Detecting Active Attacks in WiFi Network by Semi-supervised Deep Learning[1]

Muhamad Erza Aminanto* and Kwangjo Kim*

*School of Computing, KAIST

Abstract

WiFi network usage is increased rapidly these days while the number of attacks in WiFi network are growing as well. Intrusion Detection System (IDS) is one of the popular defense mechanisms that often uses *e.g.,* machine learning algorithms in order to detect both known and unknown attacks in a particular network. We leverage an unsupervised deep learning approach, so called Stacked Auto Encoder (SAE) as feature extraction scheme. Feature extraction by SAE can reduce the complexity of original features of the dataset. While regression layer with softmax activation function is implemented as supervised classification. In this paper, we test our proposed IDS using AWID dataset which is one of comprehensive WiFi network traces from real network. Our experiments show that our proposed IDS can outperform the previous work by Kolias *et al.* In addition, we provide several suggestions in order to made our proposed IDS reach an optimum result.

I. Introduction

WiFi network traffics are expected to increase rapidly due to the fact that WiFi network is a common network for tiny devices spread anywhere as Internet of Things (IoT) become more popular these days [6]. Unfortunately, vulnerabilities and attacks for WiFi networks are growing exponentially as a result [6]. Impersonation, flooding and injection attacks are popular examples of WiFi network attacks. An Intrusion Detection System (IDS) leveraging machine learning can be a great detector of these attacks.

In this paper, we use semi-supervised approach for our IDS which contains feature extractor (unsupervised learning) and classifier (supervised learning). We leverage Stacked Auto Encoder (SAE) for feature extraction, and regression layer with softmax activation function for classifier. Our experiments show that we can achieve higher WiFi network attack detection accuracy rate compared to Kolias *et al* [4] and our previous work [5]. But, there is a limitation from our proposed approach. We suggest two methods to overcome the limitation.

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

In this paper, we use AWID dataset by Kolias *et al* [4], which was verified by various machine learning algorithms in a heuristic manner. Unfortunately, classification result for two attack classes are insufficient.

[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).

We overcome this problem in our previous work [5] using deep learning approach, which are Artificial Neural Network (ANN) and SAE. However, we dealt with impersonation attack only in [5]. Therefore, in this paper, we focus on achieving higher accuracy rate for active attacks, not impersonation attack only.

III. Our Proposed Scheme

We first explain some preliminaries. We use feature extraction, which is different from feature selection. Feature extraction outputs new transformed generated features which are completely in a different form with the original features, while the feature selection involves selecting a subset of the original features only. There are several algorithms that can be used for the feature extraction, such as Auto Encoder (AE), SAE, and Convolutional Neural Network (CNN) [1]. Basically, AE is an Artificial Neural Network (ANN) with special characteristic, implying that the number of the input layer nodes are the same as the output layer nodes. Meanwhile, the nodes in middle hidden layer are able to express new lower-dimensional features set. The AE architecture as shown in Fig.1, leads to the ability that can reconstruct the data after complicated computations.

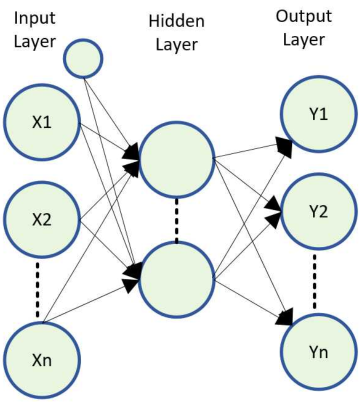


Fig.1. AE architecture

The AE can be stacked to build deep networks. Lower-dimensional features coming from each training results are cascaded, so called SAE [2], that can learn a lot of new features in different depths. Fig.2 shows our proposed SAE network which is one example of SAE architectures.

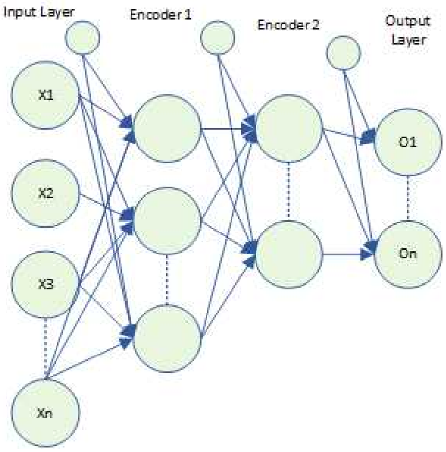


Fig.2. Our proposed SAE network

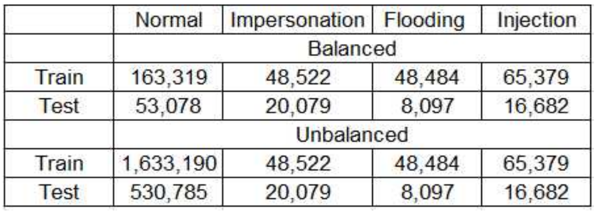
In addition, Denoising Auto Encoder (DAE) is trained to reconstruct a clear correction input from a corrupted by noise input [3]. DAE may also be stacked in order to build deep networks as well.

In this paper, we adopt an IDS architecture that consists of two main building blocks, namely feature extraction and classification part. We use SAE as our feature extraction, which is an unsupervised learning, with two and three hidden layers. The features that were generated from the first encoder layer are 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. Then, regression layer with softmax activation function is implemented as our classification part, which is classified as supervised learning.

IV. Experiment and Result

We use AWID dataset, which is the real WiFi network traces [4], for evaluating our proposed IDS. The dataset contains three type of active attacks: impersonation, flooding and injection attacks. Data preprocessing should be done in advance, since the dataset does not only contain discrete data type, but also continuous and symbolic data types. Besides, the dataset distribution is unbalanced since it was real network traces. Therefore, we prepare the balanced dataset in order to optimize training task. Table 1 shows the distribution of each classes in balanced and unbalanced AWID datasets.

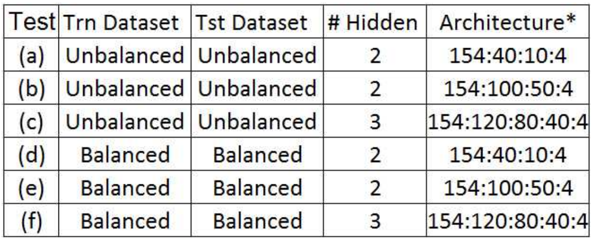
Table 1 Distribution of each classes in balanced and unbalanced AWID datasets



	Normal	Impersonation	Flooding	Injection
	Balanced			
Train	163,319	48,522	48,484	65,379
Test	53,078	20,079	8,097	16,682
	Unbalanced			
Train	1,633,190	48,522	48,484	65,379
Test	530,785	20,079	8,097	16,682

We examined 6 Tests in order to learn the SAE learning task. Table 2 shows the summary of our 6 Tests.

Table 2 Summary of our 6 Tests



Test	Trn Dataset	Tst Dataset	# Hidden	Architecture*
(a)	Unbalanced	Unbalanced	2	154:40:10:4
(b)	Unbalanced	Unbalanced	2	154:100:50:4
(c)	Unbalanced	Unbalanced	3	154:120:80:40:4
(d)	Balanced	Balanced	2	154:40:10:4
(e)	Balanced	Balanced	2	154:100:50:4
(f)	Balanced	Balanced	3	154:120:80:40:4

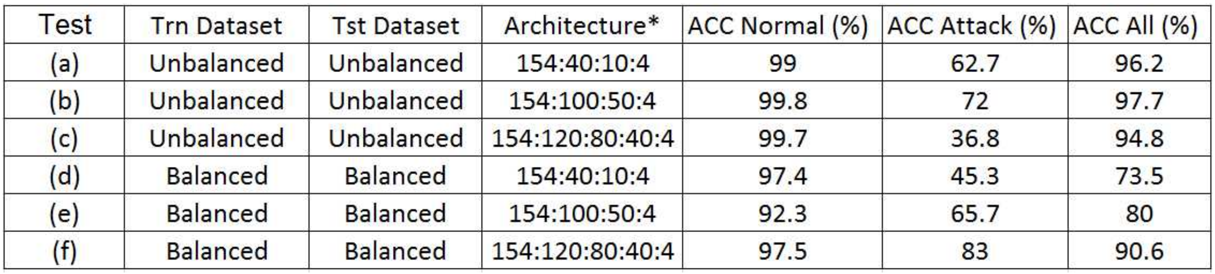
We selected our Test schemes due to the following reasons:

- Balanced dataset supposed to be made the training results better since sufficient data are provided.
- The more hidden layers are used, the more complexity of features learned.
- Reducing one fourth ratio between hidden layers is believed to give optimized SAE learning result. However, our experiment results show against it.

For experiment setup, we use: MATLAB R2016a which runs in Intel(R) Xeon(R) CPU E-3-1230v3@3.30 GHz, RAM 32GB. We define the accuracy metric, which is the number of correctly classified data instances divided by the total number of that instances' class, as our performance evaluation. We separate the accuracy between normal and attack classes due to unbalanced dataset. Therefore, we can make fair performance comparison.

Table 3 shows the performance comparison for all Tests. Tests (a) to (c) use unbalanced dataset, while Tests (d) to (f) use balanced dataset. If we look at the accuracy for all (ACC All) only, we might select Test (b) as the best result due to the highest accuracy rate. However, Test (b) has high accuracy rate because of unbalanced data between

Table 3 Performance comparison for all Tests



Test	Trn Dataset	Tst Dataset	Architecture*	ACC Normal (%)	ACC Attack (%)	ACC All (%)
(a)	Unbalanced	Unbalanced	154:40:10:4	99	62.7	96.2
(b)	Unbalanced	Unbalanced	154:100:50:4	99.8	72	97.7
(c)	Unbalanced	Unbalanced	154:120:80:40:4	99.7	36.8	94.8
(d)	Balanced	Balanced	154:40:10:4	97.4	45.3	73.5
(e)	Balanced	Balanced	154:100:50:4	92.3	65.7	80
(f)	Balanced	Balanced	154:120:80:40:4	97.5	83	90.6

*) The entry means the number of Input Layer : Encoder1 : Encoder2 : Output Layer.

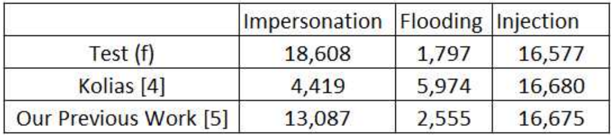
normal and attack instances. We can consider Test (f) as the optimum learning result as shown in Table 3, because it has the highest accuracy rate for attack instances and considerably high accuracy rate for normal instances.

We can observe that the accuracy of normal instances is really high when we use unbalanced dataset, because of an excessive number of data instances that belong to the normal class. Unfortunately, because of the same reason, the accuracy rate for attacks is lower than the training using balanced dataset. The results in Table 3 also support our claim that the more hidden layers are used, the higher accuracy rate because more complex features can be learned.

We also observed that achieving high accuracy rate for all attack classes in the same time seems difficult. We achieve a high accuracy rate for one particular class, but low accuracy rate for other classes.

In this paper, we also compare our proposed IDS scheme with two related previous work, Kolias *et al* [4] and our previous work [5]. Table 4 shows the performance comparison.

Table 4 Performance comparison



	Impersonation	Flooding	Injection
Test (f)	18,608	1,797	16,577
Kolias [4]	4,419	5,974	16,680
Our Previous Work [5]	13,087	2,555	16,675

From Table 4, we can see that we achieved really high accuracy rate for impersonation and injection attacks. Unfortunately, we got the lowest accuracy for Flooding attack due to the trade-off between each classes as mentioned before.

In order to overcome the limitation, we leave the following tasks to do:

- Bagging, also called bootstrap aggregating, is a method to ensemble several training with customized training data in order to achieve optimum result for *k*-ary classification.
- Dropout is a method to train many times with incomplete features, so that the model are better learned.

V. Conclusion and Future Work

We leverage SAE for feature extraction and regression layer with softmax activation function for classifier. Our experiment results show that our semi-supervised IDS approach can achieve higher WiFi network attack accuracy rate compared to the previous work. However, still there are some limitations, trade-off on each class issue. Bagging and dropout method may overcome this limitation as we plan for further research.

References

[1] Pigou, Lionel, *et al.*, "Sign language recognition using convolutional neural networks," Workshop at the European Conference on Computer Vision. Springer International Publishing, pp: 572-578, 2014.

[2] Bengio, Yoshua, *et al.*, "Greedy layer-wise training of deep networks," Advances in neural information processing systems, vol. 19, pp: 153-160, 2007.

[3] Vincent, Pascal, *et al.*, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," Journal of Machine Learning Research, vol. 11, pp: 3371-3408, 2010.

[4] Kolias, Constantinos, *et al.*, "Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset," IEEE Communications Surveys & Tutorials, vol:18.1, pp: 184-208, 2015.

[5] Aminanto, Muhamad Erza and Kim, Kwangjo, "Detecting Impersonation Attack in WiFi Networks Using Deep Learning Approach," Post-Proc. of WISA 2016, Jeju, Korea (To appear in LNCS by Springer), 2016.

[6] Kolias, Constantinos, *et al.*, "Learning Internet-of-Things Security Hands-On," IEEE Security & Privacy, vol: 14.1, pp: 37-46, 2016.