Deep Learning-based Feature Selection for Intrusion Detection System in Transport Layer [1)]

Muhamad Erza Aminanto* and Kwangjo Kim*

*School of Computing, KAIST

Abstract

Numerous machine learning algorithms applied on Intrusion Detection System (IDS) to detect enormous attacks. However, it is difficult for machine to learn attack properties globally since there are huge and complex input features. Feature selection can overcome this problem by selecting the most important features only to reduce the dimensionality of input features. We leverage Artificial Neural Network (ANN) for the feature selection. In addition, in order to be suitable for resource-constrained devices, we can divide the IDS into smaller parts based on TCP/IP layer since different layer has specific attack types. We show the IDS for transport layer only as a prove of concept. We apply Stacked Auto Encoder (SAE) which belongs to deep learning algorithm as a classifier for KDD99 Dataset. Our experiment shows that the reduced input features are sufficient for classification task.

I. Introduction

Lots of the distinct attacks within the computer networks appear every day. One feasible countermeasure against them is to apply Intrusion Detection System (IDS) on our computer network to detect the unauthorized traffic efficiently. Major improvements of the previous IDSs can be achieved by leveraging the latest breakthrough machine learning methods[1], so called *deep learning*.

However, the computations for most deep learning are heavy. Internet of Things (IoT) era is coming soon which employs tiny devices with limited computing power. Reducing the dimensionality of input features is one of candidate solutions. According to previous work [2], each TCP/IP layer has its own attack characteristics. Each network devices belongs to different and specific TCP/IP layer, such as router for network layer, switch for transport layer and server for application layer. Since each layer has its own attack characteristic and different hardware, we must consider the different IDS on each TCP/IP layer. This method has two advantages. First, the IDS algorithm on each layer are lightweight which will be appropriate for resource-constrained devices. Second, the detection performance should be comparable as good as complete IDS since it focuses on specific attack characteristic only. We leverage Artificial Neural Network (ANN) for the feature selection. The weight from trained models mimics the importance of correspondence input. By selecting the important features only, the training process becomes lighter and faster than before.

To suggest the prove of concept (PoC) of our approach, we use Stacked Auto Encoder (SAE) as a classifier which is one of popular deep learning algorithms since this employs consecutive layers of processing stages in

1) This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (B0101-16-1270, Research on Communication Technology using Bio-Inspired Algorithm) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2015R1A2A2A01006812).

hierarchical manners for pattern classification and feature or representation learning [3]. We use KDD 99 Dataset [4] in order to show the concept of reduced input features by ANN. Our experiment shows that the reduced input features are sufficient for SAE algorithm to achieve comparable detection rate as the whole features.

This paper is organized as follows: Section 2 reviews a number of related work. We describe our proposed scheme in Section 3. Section 4 reports our experiment results and analysis. Conclusion and future work of this paper will be made in Section 5.

II. Related Work

The importance of feature selection for IDS dataset introduced by Kayacik *et al.* [5]. They investigated the relevance of each feature in KDD 99 Dataset. They provided the information gain for each feature. Their conclusion ends with the list of the most relevant features for each class label. In this paper, we find the most relevant features for corresponding TCP/IP layer instead of each class label on KDD 99 Dataset. This method introduced by Zaman and Karray [6] who categorized the IDS based on the TCP/IP network model. Each of those IDS types is specialized to a specific network device. Thus, the detection process is distributed through all TCP/IP layers. Zaman and Karray [6] use Enhanced Support Vector Fecidion Function (ESVDF) and Support Vector Machine (SVM) while we use ANN and SAE for the feature selection and classification algorithms respectively. We leverage recent deep learning algorithm as published by Wang [7]. In 2015, Wang has shown that neural networks especially the deep neural networks can be used for finding features in the raw network flow data. We use KDD 99 Dataset for experiment.

III. Our Proposed Scheme

In this section, we briefly describe our IDS scheme. We divide one big IDS into four smaller parts based on TCP/IP network model. Fig.1 shows the proposed architecture. The proposed approach shows four different IDS: IDS-A for application layer, IDS-T for transport layer, IDS-N for network layer and IDS-L for data link layer. Each IDS type is responsible for different network devices which are distributed among computer networks.

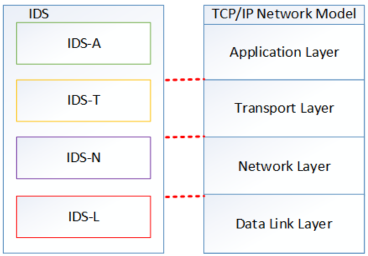


Fig.1. Proposed architecture

In order to design the four different types of IDSs, we may use any feature selection approach to choose the appropriate feature set for each layer in TCP/IP network. There are two steps for feature selection process. First, we prepare different datasets for training and testing purposes. Each IDS type has its own dataset based on TCP/IP layer properties. As an example, dataset for IDS-A contains data instances with normal and application layer attacks label. Second, we apply the feature selection method for each dataset to choose the most important feature set for each IDS type. However, we limit the IDS-T only as a PoC. IDS-N, IDS-A and IDS-L are not discussed in this paper.

We employ ANN as feature selection method. Fig.2 shows the ANN model which b1 and b2 represents bias value for the corresponding hidden layer.

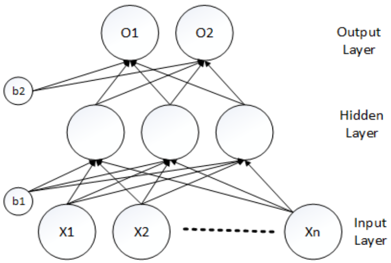


Fig.2 ANN model

In this paper, we use one hidden layer only for feature selection. We exploit the weight value between first two layers for the consideration to choose the important input features. The weight W_{ij} represents contribution of the input features x_j to the first hidden layer features h_i. If W_{ij} is very small or zero means that the corresponding input feature x_j is meaningless for further propagation. So, one hidden layer is sufficient since we consider weights in the first hidden layer only. We define the importance value of each input feature, V_j as expressed by Eq. 1.

$$V_j = \sum_{i=1}^{h} |W_{ij}|, \tag{1}$$

where h is the number of neurons in the first hidden layer. In order to select the most important features, we sort the input features by V_j in descending order. We pick some features that have V_j bigger than a threshold value.

In order to show the performance of each type of IDSs, we utilize deep learning algorithm, SAE, using the selected features as a classifier. Basically, Auto Encoder (AE) model is similar to ANN as shown in Fig.3.

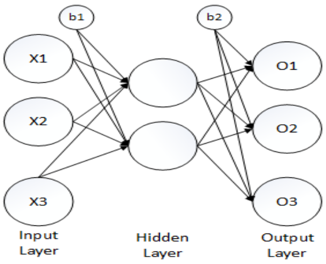


Fig.3 AE model

The main difference is the number of nodes in the input layer are equal with the number of nodes in the output layer. Meanwhile, the nodes in the hidden middle layer represent new features set with lower dimension. This architecture leads to an ability that can reconstruct the data after complicated computations. AE aims to learn a compact set of data efficiently whose can be stacked to build deep networks. Each training results of the middle layer can be cascaded. This structure is called SAE which can learn lots of new features in different depths [7]. Fig.4 shows the proposed SAE architecture used in this paper.

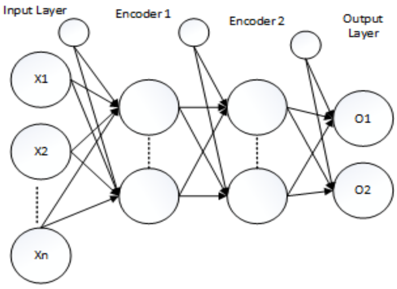


Fig.4 Proposed SAE architecture

We employ two hidden (encoder) layers. The features that were generated from the first encoder layer used as the training data in the second encoder layer. Meanwhile, the size of each hidden representation is decreased accordingly, so that the encoder in the second encoder layer learns an even smaller representation of the input data. We complete our stacked architecture with supervised learning approach by *softmax* regression function using labels from training data.

IV. Result

We use KDD 99 Dataset for evaluating the importance of feature selection on each IDSs. we evaluate our proposed scheme with IDS-T type only since the rest of IDS types should have similar results. According to

Zaman and Karray [2], there are six attack types on KDD 99 Dataset that belong to Transport layer: *land, neptune, teardrop, buffer_overflow, portsweep*, and *nmap*. The features that learned from these type of attacks are useful for IDS-T training. We deploy an experiment environments: MATLAB R2016a that runs in Intel(R) Xeon(R) CPU E-3-1230v3@3.30 Ghz, RAM 32 GB. Fig.5 shows weight of each input feature for IDS-T.

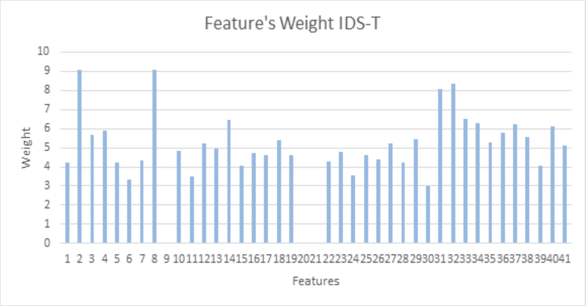
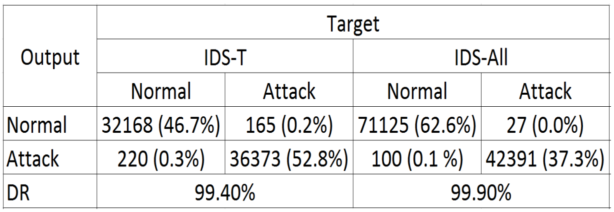


Fig.5 Weight of each input feature for IDS-T

In this case, we set the threshold value to 6. According to Fig.5, the list of important features for IDS-T are: 2, 8, 14, 31, 32, 33, 34, 37 and 40. We use this list of features as an input for classification task.

Table 1 Confusion Matrix for IDS-T and IDS-All



Output	Target			
	IDS-T		IDS-All	
	Normal	Attack	Normal	Attack
Normal	32168 (46.7%)	165 (0.2%)	71125 (62.6%)	27 (0.0%)
Attack	220 (0.3%)	36373 (52.8%)	100 (0.1 %)	42391 (37.3%)
DR	99.40%		99.90%	

Table 1 shows the confusion matrix for IDS-T and IDS-All. The table contains comparison between target class and classifying output. DR stands for Detection Rate. IDS-All trained with original input features. IDS-All still outperform IDS-T. However, the detection rate is slightly different, but training time consumed for IDS-All is much longer than IDS-T. The IDS-All took almost 10 minute while IDS-T took one minute only for classification task.

V. Conclusion and Future Work

We leverage ANN as feature selection and SAE as classifier. Our experiment result shows that lightweight IDS can be achieved by dividing IDS into smaller parts and reduce feature dimensionality. The lightweight IDS achieve comparable detection rate as the ordinary IDS. In the near future, we will complete our experiments for all types of IDSs. In addition, implementing lightweight IDS for a wireless network is a challenging issue.

References

[1] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection." Security and Privacy, 2010 IEEE Symposium on - S&P 2010, IEEE.

[2] S. Zaman and F. Karray, "TCP/IP model and intrusion detection systems." Advanced Information Networking and Applications Workshops, 2009, International Conference on - WAINA 2009, IEEE.

[3] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning." Transactions on Signal and Information Processing 3: e2, Cambridge Univ Press, 2014.

[4] S. Hettich and S.D. Bay, "The uci kdd archive [http://kdd. ics. uci. edu]. Irvine, CA: University of California." Department of Information and Computer Science: 152, University of California, 1999.

[5] H. Kayacik, A.N. Zincir-Heywood, and M.I. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets." Proceedings of the third annual conference on privacy, security and trust 2005, PST 2005, DBLP.

[6] S. Zaman and F. Karray, "Lightweight IDS based on features selection and IDS classification scheme." Computational Science and Engineering, 2009. International Conference on. Vol. 3. - CSE 2009, IEEE.

[7] Z. Wang, "The application of deep learning on traffic ddentification," BlackHat USA, 2015.