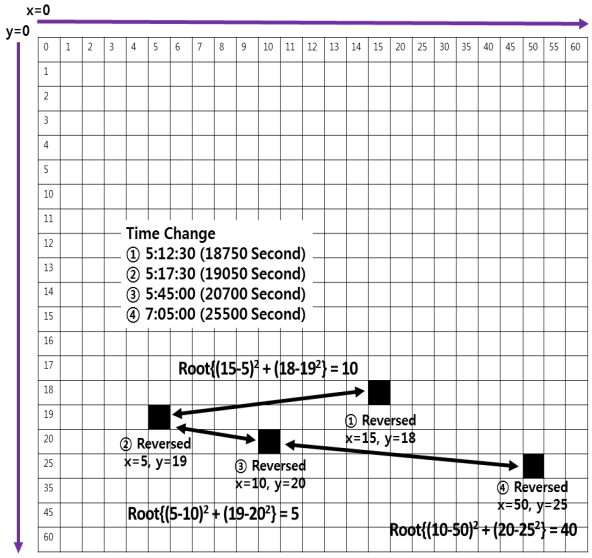

Fig. 6 Coordinates Distance between Auth_V

$$AuthDist(x,y) = \sqrt{Auth1(x^1 + y^1)^2 + Auth2(x^2 + y^2)^2} \quad (2)$$

For instance, the reverse coordinate in Fig 3 is 15, 18. New authentication vector is generated if additional login is performed after 5 minutes (5 hours 17 minutes and 30 seconds; 19020 seconds). Euclid distance between the reverse coordinate previously extracted 15, 18 and the current reverse coordinate 5, 19 is Root $\{(15-5)^2 + (18-19)^2\}$ = about 10.049. The followings are the requirements of calculation for the distance of coordinate.

① Authentication number: Only if authentication process vectors are existed with at least two, the distance can be calculated.

② Size: The size of each authentication vector is not more than the whole size.

③ Time: The distance of matrix coordinate varies by time continuously.

If the estimated distance of coordinate is not matched, the current session is terminated. The normal information of user authentication can be used to extract the access patterns as a statistical analysis.

E. Extraction of Coded Data

After confirming the test of authentication vector structure and validation of its distance value, coded data (E_Auth) are extracted. To search the input coordinate, the current information of authentication vector table is extracted from the table of authentication information. Figure 7 shows the method to extract the coded data coordinate.

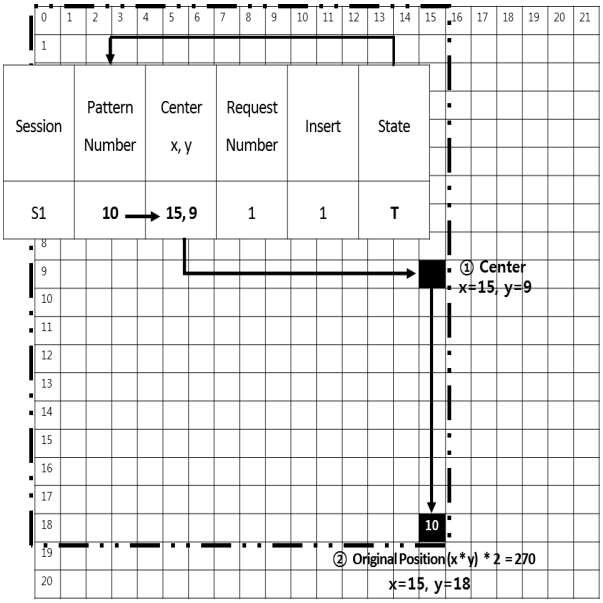

Fig. 7 Exracting Coordinates of E_Auth

Referring to the authentication vector table, reverse searching pattern number and input coordinate are initiated, and authentication vector is identified. After finding the result of reverse coordinate 15, 18, coded data are extracted.

III. Result and Discussion

Table 4 shows the environment of capacity analysis, organization elements and setup. Parameters of authentication vector during generation and verification processes should be separated independently for the security reason. User will perform registration and authentication by service provider (SP). SP stores the table of authentication information and platform provider (PP) stores authentication vector. It is designed to separately store two critical parameters.

TABLE IV
COMPONENTS AND PROGRAM SETTINGS

	Operating System	**Application**
User	Windows 10 64bit Intel® Core I7-5700HQ 2.70 GHz 16G Memory	Google Chrome 62.0.3, Internet Explorer 11.0.96
SP(AS)	Linux Kernel 3.13(Ubuntu 14.04 64bit) Intel® Core2Duo 2.66 GHz 8G Memory	Apache Tomcat 7.0 PHP 5.6.0
SP(MM)		Login(Authentication), MySQL 5(Table)
PP(HV)	Linux Kernel 3.13(Ubuntu 14.04 64bit) Intel® Core2Duo 2.66 GHz 8G Memory	Apache Spark 2.2.0 OpenStack(Mitaka)

A. Efficiency Analysis of Authentication Vector

It is to analyze the efficiency of authentication vector generation and verification processes. Table 5 and 6 show the results of capacity analysis including delayed time and data processing amount. Each value represents the mean value per second. Basic communication time including SSL/TLS authentication process is defined as 100% of 0.597 seconds. The ratios of authentication vector calculation time based on the existing communication time were once (63%), 10 times (75%), and 100 times (85%), which was confirmed that the actual total time was 1.036 seconds somewhat increased upon summation of the actual times.

Transmission rate of authentication vector processing was measured as 10.4Mb as a basis. 73.2Mb was occurred at 10 times at max and 100 times were excluded in the test due to the limitation of transmission rate of hard disk drive (HDD). One user cannot generate the multiple session over 100 times per session. In case of multiple logins in two to three devices, the number of generated authentication vectors is not more than 10. Upon the results of the capacity analysis, delayed time and calculation cost were somewhat increased on the multiple processing of authentication vector, however, it was confirmed that they would not affect the processing capacity significantly, less than 2 seconds on average.

TABLE V
GENERATING AUTH_V AND VERIFICATION TIME

Round(avg)	**Generation**	**Insert**	**Extract**	**Total Time**
1	0.102	0.023	0.025	0.150
10	0.206	0.023	0.091	0.320
100	0.313	0.027	0.099	0.439
100	TCP 0.096, SSL\TLS 0.501 (Handshake)			0.597

B. Safety Analysis of Authentication Vector

Major critical information is authentication vector table and authentication vector data. The followings show the results of security analysis on the table of authentication vector.

① Pattern number: Pattern number itself is the integer value without any meaning. It is not related to the coded data.

② Center x, y: Original authentication vector is required to search the position of input coordinate based on the mid-point coordinate.

③ Request number: Since it is used to array the pattern number, only request number has no meaning.

④ Insert: Duplication number is subject to change depending on the size of input data.

The followings show the results of security analysis on the authentication vector.

① Pattern number: Random number input during the generation of authentication vector is used as pattern number. Pattern number should be identified from the table of authentication information.

② Center x, y: There is no way to find the mid-point coordinate to coincide with the table of authentication information in the authentication vector.

③ Encrypted data: Input data is coded. When the session is exposed, the searching cost of previously shared passcode is additionally required.

④ Authentication vector distance: Whenever authentication pattern is generated, the distance is revised. Prior accumulated value of distance has no meaning, because login time is not constant to be varied continuously.

Authentication vector table of Service Provider (SP) and authentication vector of Platform Provider (PP) are stored in each server independently. Upon the results of each parameter analysis, light-weight authentication vector in this study provides with the difficulty in coding as follows.

① All the validation processes including structure test, distance test and coordinate extraction should be bypassed within the short time when one session is maintained.

② Due to the characteristics of matrix structure multiply connected, feasibility of authentication vector modulation is prevented.

③ Parameters of each critical information by SP and PP are diffused mutually. All the diffused data should be hacked.

④ Coordinate which is changed by time and the distance value between authentication vectors are safe against the attack of retransmission.

IV. Conclusion

This study suggests a light-weight authentication vector to solve a variety of vulnerabilities in virtualization environment. It does not depend on the server platform rather than the existing environment, it has the characteristics to be designed to perform the different security functions from the existing ones, such as structure validation, coordinate distance validation and so on, focusing on the virtualization environment. Also, it is complied with the capacity efficiency (1 session 10.4 Mb capacity, SLL / TLS 73% delay based on 100 sessions), and the safety of the pattern table and the authentication vector (pattern number independence, difficulty in distance and distance estimation) were verified.

References

[1] IDC (International Data Corporation), Worldwide Semiannual Cognitive/Artificial Intelligence Systems Spending Guide, 2016.

[2] Danial Hooshyar, Moslem Yousefi, and Heuiseok Lim, “Data-driven Approaches to Game Player Modeling: A Systematic Literature Review”, ACM Computing Surveys 50(6), 2017, pp. 1-19.

[3] Yeongwook Yang, Wonhee Yu, and Heuiseok Lim, “Predicting Second Language Proficiency Level Using Linguistic Cognitive Task and Machine Learning Techniques”, Wireless Personal Communications: An International Journal, 86(1), 2016, pp. 271-285.

[4] OWL Cyber Security, OWL Cybersecurity Launches Darknet Index Reranking the Fortune 500 by Darknet Footprint and Security Threat Levels, 2017.

[5] Dakota Rudesill, James Caverlee, Daniel Sui, The Deep Web and the Darknet: A Look Inside the Internet's Massive Black Box, Science+Technology Innovation Program, 2015.

[6] Peter Harrington, Machine Learning in Action, black & white, 2012.

[7] Shai Shalev-Shwartz and Shai Ben-David, “Understanding Machine Learning: From Thory to Algorithms”, Cambridge University Press, 2014.

[8] Leonardo Araujo dos Santos, Artificial Intelligence, GitBook, 2017.

[9] BENGIO, Yoshua, et al, Learning deep architectures for AI, Foundations and trends® in Machine Learning, 2(1), 2009, pp. 1-127.

[10] VMWARE, Virtualization Overview, Vmware White Paper, 2006.

[11] Fayyad-Kazan, Hasan, Luc Perneel, and Martin Timmerman, Benchmarking the performance of Microsoft Hyper-V server, VMware ESXi and Xen hypervisors, Journal of Emerging Trends in Computing and Information Sciences, 4(12), 2013, pp. 922-933.

[12] Graziano, Charles David. A performance analysis of Xen and KVM hypervisors for hosting the Xen Worlds Project, 2011.

[13] Zhang, Minjie, and Raj Jain, Virtualization security in data centers and clouds, http://www.cse.wustl.edu/~jain/index.html, 2011.

[14] Sabahi, Farzad, Secure virtualization for cloud environment using hypervisor-based technology, International Journal of Machine Learning and Computing, 2(1), 2012, pp. 39-45.

[15] Bhunia, Swarup, et al, Hardware Trojan attacks: threat analysis and countermeasures, Proceedings of the IEEE, 102(8), 2014, pp. 1229-1247.

[16] Nguyen, Anh M., et al, Mavmm: Lightweight and purpose built vmm for malware analysis, Computer Security Applications Conference, ACSAC'09, Annual. IEEE, 2009.

[17] Garfinkel, Tal, and Mendel Rosenblum, A Virtual Machine Introspection Based Architecture for Intrusion Detection, Ndss. 3, 2003, pp. 191-206.

[18] Hwang, T., Shin, Y., Son, K., & Park, H, Design of a hypervisor-based rootkit detection method for virtualized systems in cloud computing environments, In Proceedings of the 2013 AASRI Winter International Conference on Engineering and Technology, 2013, pp. 27-32.