Improving Detection of Wi-Fi Impersonation by Fully Unsupervised Deep Learning

Muhamad Erza Aminanto and Kwangjo Kim

Cryptology and Information Security Lab, School of Computing,
Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, Republic of Korea
{aminanto,kkj}@kaist.ac.kr

Abstract. Intrusion Detection System (IDS) has been becoming a vital measure in any networks, especially Wi-Fi networks. Wi-Fi networks growth is undeniable due to a huge amount of tiny devices connected via Wi-Fi networks. Regrettably, adversaries may take advantage by launching an impersonation attack, a common wireless network attack. Any IDS usually depends on classification capabilities of machine learning, which supervised learning approaches give the best performance to distinguish benign and malicious data. However, due to massive traffic, it is difficult to collect labeled data in Wi-Fi networks. Therefore, we propose a novel fully unsupervised method which can detect attacks without prior information on data label. Our method is equipped by an unsupervised stacked autoencoder for extracting features and a k-means clustering algorithm for clustering task. We validate our method using a comprehensive Wi-Fi network dataset, Aegean Wi-Fi Intrusion Dataset (AWID). Our experiments show that by using fully unsupervised approach, our method is able to classify impersonation attack in Wi-Fi networks with 92% detection rate without any label needed during training.

1 Introduction

The experts have already anticipated the growth of wireless network traffics [1]. Mobile traffics, including mobile 5G and Wi-Fi traffic are believed to increase tremendously, in the next 10 years [1]. Unfortunately, as the traffic increases, a number of malicious attacks by adversaries are have jumped accordingly [2]. An impersonation attack is one of common Wi-Fi attacks [3]. In this attack, adversaries can impersonate themselves as legitimate clients to gain unauthorized access. The impersonation attack also has a severe impact due to allowing unauthorized users to access the network as a security breach [4].

Intrusion Detection System (IDS) become a promising countermeasure for network attacks by leveraging machine learning application occasionally. Machine learning, based on data label availability, can be divided into two approaches: supervised and unsupervised learning. A supervised learning needs prior information about the class label data. The supervised learning fits for the classification task, including attack detection. In the latter case, an unsupervised learning learns without any prior information about the class label of raw data. Therefore, the unsupervised learning fits for clustering task, which makes an efficient way to group similar data. In terms of attack detection, we may leverage unsupervised approach by claiming the outlier data from big clusters as attacks, since benign data usually form a big cluster. Besides that, unsupervised learning is suitable for huge Wi-Fi networks as labeling training data may be infeasible.

There are numerous famous unsupervised learning methods such as k-means clustering [5], Principal Component Analysis (PCA) [6] and Independent Component Analysis (ICA) [7]. The key characteristics of the three methods are: k number of class partitioning, orthogonal transformation and reveal hidden independent factors, respectively [8]. However, since we are facing huge and complex Wi-Fi data, the three traditional unsupervised learning methods are insufficient because the data might be not well distributed [9]. In order to overcome this problem, we venture to transform raw data into another form of data, which can lead to better unsupervised learning result.

One acceptable candidate for the transformations is Stacked Autoencoder (SAE) which transforms original features into more meaningful representation by reconstructing its input with the decoder. It provides an efficient way to validate that the important information in the data has been captured. The SAE as a deep learning method, can be efficiently used for unsupervised learning on a complex dataset. By stacking several unsupervised feature learning layers, and greedy method training for each layer, we can consider extracted features on each hidden layer as a new space with better form for clustering task. However, SAE is originally designed for capturing complex information in lower-dimensional features than the original features, not for clustering task. Therefore, we see SAE for assisting traditional clustering algorithm to achieve better clustering result. We then forward the newly formed features from non-linear SAE transformation into k-means clustering algorithm to improve k-means clustering performance.

We implement and test our work using a comprehensive Wi-Fi network benchmark dataset, called AWID dataset [3]. Besides this dataset, Kolias *et al.* [3] also tested a series of existing machine learning models on the dataset in a heuristic manner. The lowest detection rate is observed on impersonation attack by detection rate of 22% only while our proposed approach outperforms on impersonation attack detection achieving a detection rate of 92%. Clearly, the novel way of combining deep learning transformation and traditional k-means clustering method improves the performance of impersonation attack detector and can be further generalized for different attack types in large scale Wi-Fi networks.

This paper is organized as follows: Section 2 reviews several related work. We provide our proposed approach along preliminaries in Section 3. Section 4 gives our experimental results and analysis. Conclusion and future work of this paper will be suggested in Section 5.

2 Related Work

There are several previous work which leverages deep learning techniques as a clustering method. Song *et al.* [9] proposed an autoencoder-based data clustering. They mapped original data space to a new space using autoencoder, which is more suitable for clustering, and claimed that by applying a non-linear transformation, the data become compact with respect to their corresponding cluster center in the new space. They modified original autoencoder by adding two new objective functions during training: minimize reconstruction error and distance. Comparing with Song *et al.* [9] which needs to modify the original autoencoder, our proposed method does not need to modify the original autoencoder and improves traditional clustering algorithm. While Saito *et al.* [10] proposed similar work by mapping the input data to an embedded space, using autoencoder. They concatenated the learned representations of all intermediate layers. All features learned by each neural network layer were used to generate a combined representation which is useful for effective cluster analysis. Different from Saito *et al.* [10], we leverage unsupervised k-means clustering algorithm instead of supervised k-Nearest Neighbor (k-NN) classification algorithm. We also use a single hidden layer only without concatenation of all hidden layers to maintain the process still lightweight.

A lot of proposed approaches for detecting impersonation attacks [11], [12], [13] and [14]. [11], [12] and [13] were designed to detect one particular impersonation attack by adding or modifying specific protocols. However, in particular, Aminanto and Kim [14] proposed one general model that can detect an impersonation attack by reducing the features dimensionalities and adopting SAE at final stage. Unfortunately, their approach needs data labels which is supervised learning algorithm. Therefore, we propose a fully unsupervised approach for coping with huge and complex Wi-Fi network traffics.

3 Our Approach

In this Section, we briefly describe our novel fully unsupervised deep learning-based Wi-Fi impersonation attack detector. For clarity, we firstly introduce preliminaries about SAE and k-means clustering, and then explain how the overall scheme works for detecting attacks.

3.1 Stacked Autoencoder (SAE)

An autoencoder is a symmetric neural network model as shown in Fig. 1, which belongs to unsupervised learning in the sense that a model could be built from non-labeled data. To extract new-lower dimensional features, autoencoder uses an encoder-decoder paradigm as shown in Fig. 1, which can capture relevant data from the original data. The encoder is a function that maps an input X to a representation layer H as expressed by Eq. (1).

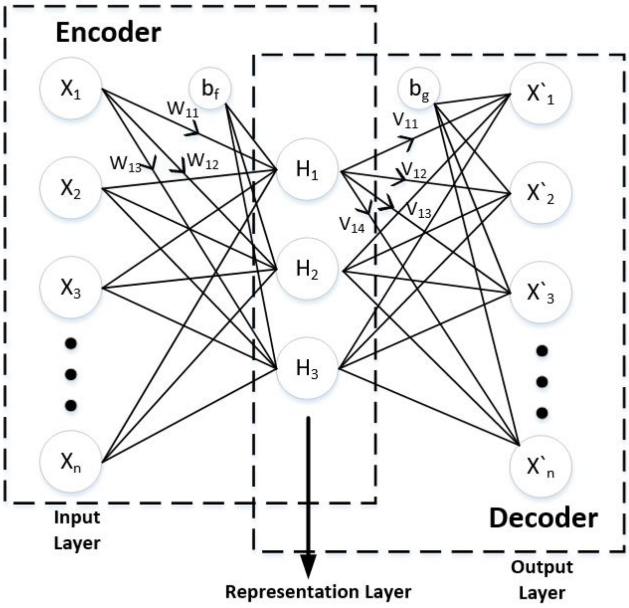

Fig. 1. Autoencoder network with symmetric input-output layers

$$H = s_f\left(WX + b_f\right) \tag{1}$$

, where s_f is a non-linear activation function, in this case of a logistic sigmoid, $s_f(t) = \dfrac{1}{1+e^{-t}}$, where t is the function input. The W and b_f denote a weight matrix for features and a bias vector for encoding, respectively. The decoder function expressed in Eq. (2) maps representation layer H back to a reconstruction X' as an output.

$$X' = s_g\left(VH + b_g\right) \tag{2}$$

, where s_g is the activation function of the decoder, which is a sigmoid function too. The V and b_g denote a weight matrix for features and a bias vector for decoding, respectively. Autoencoder training phase finds optimal parameters $\theta = \{W, V, b_f, b_g\}$ which minimize the reconstruction error E between the input data X and its reconstruction output X' on a training set as shown in Eq. (3).

$$E = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}\left(X'_{kn} - X_{kn}\right)^2 + \lambda \cdot \Omega_{weights} + \beta \cdot \Omega_{sparsity} \tag{3}$$

, where N and K denote the total number of training data and the number of variables for each data, respectively. $\Omega_{weights}$ represents L_2 regularization, while $\Omega_{sparsity}$ denotes sparsity regularization, which evaluates how close the average output activation value and the desired value. The coefficient of L_2 regularization term λ and the coefficient of sparsity regularization term β are specified during autoencoder training.

Autoencoder can be used as a deep learning technique by unsupervised greedy layer wise pre-training algorithm as depicted in Fig. 2, which is called Stacked Autoencoder (SAE). In this algorithm, all layers except the last layer are initialized in a multi-layer neural network. Each layer is then trained in an unsupervised manner as autoencoder which constructs new representations of the input.

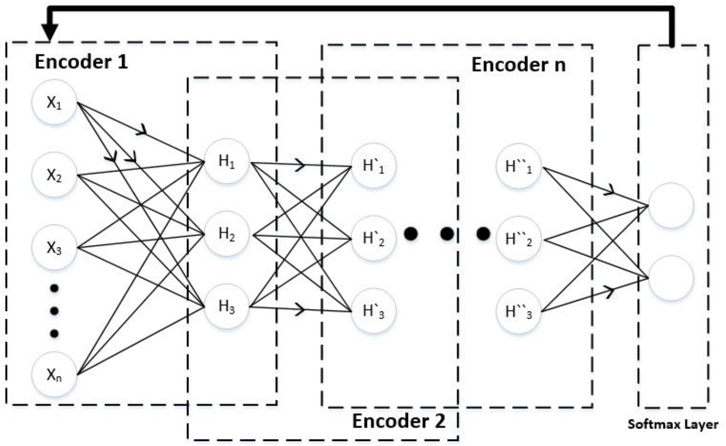

Fig. 2. Stacked Autoencoder (SAE) network with three hidden layers

The final layer implements the softmax for the classification the deep neural network. Softmax function is a generalized term of the logistic function that squashes the K-dimensional vector $\mathbf{v} \in \mathbb{R}^K$ into K-dimensional vector $\mathbf{v}^* \in (0,1)^K$ which adds up to 1. The softmax layer minimizes the loss function, which is the cross entropy function.

3.2 *K*-means Clustering

K-means clustering algorithm groups all observations data into k clusters iteratively until convergence will be reached. In the end, one cluster contains similar data since each data enters to the nearest cluster. K-means algorithm assigns a mean value of the cluster members as a cluster centroid. In every iteration, it calculates the shortest Euclidean distance from an observation data into any cluster centroid. Besides that, the intra-variances inside the cluster are also minimized by updating the cluster centroid iteratively. The algorithm would terminate when convergence is achieved, which the recent clusters are the same as the previous iteration clusters [8].

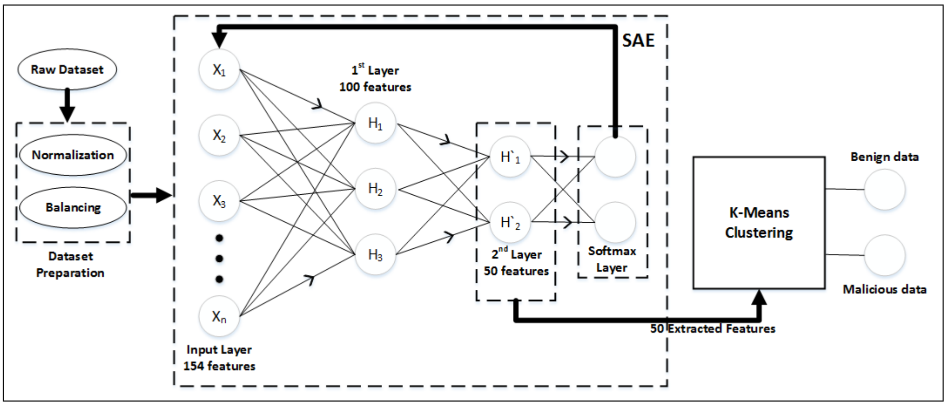

Fig. 3. Our proposed scheme contains feature extraction and clustering tasks

Algorithm 1 Pseudocode of Fully Unsupervised Deep Learning

```
procedure START
    function DATASET PREPARATION(Raw Dataset)
        for each data instance do
            Convert into integer value
            Normalization $z_i = \frac{x_i - min(x)}{max(x) - min(x)}$
        end for
        Balance the normalized dataset
        return $InputDataset$
    end function
    function SAE($InputDataset$)
        for $i$=1 to $h$ do                                  ▷ $h$=2; number of hidden layers
            for each data instance do
                Compute $H = s_f\left(WX + b_f\right)$
                Compute $X' = s_g\left(VH + b_g\right)$
                Minimize $E = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}\left(X'_{kn} - X_{kn}\right)^2 + \lambda \cdot \Omega_{weights} + \beta \cdot \Omega_{sparsity}$
                $\theta_i = \{W_i, V_i, b_{f_i}, b_{g_i}\}$
            end for
            $InputFeatures \leftarrow W_2$                    ▷ 2nd layer, 50 extracted features
        end for
        return $InputFeatures$
    end function
    Initialize clusters and k=2                          ▷ two clusters: benign and malicious
    function $k$-MEANS CLUSTERING($InputFeatures$)
        return $Clusters$
    end function
    Plot confusion between $Clusters$ and target classes
end procedure
```

3.3 Fully Unsupervised Deep Learning

In this subsection, we describe our novel fully unsupervised deep learning-based IDS for detecting impersonation attacks. There are two main tasks, feature extraction and clustering tasks. Fig.3 shows our proposed scheme which contains two main tasks in cascade. We use a real Wi-Fi networks-trace, AWID dataset [3], which contains 154 original features. Before the scheme starts, normalizing and balancing process should be done in order to achieve best training performance. Algorithm 1 explains the procedure of the proposed scheme in detail.

The scheme starts with two cascading encoders, and the output features from the second layer then forwarded to the clustering algorithm. The first encoder has 100 neurons as the first hidden layer while the second encoder comes with 50 neurons only. We follow a common rule for choosing the number of neurons in a hidden layer by using 70% to 90% of the previous layer. In this paper, we define k=2 since we consider two classes only. The scheme ends by two clusters formed by k-means clustering algorithm. These clusters represent benign and malicious data.

4 Evaluation

We evaluate the proposed scheme on AWID dataset. We firstly show the effectiveness of leveraging second layer representation of SAE training result compared to original data. We implement SAE and k-means clustering algorithm using MATLAB R2016b running on an Intel Xeon E-3-1230v3 CPU @3.30 GHz with 32 GB RAM. We verify our proposed scheme by comparing the proposed scheme with the previous work. We introduce the dataset has been used and evaluation metrics in the next subsections.

4.1 Dataset

We use AWID dataset as a benchmark dataset since the dataset might become a common benchmark dataset for wireless network research due to its comprehensiveness and real world-alike characteristics. Regarding the number of classes, the dataset has two types of attack classes: “ATK” and “CLS”. The “ATK” dataset consists of 16 classes including one benign class, while the “CLS” data contains four classes only. The 16 classes of the “ATK” dataset can be classified to four attack categories in the “CLS” dataset. In this paper, we use the “CLS” dataset which contains benign, impersonation, injection and flooding classes. However, we consider two classes only among four classes. Besides that, the AWID dataset is also divided into two types based on the size of data instances included, namely, full and reduced datasets. There are 1,795,595 data instances in the full dataset, with 1,633,190 and 162,385 benign and attack instances, respectively. While the reduced dataset contains only 575,643 instances, with 530,785 and 44,858 benign and attack instances, respectively. In this paper, we used the reduced “CLS” AWID dataset for the sake of simplicity.

The dataset expresses the nature of a network, where the number of benign instances is larger than attack instances [3]. The ratio between benign and attack instances are 10:1 and 11:1 for training and test datasets, respectively. This situation might cause a bias during training, and infer machine learning performance. Therefore, we balance the dataset for training purpose. The ratio between benign and attack instances then become 1:1 for both balanced training and test datasets. The benign instances are randomly reduced into 163,319 data instances for training dataset while 53,078 data instances for test dataset.

The AWID dataset not only consists of discrete data but also continuous and symbolic data types, with flexible value ranges. This situation might confuse any machine learning during training. The dataset preparation should be done in advance, which contains two main tasks: mapping symbolic-valued attributes to integer values and normalizing tasks. First, target classes would be mapped to integer type: 1 for benign instances and 2 for impersonation attack. Second, symbolic attributes, such as a receiver, destination, transmitter, and source address, would be mapped to integer values with a minimum value of 1 and a maximum value of i, where i is the number of all symbols. Third, some attributes that have a hexadecimal data type, such as WEP Initialization Vector (IV) and Integrity Check Value (ICV), need to be cast into integer values too. Also, there are some attributes left with a continue data type, like timestamps. Last, the dataset also contains a question mark ("?") for unavailable values for the corresponding attributes. The question marks are assigned to zero value. After all attribute values are cast into integer values, each of the attributes is linearly normalized between zero and one. Eq. (4) shows the normalizing formula:

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)} \tag{4}$$

, where z_i denotes the normalized value, x_i refers to the corresponding attribute value, and $min(x)$ and $max(x)$ are the minimum and maximum values of the attribute, respectively.

4.2 Evaluation Metrics

We use several metrics that commonly used for measuring IDS performance [15]: classification accuracy (Acc), Detection Rate (DR), False Alarm Rate (FAR). Acc shows the overall effectiveness of an algorithm. DR, also known as $Recall$, refers to the number of attacks detected divided by the total number of attack instances in the test dataset. FAR is the number of normal instances classified as an attack divided by the total number of normal instances in the test dataset. F_1 score measures a harmonic mean of precision and recall, where $Precision$ shows the number of attacks compared to the total of classified instances as an attack. Intuitively, our goal is to achieve a high Acc, DR, $Precision$ and F_1 score, and at the same time, maintain low FAR. The above measures can be defined as shown in Eqs. (5), (6), (7), (8) and (9):

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$DR = Recall = \frac{TP}{TP + FN} \tag{6}$$

$$FAR = \frac{FP}{TN + FP} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{9}$$

, where True Positive (TP) is the number of intrusions correctly classified as an attack. True Negative (TN) is the number of normal instances correctly classified as a benign packet. False Negative (FN) is the number of intrusions incorrectly classified as a benign packet. False Positive (FP) is the number of normal instances incorrectly classified as an attack.

4.3 Experimental Results

We implement our proposed scheme as shown in Algorithm 1. There are two hidden layers in the SAE network with 100 and 50 neurons accordingly. The encoder in the second layer fed with features formed by the first layer of encoder. The softmax activation function is implemented in the final stage of the SAE in order to optimize the SAE training. The 50 features extracted from the SAE are then forwarded to k-means clustering algorithm as an input. We use random initialization for k-means clustering algorithm. However, we set a certain value as a random number seed for reproducibility purpose. We compare clustering results from three inputs: original data, features from the first hidden layer of the SAE and features from the second hidden layer of the SAE as shown in Table 1.

Table 1. The evaluation of our proposed scheme

Input	DR(%)	FAR(%)	Acc(%)	$Precision$(%)	F_1(%)
Original data	100.00	57.17	55.93	34.20	50.97
1st hidden layer	100.00	57.48	55.68	34.08	50.83
2nd hidden layer	92.18	4.40	94.81	86.15	89.06

We observe the limitation of a traditional k-means algorithm, which unable to clusters complex and high dimensional data of AWID dataset, as expressed by 55.93% of accuracy only. Although 100 features coming from the 1st hidden layer achieved 100% of detection rate, the false alarm rate is still unacceptable with 57.48%. The k-means algorithm fed by 50 features from the 2nd hidden layer achieved the best performance among all as shown by the highest F_1 score (89.06%) and Acc (94.81%), also the lowest FAR (4.40%). Despite a bit lower detection rate, our proposed scheme improves the traditional k-means algorithm in overall by almost twice F_1 score and accuracy.

Fig. 4 shows cluster assignment result in Euclidean space, by our proposed scheme. Black dots represent attack instances, while gray dots represent benign instances. The location of cluster centroid for each cluster is expressed by X mark.

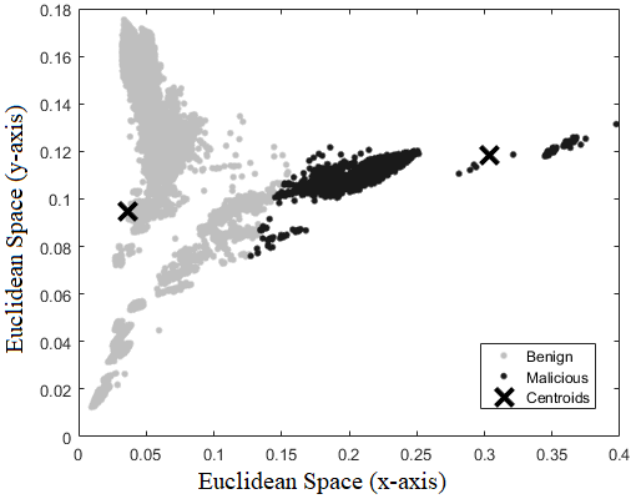

Fig. 4. Cluster assignment result in Euclidean space by our proposed scheme

We also compare the performance of our proposed scheme against two related previous work by Kolias *et al.*[3] and Aminanto and Kim [14] as shown in Table 2. Our proposed scheme is able to classify impersonation attack instances with a detection rate of 92.18% while maintaining low FAR, 4.40%. Kolias *et al.* [3] tested various classification algorithms such as Random Tree, Random Forest, J48, Naive Bayes, *etc.*, on AWID dataset. Among all methods, Naive Bayes algorithm showed the best performance by correctly classifying 4,419 out of 20,079 impersonation instances. It achieved approximately 22% DR only, which is unsatisfactory. Aminanto and Kim [14] proposed another impersonation detector by combining Artificial Neural Network (ANN) with SAE. They successfully improved the IDS model for impersonation attack detection task by achieving a DR of 65.18% and a FAR of 0.14%. In this study, we leverage SAE for assisting traditional k-means clustering with extracted features. We still have a high false alarm rate, which leads to a severe impact of IDS [16]. However, we can accept false alarm rate value about 4% since we use fully unsupervised approach here. We can adjust the parameters and cut the FAR down, but, less FAR or high DR remains a tradeoff for users and will be discussed in further work. We observe the advantage of SAE for abstracting a complex and high dimensional data to assist traditional clustering algorithm which is shown by reliable DR and F_1 score achieved by our proposed scheme.

Table 2. Comparison with previous work

Method	DR(%)	FAR(%)	Acc(%)	$Precision$(%)	F_1(%)
Kolias *et al.*[3]	22.01	0.02	97.14	97.57	35.92
Aminanto and Kim [14]	65.18	0.14	98.59	94.53	77.16
Our proposed scheme	92.18	4.40	94.81	86.15	89.06

5 Conclusion and Future Work

In this paper, we improve traditional k-means clustering algorithm by proposing a novel fully unsupervised-based intrusion detection system incorporating deep learning technique, a stacked autoencoder. We implement SAE to achieve high level abstraction of complex and huge Wi-Fi network data. The SAE has important features: model-free and learnability on large-scale data, which is suitable for the open nature of Wi-Fi networks where attackers can easily impersonate as legitimate users. We believe that the extracted features by SAE are in the new space that can improve clustering algorithm performance. Our experiments show significant improvements compared to previous work with notably 94.81% of accuracy.

In the near future, we will further investigate and propose a method to reduce false alarm rate in order to achieve a reliable IDS. In addition, we will discuss using deep learning techniques, especially stacked autoencoder as an outlier detection for detecting unknown attacks.

Acknowledgment

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)

References

1. Osseiran, A., Boccardi, F., Braun, V., Kusume, K., Marsch, P., Maternia, M., Tullberg, H. :Scenarios for 5G mobile and wireless communications: the vision of the METIS project. IEEE Communications Magazine 52.5, pp. 26–35. IEEE (2014)
2. Kolias, C., Stavrou, A., Voas, J., Bojanova, I., Kuhn, R. :Learning Internet-of-Things Security Hands-On. IEEE Security & Privacy 14.1, pp. 37–46(2016)

3. Kolias, C., Kambourakis, G., Stavrou, A., Gritzalis, S.: Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Communications Surveys and Tutorials, 18(1), pp. 184-208. IEEE (2015)
4. Beyah, R., Kangude, S., Yu, G., Strickland, B., Copeland, J.: Rogue access point detection using temporal traffic characteristics. Global Telecommunications Conference, 2004. GLOBECOM'04. Vol. 4, pp. 2271–2275. IEEE (2004)
5. Jain, A. K. :Data clustering: 50 years beyond k-means. Pattern recognition letters, 31(8), pp. 651–666. Elsevier (2010)
6. Abdi, H., Williams, L. J.: Principal component analysis. Wiley interdisciplinary reviews: computational statistics, 2(4), pp. 433–459. Wiley (2010)
7. Lee, T. W.: Independent component analysis. Independent Component Analysis pp. 27–66. Springer (1998)
8. Jiang, C., Zhang, H., Ren, Y., Han, Z., Chen, K. C., Hanzo, L.: Machine learning paradigms for next-generation wireless networks. IEEE Wireless Communications 24(2). pp. 98–105. IEEE (2016)
9. Song, C., Liu, F., Huang, Y., Wang, L., Tan, T.: Auto-encoder based data clustering. Iberoamerican Congress on Pattern Recognition. pp. 117–124. Springer Berlin Heidelberg (2013)
10. Saito, S., Tan, R.T.: Neural clustering: concatenting layers for better projections. Workshop Track of International Conference on Learning Representations (ICLR). (2017)
11. Shang, T., Gui, L. Y.: Identification and prevention of impersonation attack based on a new flag byte. Computer Science and Network Technology (ICCSNT), 2015 4th International Conference on Vol. 1, pp. 972-976. IEEE (2015)
12. Yilmaz, M. H., Arslan, H.: Impersonation attack identification for secure communication. Globecom Workshops (GC Wkshps), 2013 IEEE. pp. 1275-1279. IEEE (2013)
13. Laksmi, B., Sanmuga, L., Karthikeyan, R.: Detection and prevention of impersonation attack in wireless networks. International Journal of Advanced Research in Computer Science & Technology, 2(1), pp. 267-270. IJARCST (2014)
14. Aminanto M.E., Kim, K.: Detecting impersonation attack in Wi-Fi networks using deep learning approach. Information Security Applications. WISA 2016. Lecture Notes in Computer Science, Vol. 10144, pp. 136–147. Springer (2017)
15. Al-Jarrah, O. Y., Alhussein, O., Yoo, P. D., Muhaidat, S., Taha, K., Kim, K.: Data randomization and cluster-based partitioning for Botnet intrusion detection. IEEE transactions on cybernetics, 46(8), pp. 1796-1806. IEEE (2016)
16. Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. Security and Privacy (S&P), 2010 IEEE Symposium on, pp. 305-316. IEEE (2010)