An Anomaly Detection Method to Detect Web Attacks Using Stacked Auto-Encoder

1st Ali Moradi Vartouni
Faculty of Computer Engineering
K.N. Toosi University of Technology
Tehran, Iran
alimoradivartouni@ee.kntu.ac.ir

2nd Saeed Sedighian Kashi
Faculty of Computer Engineering
K.N. Toosi University of Technology
Tehran, Iran
sedighian@eetd.kntu.ac.ir

3rd Mohammad Teshnehlab
Faculty of Computer Engineering
K.N. Toosi University of Technology
Tehran, Iran
teshnehlab@eetd.kntu.ac.ir

***Abstract*—Network borne attacks are currently major threats to information security. Enormous efforts such as scanners, encryption devices, intrusion detection systems and firewalls have been made to mitigate these attacks. Web application firewalls use intrusion detection techniques to protect servers form HTTP traffic and, Machine learning algorithms have used based on anomaly detection in these firewalls. In this work, we proposed a method based on the deep neural network as feature learning method and isolation forest as a classifier. We compared this method with the methods does not include feature extraction models on CSIC 2010 data set. Additionally, we applied different activation function and learning for our deep neural network. Results show that deep models are more accurate than methods do not have feature extraction.**

***Index Terms*—Anomaly Detection, Web Application Firewall (WAF), Stacked Auto-Encoder (SAE), Isolation Forest**

I. Introduction

The security of information is one of the key problems in different domains of computer science. many efforts such as Intrusion Detection System (IDS) and firewall have been made to establish various security solutions. Firewalls and IDSs in the network layer usually do not examine communication packets in the application layer. The Hypertext Transfer Protocol (HTTP) has evolved to one of the most employed application layer protocols on the Internet. In order to detect and mitigate Web services threat, Web Application Firewall (WAF) is a tool to identify and prevent many types of attacks, such as XSS and SQL injection. It is used on web applications to prevent information leakage of organizations.

In general, the IDS methods are divided into two groups based on signature and anomaly detection [1]. The signature-based approach is usually simple and effective for known attacks, but the method is inefficient facing new attacks. The second type of IDSs, anomaly detection approach, looks for a pattern that does not conform the expected behavior using Machine Learning and Pattern Recognition methods. These algorithms have the advantage that they can detect new types of intrusions as deviations from normal behavior.

There are main steps for anomaly detection in web application firewalls based on machine learning methods include attribute construction, feature selection, and anomaly detection classifier. To construct feature from the web request, [2] have been analyzed statistical characteristics of HTTP traffic. Another feature construction method so-called tokenization has been introduced by [3], [4]. n-gram also is used by [5], [6] to characterize the distribution of content and structure of payload.

The number of features increases exponentially with n of n-gram model. Thus, it is necessary to consider solutions to deal with the curse of dimensionality problem. The filter-based feature selection provided in [7]–[9]. An effective method proposed by [10], [11] extracts exemplars from HTTP traffic before the detection model is built. The affinity propagation is employed to dynamic clustering and extract the exemplars from HTTP traffic without the need to specify the number of clusters.

Neural network approaches have been widely used in several works such as [12] for web applications firewalls. Deep neural network is very useful for real-world applications such as speech recognition, computer vision, and natural language processing [13]. Deep learning algorithms extract automated complex data representation at high levels of abstraction [14]. Researchers have applied deep learning based on Stacked Auto-Encoder (SAE) for the implementation of various intrusion detection systems [15] and character-level convolutional neural network for web attacks detection [16].

In this paper, we propose a web application firewall based on anomaly detection using machine learning algorithms. Our approach analyzes the HTTP traffic and to construct features from HTTP data, n-gram model that is based on character is applied. To reduce the dimensionality of the problem, SAE with different configurations are implemented to extract relevant features from data. In the final, an isolation forest

978-1-5386-2836-2/18/$31.00 ©2018 IEEE

method is applied to detect anomalies.

This paper is organized as follows. After the introduction, section two considers the architecture of the proposed system. We describe our experiments and result in section three and present concluding remarks in section four.

II. Methodology

Deep learning is a powerful set of techniques for learning in neural networks. What distinguishes neural networks from one another can be the architecture layer and network structure, neuron activator functions, loss or cost function and optimization algorithm of that network. For example, we can use fully-connected or convolutional layers with different size of layers and neurons, sigmoid or ReLU activation function for neurons, quadratic or cross-entropy as loss function and stochastic gradient descent as an optimization. Whereas in this paper are used different activation functions and various optimization, we describe these methods briefly.

A. Preliminaries

An autoencoder neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs i.e. $y^{(i)} = x^{(i)} for\ i = 1...N$ [17]. An autoencoder always consists of two parts, the encoder $h = f(x)$ and the reconstruction part $r = g(h)$.

$f(.)$ and $g(.)$ are the activation function of neurons which g usually is linear. But f can be various activation functions [18]. Figure 1 demonstrate the plot of three activation functions namely sigmoid, linear rectifier (ReLU) and soft linear rectifier (softplus).

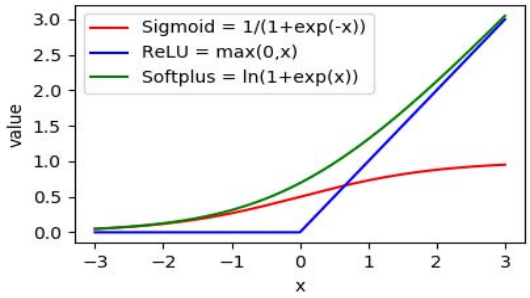

Fig. 1. The plot of three activation functions: sigmoid, ReLU and softplus

the learning process is to find a value for parameter vector Θ to minimize reconstruction error as in (1):

$$J_{AE}(\Theta) = \sum_{i} L(x^{(i)}, r^{(i)}) \tag{1}$$

Stochastic gradient descent (SGD) performs a parameter update for each training example. In general, a parameter Θ updates based on gradient descent by (2):

$$\Theta_{(t+1)} = \Theta_t - \eta.\nabla_{\Theta} J(\Theta) \tag{2}$$

Adagrad [19], RmsProp [20], Adam [21] are algorithms for gradient-based optimization that update the learning rate to the parameters adaptively.

B. Proposed model

HTTP request gets from WAF server and collects in the database server. The main steps of the proposed model are illustrated in Figure 2. To detect malicious requests based on machine learning is necessary to construct features from HTTP logs. For this purpose, n-gram character-based model is applied [22]. The n-gram model is a subsequence of n overlapping character from a given data. The j^{th} feature of the i^{th} feature vector x_{ij} is equal to the frequency of occurrences of the j^{th} n-gram in the i^{th} request.

With growth n in n-gram, the size of features grows exponentially and the problem becomes ill-posed. To overcome this problem, we are applied autoencoder for extract attributes which are complex abstract from data. For possess a hierarchical learning process, deep learning algorithms can be used in stacked autoencoder (SAE) manner. Deep learning models are able to learn good semantic attributes.

To detect anomalies, isolation forest model is applied in highest level output representation of SAE. This model partition data using random tree until all instances are isolated from each other [23].

III. Results

We implemented n-gram model for the construction of features and SAE algorithm for feature extraction and isolation forest to detect of malicious requests with following characteristics:

(1) We conducted experiments on the CSIC 2010 dataset [24]. The training set contains 28000 normal data and test set which contains 28000 normal and 15000 abnormal data.
(2) The proposed method builds a model for only normal data as in one-class classifiers i.e. isolation forest.
(3) Only 80 code of ASCII code appears in CSIC dataset. Therefore, the 1-gram model is computed as the frequency of occurrence each ASCII code (80 characters) in the dataset.
(4) As a consequence, each HTTP request in 1-gram model is represented by a vector of 80 dimensions, while bigrams feature vectors have 3603 dimensions. In order to reduce irrelevant features, we filtered out most frequent items in the dataset, thus, which according to our experiments, each HTTP request in 2-grams are represented by a vector of 2572 features.
(5) The structure of SAE for 2-gram models are 1000, 400 and 100 hidden neurons respectively for each layer. Also, the loss function of SAE is quadratic.

Additionally, we examined experiments on a system with Intel Core(TM), 2.67 GHz (2 Processors), 16 GB RAM and Windows 7 (64-bit) OS characteristics. Also, Algorithms implemented in Python 3.6 programming language and Tensorflow library [25]. We used several measures namely accuracy, detection rate, precision, specificity and F1-value to evaluate the performance of methods. These measures demonstrate in Table I.

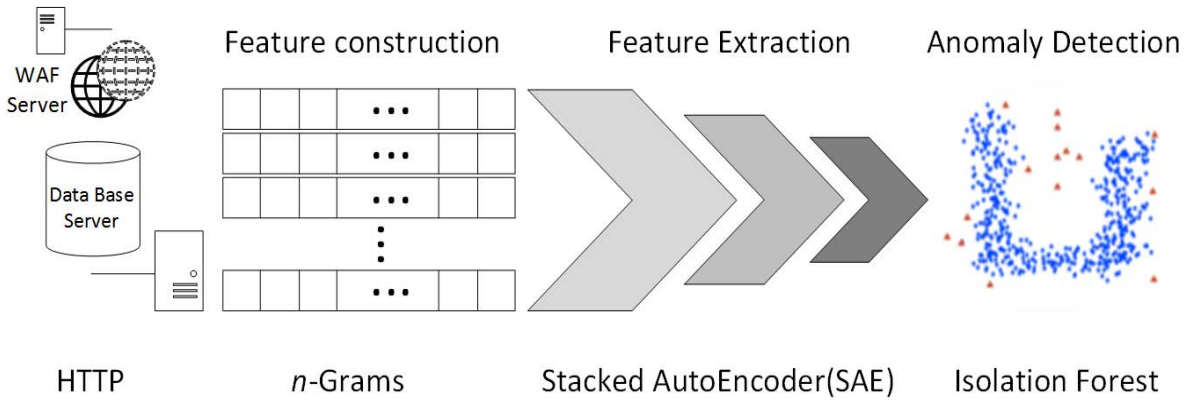

Fig. 2. Scheme of proposed anomaly detection method for WAF

TABLE I
THE FORMULATE OF DETECTION MEASURES. EACH VALUE HAS BEEN MULTIPLIED BY 100

Measure	Accuracy	Detection Rate	Precision	Specificity	F-score
Abbreviation	Acc	DR	PR	Spec	F1
Formulate	$\frac{TP^{*}+TN^{\S}}{TP+TN+FP^{\dagger}+FN^{\ddagger}}$	$\frac{TP}{TP+FN}$	$\frac{TP}{TP+FP}$	$\frac{TN}{TN+FP}$	$\frac{2*PR*DR}{PR+DR}$

* True Positive § True Negative †False Positive ‡ False Negative

TABLE II
THE RESULTS OF ANOMALY DETECTION OF VARIOUS MODELS

n-gram	Deep	Neuron	Training	Acc	DR	PR	Spec	F1
unigram	——			74.82	46.85	71.41	89.89	56.58
bigram	——			71.90	37.94	67.58	**90.20**	48.60
bigram	SAE	Sigmoid	SGD	78.17	56.32	75.10	89.94	64.37
bigram	SAE	Softplus	SGD	Non-feasible				
bigram	SAE	Sigmoid	Adam	**88.32**	**88.34**	80.29	88.32	**84.12**
bigram	SAE	Softplus	Adam	73.49	44.13	69	89.32	53.83
bigram	SAE	Sigmoid	Adagrad	84.87	75.93	79.88	89.69	77.85
bigram	SAE	Softplus	Adagrad	73.91	44.35	70.18	89.84	54.35
bigram	SAE	Sigmoid	RmsProp	88.12	86.70	**80.79**	88.89	83.64
bigram	SAE	Softplus	RmsProp	74.11	45.86	69.84	89.33	55.37

Table II presents the results of models which are implemented on without and with SAE. Although the increasing n in n-model should have the better estimate, the irrelevant features reduce the measures of the model. Table II shows the accuracy value of deep models especially the sigmoid activated function of neurons with keeping specificity is remarkable. Also, as we see in detection rate and specificity shows deep learning methods based on sigmoid have the good generalization. In addition, although the Adam optimization is better than others generally, for this problem, the selection of efficient neuron activation function is more important than optimization algorithms. Of course, it's better to use modified gradient descent instead of it.

CONCLUSION

In this study, we have leveraged SAE method to extract relevant features from HTTP request logs for anomaly detection in web applications firewalls. Isolation Forest has been used as a one-class learner to distinguish abnormal data from normal. Results show that deep models have the various performance with different structures of SAE and in general, deep models have better performance. For future work, an extension of the method for other deep learning methods and data streams can be considered.

REFERENCES

[1] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, Intrusion detection system: A comprehensive review, *Journal of Network and Computer Applications*, **36(1)**(2013), 16–24.

[2] C. Kruegel, and G. Vigna, A multi-mode approach to the detection of web-based attacks, *Computer Networks* , **48(5)**(2005), 717–738.

[3] K. L. Ingham, and H. Inoue, Comparing anomaly detection techniques for http, RAID. Vol 4637, (2007), 42–62.

[4] K. L. Ingham, A. Somayaji, J. Burge, and S. Forrest, Learning DFA representations of HTTP for protecting web applications, *Computer Networks*, **51(5)**(2007), 1239–1255.

[5] Y. Song, A. D. Keromytis, and S. J. Stolfo, Spectrogram: A Mixture-of-Markov-Chains Model for Anomaly Detection in Web Traffic, *NDSS*, **36(1)**(2009), 1–5.

[6] M. Zolotukhin, T. Hmlinen, T. Kokkonen, and J. Siltanen, Analysis of http requests for anomaly detection of web attacks, Autonomic and Secure Computing (DASC), (2014), 406–411.

[7] H. T. Nguyen, C. Torrano-Gimenez, G. Alvarez, K. Franke, and S. Petrovi, Enhancing the effectiveness of Web Application Firewalls by generic feature selection, *Logic Journal of IGPL*, **21(4)**(2012), 560–570.

[8] C. TorranoGimenez, H. T. Nguyen, G. Alvarez, and K. Franke, Combining expert knowledge with automatic feature extraction for reliable web attack detection, *Security and Communication Networks*, **8(16)**(2015), 2750–2767.

[9] C. Torrano-Gimenez, H. T. Nguyen, G. Alvarez, S. Petrovic, and K. Franke, Applying feature selection to payload-based web application firewalls, IEEE Security Communication Networks, (2011), 75–81.

[10] W. Wang, T. Guyet, R. Quiniou, M.-O. Cordier, F. Masseglia, and X. Zhang, Autonomic intrusion detection: Adaptively detecting anomalies over unlabeled audit data streams in computer networks, *Knowledge-Based Systems*, **70**(2014), 103–117.

[11] W. Wang, and X. Zhang, High-speed web attack detection through extracting exemplars from HTTP traffic, Proceedings of the 2011 ACM Symposium on Applied Computing. ACM, (2011), 1538-1543.

[12] J. J. Stephan, S. D. Mohammed, and M. K. Abbas, Neural Network Approach to Web Application Protection, *International Journal of Information and Education Technology*, **5(2)**(2015), 150.

[13] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, Deep learning applications and challenges in big data analytics, *Journal of Big Data*, **5(2)**(2015), 1.

[14] Y. Bengio, A. Courville, and P. Vincent, Representation learning: A review and new perspectives, *IEEE transactions on pattern analysis and machine intelligence*, **35(8)**(2013), 1798–1828.

[15] T. Ma, F. Wang, J. Cheng, Y. Yu, and X. Chen, A Hybrid Spectral Clustering and Deep Neural Network Ensemble Algorithm for Intrusion Detection in Sensor Networks, *Sensors*, **16(10)**(2016), 1701.

[16] M. Zhang, B. Xu, S. Bai, S. Lu, and Z. Lin, A Deep Learning Method to Detect Web Attacks Using a Specially Designed CNN, Springer International Publishing AG, (2017), 828–836.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning. Book in preparation for MIT Press*, MIT Press,deeplearning book, (2016).

[18] M. Harmon, and D. Klabjan, Activation Ensembles for Deep Neural Networks, *arXiv preprint arXiv*, **()**(2017), 1702.07790.

[19] J. Duchi, E. Hazan, and Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *Journal of Machine Learning Research*, **12(Jul)**(2011), 2121–2159.

[20] T. Tieleman, and G. Hinton, Lecture 6.5-RMSProp, COURSERA: Neural networks for machine learning, University of Toronto, (2012), Tech. Rep.

[21] D. Kingma, and J. Ba, Adam: A method for stochastic optimization, *arXiv preprint arXiv*, (2014), 1412.6980.

[22] D. Jurafsky, and J. H. Martin, *Speech and language processing*, Pearson London, (2014).

[23] F. T. Liu, K. M. Ting, and Z.-H. Zhou, Isolation forest, Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on. IEEE, (2008), 413–422.

[24] R. Jensen, S. Vluymans, N. Mac Parthaláin, C. Cornelis and Y. Saeys, The HTTP dataset CSIC 2010, Instituto de Seguridad de la Informacin (ISI), (2010).

[25] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, and M. Devin, Tensorflow: Large-scale machine learning on heterogeneous distributed systems, *arXiv preprint arXiv*, (2016), 1603.04467.