Journal of Theoretical and Applied Information Technology
31st March 2020. Vol.98. No 06

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DEVELOPING THE SECURITY THREAT DETECTION MODEL FOR THE WEB SERVICE USING DEEP NEURAL NETWORK

HYUNCHUL CHANG[1] , SUNGBUM PARK[2]
[1]Cyber Security Center, Korea Public Finance Information Service, Seoul, Korea
[2]Graduate School of MOT, Hoseo University, Asan, Korea
E-mail: [1]suu8@naver.com, [2]parksb@nia.or.kr

ABSTRACT

The co-evolution of broadband networks and intelligent information system development ushers present golden days of web service. However, cyber attackers find loopholes easily for security threats under the web service environment. Detection of web service attacks requires to develop a new security threat counter-measure differentiated from the existing either signature-based or anomaly-based algorithms. In this regard, this research introduces a state-of-the-art intrusion detection model specialized in cyber-attacks on the web-service using the combination of deep neural network algorithms. Then, we evaluate the intrusion detection performance of the proposed deep neural network models with the Big Data of real time network traffic. Our research helps improve the limitation of existing intrusion detection systems and to overcome web service vulnerability on cyber threats.

Keywords: *Intrusion detection system, Web application firewall, Convolution Neural Network (CNN) Long Short-Term Memory (LSTM), C-LSTM.*

1. INTRODUCTION

The co-evolution of broadband networks and intelligent information system development ushers present golden days of web service [1]. Under the circumstances, however, cyber intruders easily find security threats pathways on the web service, with various attack methods, techniques, and targets [2]. The rate of web service attacks to the total cybersecurity threats more than doubled from 2014 to 2017 [3]. The existing signature-based analysis approach for intrusion detection has low detection capability for these types of attacks and only finds attacks exceeding certain thresholds level to reduce false-positives [4].

The anomaly detection approach, as an alternative, detects an attack or an abnormal behavior by judging the traffic deviating substantially from a rule for a predetermined normal behavior. Although anomaly detection model is an effective method for discovering new attack types, it is impossible to predefine all the normal activity rules and network protocols [12]. In order to overcome this problem, previous studies have used machine-learning techniques. Recently deep neural network-based studies have also been conducted [5-7] however; they found it challenging to obtain Big Data for model learning. Further improvement in the processing speed and accuracy is required for real-time intrusion detection. In addition, those researches concentrate on the specific types of attacks, such as Distributed Denial of Service (DDoS) or information system scanning, which occurs mainly in a lower (network) layer of the TCP/IP model.

In this regard, this research introduces the deep neural network-based intrusion detection models to deal with the threats on the application layer, which are difficult to detect due to the complex syntax of HTTP (Hypertext Transfer Protocol) of the web service protocol. To achieve the research objectives, we collected the big data of real-time network traffic from the web service server farm, and propose an intrusion detection method for a web service application to identify security threats that bypass the signature-based security systems. This research showed that the deep neural network technique provided excellent performance for the detection of web application intrusion that are not detected by signature-based intrusion detection system.

Journal of Theoretical and Applied Information Technology
31st March 2020. Vol.98. No 06

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Our research contents are as follows; In Chapter 2, we describe existing attack detection techniques and recent web service attack techniques in previous studies. We also discuss the research trends on the detection of security threats based on the deep neural network. Then, we propose the research methodology and models for building an intrusion detection system for web service. We also performed a model evaluation and derived the result in Chapter 3. In the last chapter, conclusions and research tasks for future studies are discussed.

2. THEORETICAL BACKGROUND

2.1 Web application attack and intrusion detection system

Unlike connection-oriented Internet services such as telnet, FTP, and e-mail, web services are connectionless open services. Even though web services require user authentication through login, most of the web services also have service pages for outsiders or non-members. In addition, web services have complex systems with hierarchical architecture [8]. A large number of web application programs that use scripting languages embedded in HTML, PHP, ASP, and JAVA are connected to the database and retrieve stored data on the website. Web contents may contain privacy such as a credit card number and personal information [9]. An attacker attempts to manipulate, destroy, or leak information without access authorization using a variety of web scripting languages.

2.2 Intrusion detection analysis technique

The Intrusion Detection System (IDS) is designed to detect malicious activities that may threaten the reliability and security of computer systems [10]. The existing IDS has either the signature-based analysis or anomaly-based analysis as the intrusion detection method [11]. The signature-based analysis technique finds a specific pattern of a known attack threat to analyze the list of already stored signatures by comparing the corresponding string with a regular expression. This technique is very successful if it keeps the database of signature patterns up to date, but it cannot detect unknown attacks or new malware such as zero-day attacks [11].

Compared to the signature-based analysis technique, abnormal-based analysis technique can detect attacks deviated from standard traffic patterns even for attacks whose signatures are not defined, and quickly finds new types of attacks [12]. However, the performance of an anomaly-based detection system depends on how well it is executed and tested on all protocols. Since the definition of the standard traffic patterns variable according to the protocols generated by specific vendor, it is challenging to define detection rules and to describe protocols [12].

Table 1: The screen of the demonstration system

Ref.	Alg.	Attack type	dataset	Acc
S.Chordia (2015)	K-means, KNN, DT	R2L, U2R, DoS, Probe	KDDcup 99	96.55
P.Jongsuebsuk (2013)	FL, GA	DoS, Probe	Real life	97.00
B.Sentilnavaki (2015)	SVM, GA	R2L, U2R, DoS, Probe	KDDCup 99	96.50-99.08
B.Masduki (2015)	SVM	R2L	KDDCup 99	96.08
A.Enache (2015)	SVM, BAT	Malicious	NL-KDD	99.38
S.Akbar (2012)	GA	R2L, U2R, DoS, Probe	KDDCup 99	92.60
A.Aziz (2013)	DT	DoS, Probe	NSL-KDD	82.00-64.00
Chao Lin (2015)	Cluster Center+KNN	DoS, Probe	KDDCup 99	99.98-99.99
A.Aburomman (2016)	SVM, KNN	R2L, U2R, DoS, Probe	KDDCup 99	87.41-91.69
Shi-jin Horng (2011)	SVM,Hierachy Clustering	R2L. U2R. DoS. Probe	KDDCup 99	95.72
E.Hodo (2016)	ANN	DDoS, DoS	Real-life	99.4
J.Kim (2016)	RNN(LSTM)	R2L. U2R, DoS, Probe	KDDCup 99	99.8
L.Amaldo (2017)	RF, FFNN, CNN,RNN	Malicious, Botnet	Real-life	70.00-94.30

2.3 Trends in research on AI-based intrusion detection

Recently, studies on artificial intelligence algorithms have been actively conducted to solve the problems of both signature-based and anomaly-based intrusion detection system

2.3.1 Machine Learning

Machine learning based methods enable the detection of new and subtle attacks occurring at the moment without extensive human-oriented inspection or intervention[2]. As shown in Table 1 representative machine learning models include Decision Tree, Bayesian network, SVM (Support

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Vector Machine), GA (Genetic Algorithm), and K-NN (k-nearest neighbor). Although studies above showed more than 90% accuracy, but most of them used test data sets or designed to detect already known specific type of attacks.

2.3.2 Deep Neural Network

The deep neural network-based intrusion detection systems have attracted considerable interest both in industry and in academia. Kim et al. (2016) applied the LSTM (Long-term and Short-term Memories) architecture to the RNN and trained the intrusion detection system using the KDDCup'99 dataset [7].

Compared to other intrusion detection classifiers, LSTM-RNN achieved an accuracy of 96.93% and a detection rate of 98.99%. Notably, a recent study by Arnaldo et al. (2017) compared the network intrusion detection performances of FFNN with RNN (LSTM), and CNN models by training them with the log data collected from a corporate security system. However, their study only focused on the network intrusion detection through the attributes of lower layers (IP address, etc.) of the TCP/IP model [13-15].

Unlike to network layer, web servers and applications are very complex systems, which increase the probability that vulnerability exists and makes it challenging to detect cyber threats. Also, a desirable intrusion detection system for web service needs to process noisy data with a high computation speed and accuracy since the noise level in a data set increases with a data set size [12]. Single technique has a limit to obtaining high performance considering these issues on the web service attack. To deal with these issues, it needs to compare and analyze the intrusion detection performances of hybrid deep neural network models with unstructured letter and number-based syntax structures

3. RESEARCH METHOD

3.1 Data collection and preprocessing

3.1.1 The Architecture for Network Traffic Collection

We collected and preprocessed real-time network traffic that had flown into the public website of the NEC (National Election Commission) in Korea. Figure 1 is a conceptual diagram of the network for building the data set. The firewall plays the role of primary access control for IP and the transport protocol (TCP, UDP, etc.), which is the third and fourth layer respectively. Is also responsible for defense against DDoS attacks that overload the homepage server. Traffic that passes through the firewall undergoes the second access control or the IPS (Intrusion Prevention System).

Based on pre-defined detection rules, IPS detects and blocks intrusion threats such as the inflow of malicious code, abnormal protocol, and a DDoS attack.

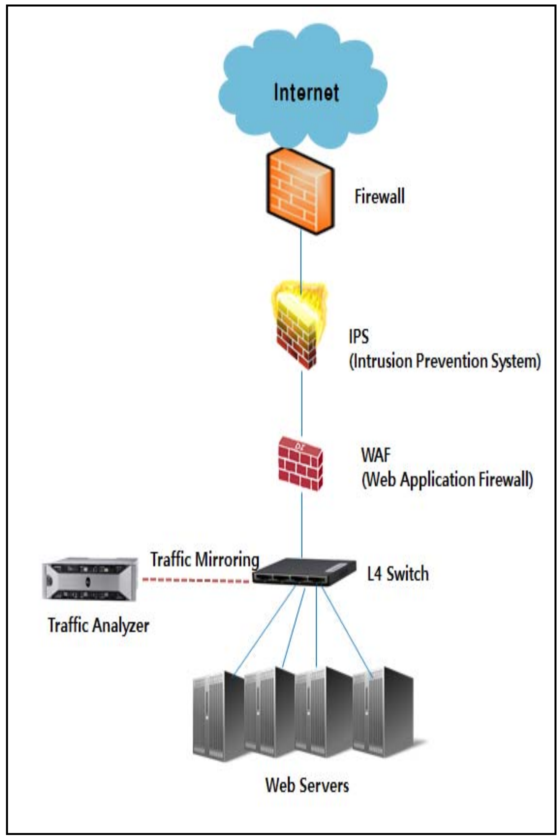


Figure 1: The architecture for network traffic collection

Then, the traffic goes through a WAF (Web Application Firewall) that defends against web service security threats. Although, a WAF detects and blocks various homepage threats defined by OWASP (Open Web Application Security Project, 2001), the same-for-all signature-based intrusion detection policy is only available. To collect and analyze all the traffic that passes through the security system and flows into the web server, we installed the traffic mirroring system in the server farm network switch composed of numerous homepage servers. We used Bro 2.5, an open-source, Unix-based network traffic monitoring system, for traffic analysis and classification for intrusion threat detection. It is a traffic classifier capable of all the port-based, payload-based, and host-based analysis. Also, it analyzes parsing network traffic to perform feature extraction at the application level [11].

3.1.2 Data collection and classification

We collect the traffic from 5 pm to 8 pm on May 5, 2017, and the amount of the collected traffic data reached about 13 Terabyte. All network traffic collected in real time is raw packet data. Of 14,215 data records in total, 13,942 records were classified as normal behaviors and 273 records as attacks. After 90% of the dataset was extracted as the learning set and the remaining 10% as the validation set.

Table 2: Research attributes from network traffic

No	attribute	meaning	value(example)
1	ts	The timestamp for when the request happened	1507257604.889 63
2	uid	Unique ID for the connection	CR4Kaj3HVuU OSPEN
3	orig_h	Source IP Address	192.168.0.30
4	orig_p	Source TCP port address	48477
5	rsp_h	Destination IP Address	11.1.1.3
6	resp_p	Destination tcp port Address	80
7	trans_depth	The pipelined depth into the connection of request/response transaction	integer (1, 3, 15 ...etc)
8	Method	The verb used in the HTTP request	GET, POST, HEAD, etc
9	Host	Value of the HOST header	www.bro.org
10	Uri	URI used in the request.	/board/index.htm l
11	referrer	Value of the "referer" header.	www.naver.com
12	version	Value of the version portion of the request	1.1
13	user_agent	Value of the User-Agent header from the client.	Mozilla/5.0
14	request_bod y_len	Actual uncompressed content size of the data transferred from the client	integer (default value = 0)
15	response_b ody_len	Actual uncompressed content size of the data transferred from the server	integer (default value = 0)
16	status_code	Status code returned by the server	200, 404, 300 ..etc.
17	status_msg	Status message returned by the server.	OK, Moved Temporary, no Content ..etc.

Table 2 shows the attribute as the web service traffic big data classified through Bro 2.5 [16]. They are not only related to the web service traffic information including the contents of the HTTP request and response packet but also display additional information such as the session information between the web client and the server.

Figure 2 shows an example of attributes 1 ~ 6. ts, UID and the 4-tuple features (origin_host, origin_port, response_host, response_port) have network connection information obtained during connection lifetime and is associated with other service traffic information (telnet, FTP, mail, etc.), thereby enabling identification of a user's activity history. Attributes 7 ~ 17 show the activity history generated in detail using UID as the key index. For example, Figure 2 [2-2] shows that a specific UID accessed "http://bro.org" by the "get" method using the Chrome browser

These are additive attributes that provide status information, such as the data size and messages, during the process of transmitting and receiving data between the client and the web server. They are not directly related to intrusion detection, but they are helpful when detailed traffic analysis is needed. Web services use a variety of scripting languages connected to the database. Users access the database through these scripting languages to retrieve, store, and modify information. This vulnerability in script composition causes many web service hacking incidents. Among these variables, this paper focuses on the detection of intrusion threats using 8, 10, 13 attributes (method, URI, user_agent) that are difficult to detect with existing intrusion detection systems. Figure 2-3 shows the general request values (character strings) transmitted from the client (browser) to the server when using a web service

Journal of Theoretical and Applied Information Technology
31st March 2020. Vol.98. No 06

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.2 Data Preprocessing

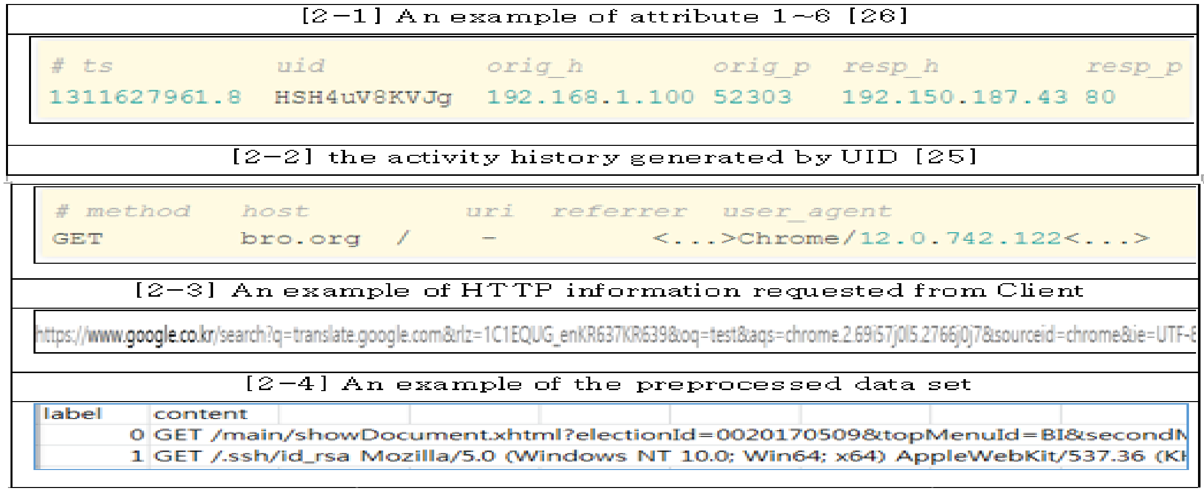


FIGURE 2. The example of research attributes

Attributes 8, 10, and 13 have time-series properties due to the HTTP protocol structure, and they are closely associated with each other. Therefore, we merged three attributes into one field to change them into a continuous sentence structure (Figure 2-4). We then added a label field to distinguish between normal (as 0) and abnormal (as 1) behaviors as shown in Figure 2-4. In the second step, after clarifying the distinction among words by removing special characters and stop characters, we converted words composed of text into numerical vector values using the word embedding method to apply them as the inputs to the neural network models.

4. IMPLEMENTATION

4.1 Intrusion Detection System development

Figure 3 shows the analysis model of the deep neural network-based intrusion detection system. System components largely divided into the traffic classifier, preprocessor, and intrusion detector. Bro 2.5, an open-source intrusion detection platform, was used as the traffic classifier. It collects all real-time traffic of the target to be protected, removes unnecessary information from it, and classifies the traffic data according to each

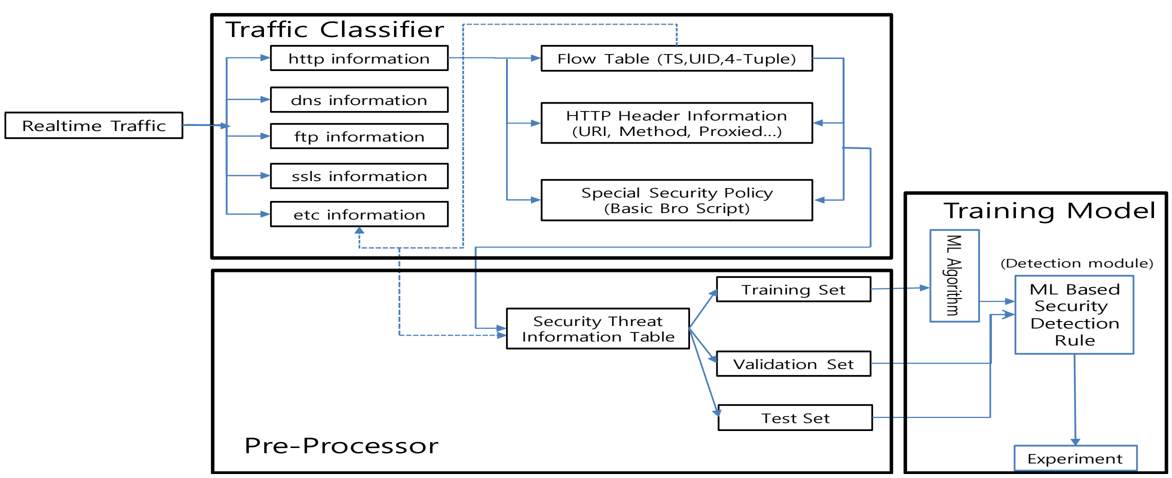


Figure 3: The data analysis model of Deep Neural Network

service protocol.

The preprocessor processes data through modification, deletion, and addition of the result data of the traffic classifier, extracts security threat information and constructs the input data set to the intrusion detector.

4.2 Training Model for Intrusion Detector

The syntax of the text format (HTTP header message information) in the dataset converted into vector values through data preprocessing. After the vector input values passed through the models of LSTM-RNN, CNN, and C-LSTM, we compare each result.

The comparison with machine learning models such as Decision Tree, SVM, etc. was excluded because the size of input data was variable depending on the length of the phrase or the number of words. Instead, it is a reasonable approach to compare deep neural network model in that the value of an input variable is not meaningful but the whole meaning of the text phrase must be mechanically understood. Figure 4 is a simplified representation of each deep learning structure.

First, words are extracted from the content attribute of the data set by word embedding method that are indexed to construct a lookup table. Embedding refers to mapping words to vector values of a specific dimension, and the converted vector values are updated through weighting during learning for the association and distinction of meanings among words. In the convolution layer used in CNN and C-LSTM, filters with the sizes of 3, 4, and 5 are used, and the number of filters is 128. The feature map is generated by extracting the local information while sliding the filter by 1 pixel at a time (stride=1) and by extracting as many features as the number of filters.

While the max-pooling process is performed after feature map creation in the CNN model, but the max-pooling process is omitted in the C-LSTM model. If max pooling is performed, it samples the input value by taking the maximum value in each feature map as the output of a fixed dimension with reduces size. However, in C-LSTM, the information extracted from the feature map is concatenated without dimension fixation or reduction, because the output value is used as the next input to LSTM.

The LSTM model is comprised of two cells, with 128 hidden units per LSTM cell. The values obtained from the LSTM passes through the fully connected layer, and the score corresponding to each class (0, 1) is calculated. Then, intrusion detection classification is performed, followed by error backpropagation for the results, and weights and other parameters are updated. Table 3 shows the summary of the hyper-parameters of the proposed model.

Table 3: Hyper parameter values in research models

model	Embedding size	Filter nim	Filter size	Conv layer	lstm cell	Hidden unit	drop out	Learning rate	Activation func.
CNN	256	3,4,5	128	1	-	-	50%	0.001	ReLU,Softmax
LSTM	256	-	-	-	2	128	50%	0.001	Tanh, sigmoid,Softmax
C-LSTM	256	3,4,5	128	1	2	128	50%	0.001	ReLU,sigmoid, Softmax

5. ANALYSIS AND RESULTS

The target variable has the binary classification system that categorizes a normal behavior as '0' and an attack as '1'. We used precision, recall, and F-score as well as the ROC curve as the indices using the confusion matrix that is used to evaluate intrusion detection performance. The results of the detection performance of each model based on the evaluation indices described in Table 4, 5, and 6.

Table 4: Hyper parameter values in research models

Model	Precision	Recall	Accuracy	F1 score
LSTM-RNN	0.838	0.966	0.997	0.898
CNN	0.899	0.888	0.995	0.893
C-LSTM	0.787	0.793	0.988	0.790

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 5: Confusion matrix for intrusion detection system analysis

Confusion Matrix		Predicted	
		Negative Class (normal)	Positive Class (attack)
Observed	Negative Class (normal)	TN (True Negative)	FP (False Positive)
	Positive Class (attack)	FN (False Negative)	TP (True Positive)

Table 6: Performance evaluation indices and calculation formulas

Metric	Calculation formulas
Precision	TP / (TP + FP)
Recall (Detection Rate)	TP / (TP + FN)
Accuracy	(TP + TN) / (TN + TP + FN +FP)
F1 score	(2*Precision*Recall)/ (precision+recall)

The analysis results were obtained by dividing the total of 14,215 data records into the sets of 256 input records (batch size = 256) to be learned at a time. This process was repeated 20 times (epoch = 20), and the results of each step (the total number of steps = 989) were averaged. As shown by the results, the LSTM model showed better overall performance than the other models, with a recall of 0.966, an accuracy of 0.997, and an F1 score of 0.898. In terms of precision, the CNN model showed excellent performance with 0.899 precision. On the other hand, the overall performance of C-LSTM in recall, accuracy, and precision was lower compared to other models. Table 5 shows the ROC curve, which represents the accuracy and loss of each step for each model. CNN, LSTM-RNN, and C-LSTM models showed the excellent performance for the intrusion detection in the web service envrironments.

Finally, Figure 5 shows the screen of the demonstration system representing the threat information through the machine-learning algorithm and it displays the information about the attacks that occurred during a specific period. The intrusion detector is trained to learn the data set defined according to the attack type with an optimal artificial intelligence technique to generate the security alarm when an intrusion attempt is detected

Thanks to the nature of the HTTP protocol, or a representative web service protocol, the length, and pattern of HTTP header messages used in cyberattacks are limited.

6. CONCLUSION

Through the real-time website traffic data analysis, this study showed that it is possible to conduct a big data collection as well as analysis in the presentation and application layer in the TCP/IP protocol. In addition, it was also shown that the

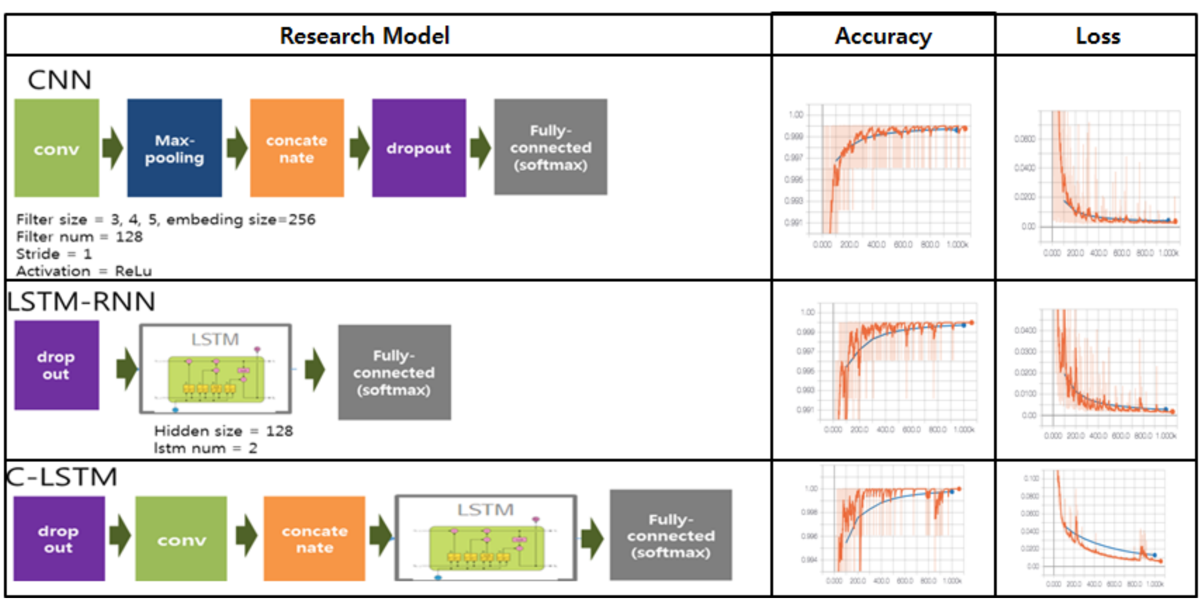


Figure 4: Research Model for Intrusion Detection

Journal of Theoretical and Applied Information Technology
31st March 2020. Vol.98. No 06

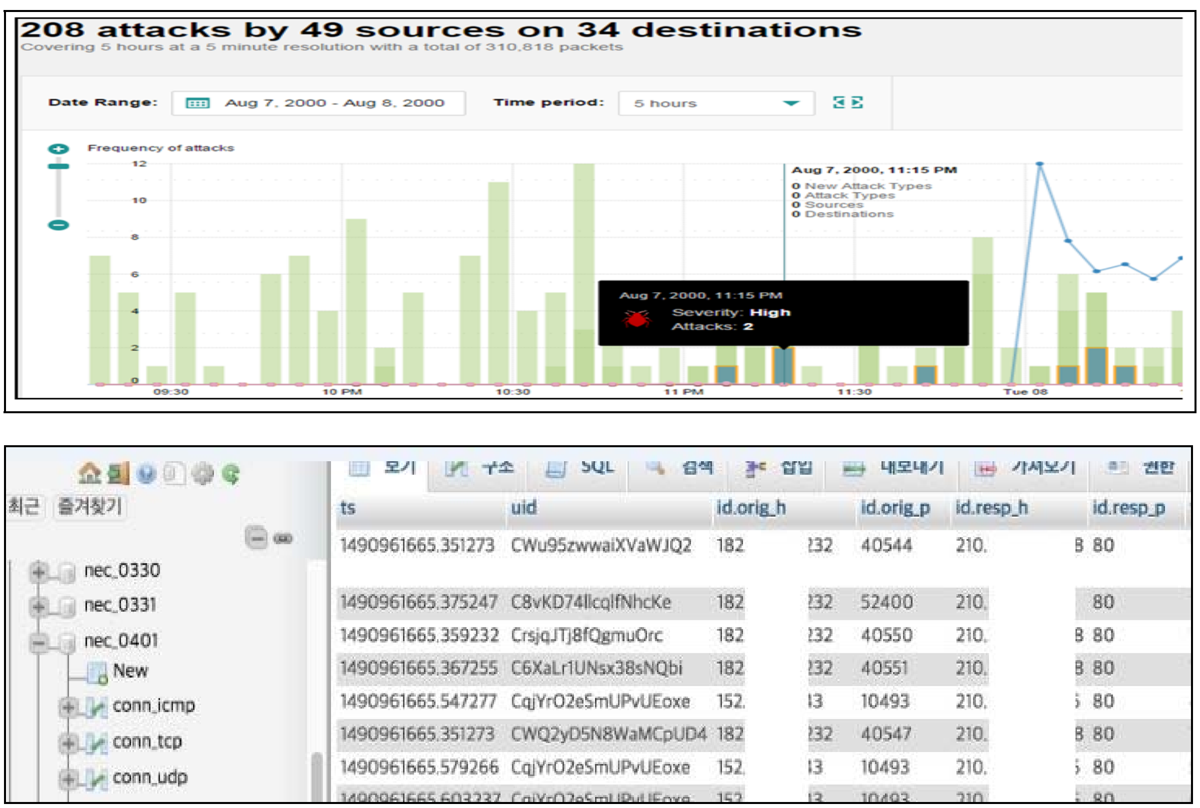
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

deep neural network technique, which has an excellent performance in the ranking of images or sentences, also shows excellent performance for the detection of web application intrusions, which are not possible, by signature-based intrusion detection systems. Signature-based technique have not been able to detect all types of attacks if the signature list of intrusion detection systems did not contain the right signature. To the best of our knowledge, the hybrid deep neural network model of CNN and LSTM has not been introduced in the field of web service intrusion detection but has proven to be an excellent classifier. Thus, there is no need to stack multiple hidden layers in the deep neural networks, so the burden of processing performance is low, which makes it possible to develop and commercialize an intrusion detection system with excellent performance. However, since it was not possible to control all of the variables in the model training process due to the nature of artificial neural networks, there is a possibility that different results will be obtained depending on the operating environment of the web service and the experimental setting.

Therefore, in order to generalize the results of this study, there is a need to verify them using the bigger size of web service data. Since cyber-attack types and attack techniques are variable, securing high-quality data sets is a significant success factor. In addition to web service hacking, cyber intrusions include various types of attacks such as malware infection, DDos attacks, phishing, and p harming. Therefore, research on intrusion detection based on deep learning remains a major task that needs to be conducted. Moreover, there is also a need to consider which deep learning algorithms are appropriate or optimal for the type of service or protocol of the information system to be protected.

ACKNOWLEDGMENTS

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2018-0-01417) supervised by the IITP (Institute for Information & communications Technology Promotion).



. *Figure 5: The Screen Of The Demonstration System*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This paper is the revision of conference proceedings manuscripts 'Design and Implementation of an Intrusion Detection System using Deep Neural Network', 15th International Conference on Machine Learning and Data Mining, MLDM 2019

Data Availability Statement

The data that support the findings of this study are available on request from the corresponding author.

REFRENCES:

[1] Storey, V.C. and I.-Y. Song, Big data technologies and management: What conceptual modeling can do. Data & Knowledge Engineering, 2017. 108: p. 50-67.

[2] Hodo, E., et al., Shallow and deep networks intrusion detection system: A taxonomy and survey. arXiv preprint arXiv:1701.02145, 2017.

[3] Nadav Avital, N.E. The State of Web Application Vulnerabilities in 2017. 2017 [cited 2018 09-28]; Available from: https://www.imperva.com/blog/2017/12/the-state-of-web-application-vulnerabilities-in-2017.

[4] Noel, S., D. Wijesekera, and C. Youman, Modern intrusion detection, data mining, and degrees of attack guilt, in Applications of data mining in computer security. 2002, Springer. p. 1-31.

[5] Fiore, U., et al., Network anomaly detection with the restricted Boltzmann machine. Neurocomputing, 2013. 122: p. 13-23.

[6] Gao, N., et al. An intrusion detection model based on deep belief networks. in Advanced Cloud and Big Data (CBD), 2014 Second International Conference on. 2014. IEEE.

[7] Kim, J., et al. Long short-term memory recurrent neural network classifier for intrusion detection. in Platform Technology and Service (PlatCon), 2016 International Conference on. 2016. IEEE.

[8] Yongho Lee, et al., Counter-measure development by web application threat analysis for safe web service, Korea Institute of Information Security and Cryptology, 2004. 14(4): p. 1-9.

[9] Skaruz, J. and F. Seredynski. Recurrent neural networks towards detection of SQL attacks. in Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International. 2007. IEEE.

[10] Chandola, V., A. Banerjee, and V. Kumar, Anomaly detection: A survey. ACM computing surveys (CSUR), 2009. 41(3): p. 15.

[11] Moya, M.A.C., Analysis and evaluation of the snort and bro network intrusion detection systems. Intrusion Detection System, Universidad Pontificia Comillas, 2008. 80: p. 80.

[12] Jyothsna, V., V.R. Prasad, and K.M. Prasad, A review of anomaly based intrusion detection systems. International Journal of Computer Applications, 2011. 28(7): p. 26-35.

[13] Jongsuebsuk, P., N. Wattanapongsakorn, and C. Charnsripinyo. Network intrusion detection with Fuzzy Genetic Algorithm for unknown attacks. in The International Conference on Information Networking 2013 (ICOIN). 2013. IEEE.

[14] Senthilnayaki, B., K. Venkatalakshmi, and A. Kannan. Intrusion detection using optimal genetic feature selection and SVM based classifier. in Signal Processing, Communication and Networking (ICSCN), 2015 3rd International Conference on. 2015. IEEE.

[15] Gupta, S. An effective model for anomaly IDS to improve the efficiency. in 2015 International Conference on Green Computing and Internet of Things (ICGCIoT). 2015. IEEE.

[16] Project, T.B., Monitoring HTTP Traffic with Bro. 2014.