SCIS 2017 2017 Symposium on
Cryptography and Information Security
Naha, Japan, Jan. 24 - 27, 2017
The Institute of Electronics,
Information and Communication Engineers

Weighted Feature Selection Techniques for Detecting Impersonation Attack in Wi-Fi Networks

Muhamad Erza Aminanto* Harry Chandra Tanuwidjaja*
Paul D. Yoo† Kwangjo Kim*

Abstract: As Internet-of-Things (IoT) devices enable pervasive computing in our daily lives, more and more devices are connected to Wi-Fi networks. The public access to Wi-Fi network leads to exploitable vulnerabilities that can be inverted as attacks. Impersonation attack is an active malicious action where unauthorized users masquerade themself as authorized to gain privileges. Detecting impersonation attacks remains a challenging task due to its similar properties with benign packets. Moreover, the pervasiveness of IoT devices connected to a Wi-Fi network generates a complex, large-scale, and high-dimensional data, which leads to difficulties in real-time detection and mitigation. Selecting the best features is one of the challenging issues to improve the performance of the classifier. In this study, we examine the feature weighting methods of existing machine learners and how they could be used for the accurate selection for impersonation attack features. We test and validate the utility and usefulness of the selected features using a standard neural network. This study finally demonstrates that the proposed weight-based machine learning model can outperform other filter-based feature selection models. We evaluated the proposed model on a well-referenced Wi-Fi networks benchmark dataset, namely, AWID. The experimental results not only demonstrate the effectiveness of the proposed model achieving an accuracy of 99.86% but also prove that combining a weight-based feature selection method with a light machine-learning classifier leads to a significantly *better* performance compared to the best result reported in the literature.

Keywords: intrusion detection system, impersonation attack, feature selection, Wi-Fi networks.

1 Introduction

Wireless network traffic from cellular users are increasing day by day at a very rapid rate. According to Cisco Visual Networking Index report [1], wireless traffics will account for two-thirds of total Internet traffics by 2020. We anticipate that 66% of IP traffics comes from Wi-Fi and cellular devices only. As Wi-Fi networks (IEEE 802.11) have been widely deployed for high-speed local area connectivity, the number of attacks has grown exponentially [2]. Impersonation attack is one of the most common vulnerabilities of Wi-Fi network where a legitimate user can be impersonated to obtain unauthorized access to a wireless network. Impersonation may take forms of unauthorized access, device cloning, rogue access point, address spoofing, replay, *etc* [3].

Unauthorized access may result in privacy violation due to possible theft and vandalism of network resources. Device cloning is an action that reprograms a device with another device's credentials such as IP address, MAC address, and SSID. MAC address spoofing belongs to device cloning for manipulating the duration of a frame. Rogue access point is a bait access point made by attacker that impersonates a legitimate base station. The rough access point confuses many clients trying to send or receive packets through what they believe to be a legitimate base station. Because of long disruptions of service caused by rogue access point, it can be categorized as high impact threat. Replay attack prevention can make sure that each message is freshly generated, not re-transmitting previously intercepted messages by attackers.

Impersonation attack may cause a serious breach of network security as it allows unauthorized or malicious users into internal network [4]. Some publications [5],[6] and [7] proposed new detectors particularly for impersonation attacks. Shang and Gui [5] proposed a new way of identifying impersonation attacks using differential flag byte generated from transmission data information. Yilmaz and Arslan [6] developed an impersonation detector that can determine if the signal coming from legitimate or illegal transmitter. It can also detect spoofing signals by using delay information between different transmitters. Lakshmi *et al.* [7] showed a novel way of detecting impersonation attacks by leveraging special data, a property information relating to the nodes that are complex to falsify and not dependent on any cryptographic protocol. All of the above-mentioned methods, however, are designed for particular conditions and assumptions only. There has been no general model that is capable of detecting both

* School of Computing, Korea Advanced Institute of Science and Technology (KAIST), 291 Gwahak-ro, Yuseong-gu, Daejeon, 34141, Korea. {*aminanto, elevantista, kkj*}@kaist.ac.kr.
† Bournemouth University, Fern Barrow, Poole BH12 5BB, UK. {*paul.d.yoo*}@ieee.org.

known and unknown impersonation attacks reported in the literature. Intrusion Detection System (IDS) is one of the most common components for every network security infrastructures [8]. Machine learning techniques have been well adopted as the main detection algorithm in IDS due to their model-free properties [9]. We believe that leveraging recent machine-learning methods will bring significant benefits in improving existing IDS models particularly for detecting impersonation attacks in large-scale networks.

The wide spread of computing devices using Wi-Fi networks outputs a complex, large, and high-dimensional data, which leads to difficulties in attack detection tasks. We believe feature selection techniques can improve the performance of existing machine-learning-based IDSs. One of the key contributions of this study is the introduction of the novel feature-selection-based method considering the weights of each feature coming from light-weight machine-learning models. The existing machine learners namely Support Vector Machine (SVM), Artificial Neural Network (ANN) and decision tree C4.5 are capable of extracting relevant information needed from the data. This relevant information is then presented in the weight of nodes or neurons. The weight values from a trained model indicate how important the corresponding inputs are. We select the most suitable features according to the weights provided. The small set of selected features is not only essential to real-time process but also suitable for the large-scale nature of Wi-Fi networks. The proposed approach finally ends by leveraging ANN as a classifier building an IDS model using minimum number of features only.

We evaluated the proposed approach on the AWID dataset, a Wi-Fi network benchmark dataset built by Kolias *et al.* [10]. They tested a number of existing machine-learning models on the dataset in a heuristic manner. The lowest detection rate is observed on impersonation attack reaching 22% detection accuracy only while the proposed approach outperforms on that particular category achieving 99.86% detection accuracy. Clearly, the novel way of combining weighted-based feature selection with ANN classifier improves the detection capability on impersonation attacks and can be further generalized for different attack types, both known and unknown attacks in large-scale Wi-Fi networks.

The remainder of this paper is organized as follows: Section 2 reviews related work. We describe our proposed approach in Section 3. Section 4 presents the experimental results and analysis. Finally, conclusion and future work are provided in Section 5.

2 Related Work

A lot of efforts on the detection of impersonation attacks have been reported in the literature. Shang and Gui [5] proposed a novel strategy considering differential flag byte (DFB) to detect impersonation attacks at the bottom of protocol stack with low computational complexity. While Yilmaz and Arslan [6] developed an impersonation attack detector by measuring power delay profile differences of transmitters located in different places. Last, Lakshmi *et al.* [7] showed a novel way of detecting impersonation attacks by leveraging special data, properties relating to every node which are complex to falsify and not dependent on any cryptographic protocol. They leverage special correlation of Received Signal Strength (RSS) transmitted from wireless nodes to see the impersonation attacks and using Efficient Probabilistic Packet Marking (EPPM) to detect the adversaries. Cluster-based mechanisms square measure developed to count the amount of attackers. They use SVM learner to improve the accuracy of counting the attackers. Additionally, they develop an integrated detection and localization system which will localize the positions of multiple attackers. However, abovementioned schemes above needs some modification of protocol to be done. We need one general model that able to detect an impersonation attack.

Feature selection techniques are useful in reducing model complexity leading to faster learning and real-time process. Kayacik *et al.* [11] discussed the importance of the roles of feature selection in building IDS models. They investigated the relevance of each feature in KDD 99 Dataset and provided useful discussions on the roles of information gain theories. Their work concluded with the introduction of the list of the most relevant features for each class label. A few more studies that employ feature selection techniques had been reported in the literature. Zaman and Karray [12] categorized the IDS based on the TCP/IP network model using a feature selection method named Enhanced Support Vector Decision Function (ESVDF). Louvieris *et al.* [13] proposed an effects-based feature identification IDS using Naive Bayes as a feature selection method. Manekar *et al.* [14] leveraged Particle Swarm Organization (PSO) and SVM. PSO performed feature optimization to get optimized feature, then SVM performed classification task. Similar approach also introduced by Saxena *et al.* [15]. While the concept of weighted feature selection was introduced by Schaffernicht *et al.* [16]. Exploiting SVM-based algorithms as a feature selection method was introduced by Guyon *et al.* [17]. This leveraged the weights adjusted during support vector learning resulting in ranking the importance of input features. Another related approach was proposed by Wang [18] where he ranked input features based on weights learned by an ANN. He showed that deep neural networks can be used for finding useful features existing in raw network flow data.

This paper focuses on Wi-Fi networks. Kolias *et al.* [10] published a comprehensive Wi-Fi network traces that become a public dataset for 802.11 networks. They checked various machine learning algorithms to validate their dataset in a heuristic manner. Among all the classification results obtained, impersonation attack detection was the most unsatisfactory. One of our goals in this study thus is to improve the impersonation attack detection. Recently, Usha and Kavitha

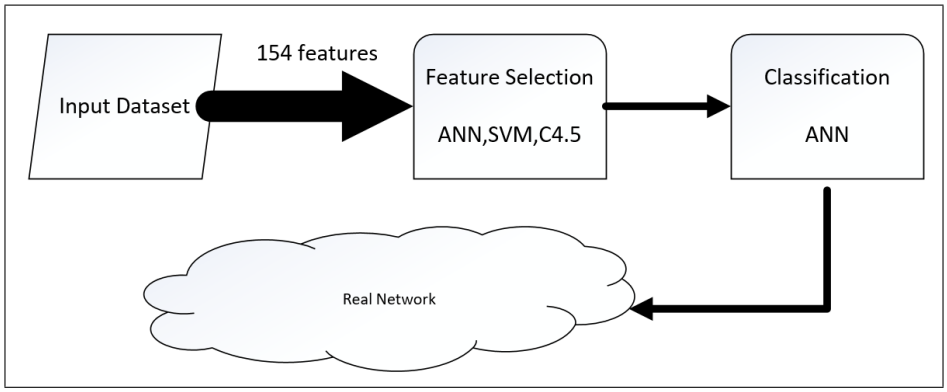

Figure 1: Our Proposed Architecture

[19] leveraged AWID dataset and successfully improved the overall detection rate. However, their work did not focus on improving impersonation attack detection which is the one of most concerns by Kolias *et al.* [10].

3 Our Approach

This section describes the proposed approach to improve impersonation attack detection. There are two main steps in this approach, feature selection and classification. Fig.1 shows the proposed architecture which begins with feature selection, ends with classification. We leverage weighted-feature selection methods using SVM, ANN and decision tree. ANN is employed for classification task in the final step.

Feature learning includes feature extraction and feature selection [18]. Feature learning is defined as a technique to model the behavior of data from the subset of attributes only. Feature learning shows the correlation between the detection performance and the traffic model quality [20]. However, feature extraction and selection are different terms. Feature extraction refers to deriving new features from raw feature space to be informative and non-redundant. Those features in raw feature and newly generated features which are usually different from each other. On the other hand, feature selection is performed to select several features from the raw feature space. Thus, new generated features are simply selected from the raw one without transformation. Both feature extraction and selection are aiming the smaller number of new generated features than the raw one.

3.1 ANN

We apply ANN as one of weighted-feature selection method. By using ANN, we are able to choose a subset of features which are important to learn the impersonation attack model based on the heuristic weights from ANN learning. We train an ANN with two target classes only, normal and impersonation attack, instead of four target classes. Fig. 2 shows the ANN model where $b1$ and $b2$ represent the bias values for the corresponding hidden layer, respectively.

We use the first hidden layer only for feature selection and consider the weight values between the first two layers in order to select the important input features. The weight represents the contribution of the input features to the first hidden layer features. The values close to zero W_{ij} means that the corresponding input feature x_j is meaningless for further propagation. Thus, one hidden layer is sufficient since we consider the weights in the first hidden layer only. We define the importante value of each input feature, as expressed by Eq. (1).

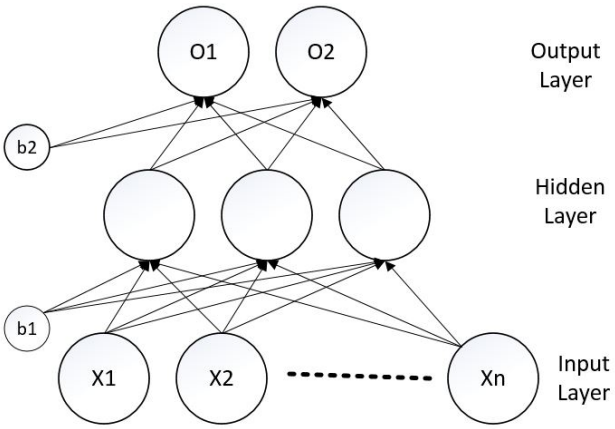

Figure 2: ANN Model

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

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 according to V_j value in a descending order. We pick some features that have V_j value bigger than a threshold value.

Besides using ANN as a weight-based feature selection, we also use an ANN as a classifier. ANN is one of most widely used pattern recognition algorithms. When learning using ANN minimum global error function is executed. It has two learning approach, supervised and unsupervised. In this study, we use a supervised ANN and leverage with scaled conjugate gradient optimizer, which is suitable for large-scale problem [21].

3.2 SVM

Support vector machine (SVM) is a supervised machine learning algorithm that can be used for classification or regression. If n is number of features, SVM plots each data as a point in n-dimensional space. The value of each feature becomes the value of a specific coordinate. After that, classification process is done by finding the hyper plane that distinguishes the two classes. SVM can handle non-linear decision border with arbitrary complexity, however in this study, we use a linear SVM because of its unique characteristics found in the dataset. As a result, the decision boundary of the SVM is a straight line in two dimensional spaces. Main computation property of SVM is called support vectors. Support vectors are the closest vector to the decision boundary. The decision function of SVM is based on support vectors, so it really affects SVMs competitive classification performance. The decision function of an input vector x can be described as shown by Eqs. (2), (3) and (4).

$$D(x) = wx + b, , \tag{2}$$

$$w = \sum_{k} \alpha_k y_k x_k \tag{3}$$

$$b = (y_k - wx_k), \tag{4}$$

From Eq. (2), we can see that decision function $D(x)$ of an input vector x is defined as the sum between multiplication of weight vector and input vector x with bias value. Weight vector w is a linear combination of training patterns. The training patterns with non-zero weights are support vectors. Bias value is the average of marginal support vector.

SVM Recursive Feature Elimination (SVM RFE) is an application of RFE using weight magnitude to perform rank clustering. In this study, we use SVM-RFE using linear case [17]. The algorithm of SVM-RFE can be described in some simple steps. The inputs are training examples and class labels. First, we initialize subset of surviving features and feature ranked list. Then, we restrict training examples to good feature and train the classifier and the weight vector of dimension length. After the value of weight vector is obtained, we compute the ranking criteria and find the feature with the smallest ranking criterion. Using that feature, the feature ranking list is updated and feature with smallest ranking criterion is eliminated. Finally, we get the feature ranked list as the output.

3.3 Decision Tree

Decision tree is one of the most popular methods implemented for classification tasks [22]. In this study, we adopt C4.5 decision tree [23] since one of most widely used decision tree method as inductive reference. C4.5 decision tree is robust from noisy data and able to learn disjunctive expressions. It has k-ary tree structure where each node inside the tree is representing a test on several attributes from the input representation data. Every branch coming down from the tree expresses possible values of feature residing at that node and different test results. Basically, C4.5 decision tree is using greedy algorithm to construct a tree in a top-down recursive divide-and-conquer approach [22]. Selecting the best attribute that results an important information for classification and generating a test node for corresponding attribute, are the beginning of C4.5 algorithm. After that, it divides the data based on their value according to test attribute which reside in the parent node. The algorithm will terminate when all data are grouped in the same class, or the process of adding additional separation is not worth anymore based on some predefined threshold.

4 Evaluation

4.1 Dataset

We evaluate our methods on AWID Dataset [10], which is one of the largest Wi-Fi network dataset collected from real network trace, as a benchmark dataset. The dataset was published in 2015 with huge and real Wi-Fi network traces. Due to its comprehensiveness and real characteristics, the AWID dataset might become common benchmark dataset for Wi-Fi network related researches. There are two types of AWID dataset based on the number of target classes. The first type named "CLS" with four target classes and the second named "ATK" with 16 target classes. The 16 classes of "ATK" dataset belong to four attack categories in "CLS" dataset. As an example, *Caffe-Latte*, *Hirte*, *Honeypot* and *EvilTwin* attack types listed in "ATK" dataset, are classified as Impersonation Attack in "CLS" dataset. Besides that, AWID dataset also divided into two types based on the size of data instances included, namely full and reduced dataset. There are 1,795,595 instances existing in full dataset, with 1,633,190 and 162,385 normal and attack instances, respectively. While, there are 575,643 instances existing in reduced dataset, with 530,785 and 44,858 normal and attack instances, respectively. In this study, we use the reduced "CLS" AWID dataset for simplicity.

The dataset express natural of network that normal instances are significantly outnumbers the attack instances [10]. The ratio between normal and attack instances is 10:1 and 11:1 for unbalanced training and test dataset, respectively. This property might be biased the training model and reduced the classification accuracy. In order to avoid this problem, we balance the dataset beforehand. The ratio between normal and attack instances after balancing process is 1:1 for both balanced training and test dataset. We train our proposed approach using balanced dataset and verify the trained model using unbalanced dataset. Table 1 shows the distribution of each classes in balanced and unbalanced dataset.

AWID dataset [10] has diverse value data types consists not only discrete data, but also continuous, and symbolic data types with flexible value range. Such kind of format will be difficult for most of pattern

Table 1: Distribution of each classes in balanced and unbalanced dataset

	Balanced	
	Normal	Impersonation
Train	163,319	48,522
Test	53,078	20,079
	Unbalanced	
	Normal	Impersonation
Train	1,633,190	48,522
Test	530,785	20,079

classification methods to learn [24]. The preprocessing process should be conducted in advance. There are two main steps for the preprocessing, the mapping step from symbolic-valued attributes into numeric values and the normalizing step. Target class will be mapped into one of these integer-valued classes: 1 for normal, 2 for impersonation, 3 for flooding and 4 for injection attack instances. Meanwhile, symbolic attributes such as receiver, destination, transmitter, source address, *etc.*, will be mapped into integer values with minimum value 1 and maximum value N, where N is the number of symbols. Some attributes that has hexadecimal data type such as WEP Initialization Vector (IV) and Integrity Check Value (ICV) need to be casted into the integer value as well. Also, there are some attributes left with continues data type, like timestamp. In addition, the dataset also contains the question mark ("?") for those not available value on the corresponding attributes. This question mark can be assigned with zero value. After all attributes values casted into the integer values, each of the attributes linearly normalized between zero and one. Eq. (5) shows the normalizing formula.

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)}, \tag{5}$$

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

Different measures that commonly used [25] are operated to evaluate the performance of our approach, namely, classification accuracy (Acc), detection rate (DR) which also known as sensitivity, false alarm rate (FAR) which also known as false positive rate, time to build model (TBM) and time to test the model (TT). We can see classifiers ability to correctly classify normal and impersonation attack by Acc value. DR refers to the number of impersonation attack detected divided by the total number of impersonation attack instances in test dataset. FAR is the number of normal instances classified as an attack divided by total number of normal instances in test dataset. While TBM and TT measure time has been taken to train and test the model, respectively. Intuitively, our goal is to achieve high Acc, DR, in the same time maintain low FAR, TBM and TT. The above measures can be defined as shown by Eqs. (6), (7) and (8).

Table 2: Classification Metric

Classification Result	Real Label of Dataset	
	Attack	Normal
Positive (Intrusion)	True Positive	False Positive
Negative (Normal)	False Negative	True Negative

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}, \tag{6}$$

$$DR = \frac{TP}{TP + FN}, \tag{7}$$

$$FAR = \frac{FP}{TN + FP}, \tag{8}$$

where true positive (TP) is the number of intrusions that correctly classified as an attack. True negative (TN) is the number of normal instances that correctly classified as a benign packet. False negative (FN) is the number of intrusions that incorrectly classified as a benign packet. False positive (FP) is the number of normal instances that incorrectly classified as an attack. In other words, we can see above definitions in Table 2

4.3 Experimental Setup

Our proposed approach is evaluated in several steps. First, we verify two feature selection approaches: filter-based and wrapper-based methods which are implemented in Waikato Environment for Knowledge Analysis (WEKA) [26]. Second, we implement ANN classifier using MATLAB R2016a which runs in Intel(R) Xeon(R) CPU E-3-1230v3@3.30 GHz, RAM 32 GB. We validate our proposed approach using unbalanced dataset in order to show our proposed approach is able to run in real Wi-Fi network and comparing with previous work [10] and [27].

4.4 Experimental Result

4.4.1 Feature Selection

We compare some weighted-feature selection methods which belong to wrapper-based featrure selection method with other filter-based feature selection methods as follow:

- **CfsSubsetEval [28] (CFS)**: Considers the predictive ability of each feature individually and the degree of redundancy between them in order to evaluates the importance of a subset of features. This approach will select subsets of features that are highly correlated with the class while having low inter-correlation.

- **Correlation (Corr)**: Measures the correlation between the feature and the class in order to evaluates the importance of a subset of features.
- **ANN**: The weight from trained ANN model mimics the importance of the correspondence input. By selecting the important features only, the training process becomes lighter and faster than before [18].
- **SVM**: Measures the importance of each feature based on the weight came from SVM classification result.
- **C4.5**: C4.5 is one of decision tree approach. It can select the subset of features that are not highly correlated. Correlated features should be in the same split, so, features that belong to different splits are not highly correlated [22].

There are two common approaches in feature selection, namely filter and wrapper methods. Filter method usually measures the correlation and redundancy of each attributes without executing learning algorithm. Therefore, filter method is lightweight and fast. On the other hand, wrapper method considers the result of learning algorithm, which leads to fit the subset of features for the chosen algorithm [29]. **CFS** and **Corr** belong to filter method, while ANN, SVM and C4.5 belong to wrapper method.

We select subset of features using wrapper method by considering each feature weight. For ANN case, we set a threshold weight value, and select features that are higher than the threshold. SVM attribute selection function results ranked features based on their weight, then we select subset of features from the highest value. Similarly, decision tree C4.5 outputs a binary tree with several level depths. We choose features that belong to most top-three layers in the tree. While CFS results fixed number of selected features and Coor provides a correlated feature list. Table 3 shows all feature lists selected from various feature selection methods. Among all selected features, we can notice the characteristic of selected features as shown in Fig. 3 that shows feature number 38 characteristic. Blue area expresses normal instances, in the same time, red area depicts impersonation attack instances characteristic. Intuitively, we are able to distinguish between normal and attack instances in the future based on the value of data instance.

The performance evaluation between feature selection methods is shown in Table 4. We can see that filter-based methods are fast to learn but lower TP accuracy. While wrapper-based methods are taking longer time to learn but we can achieve high TP accuracy.

4.4.2 Validation and Comparison with Previous Work

After completing the building model process, we test our model using ANN classifier. In order to optimized the classification result, we validate our classification task using validation dataset which is separated from training and test datasets. We separate a dataset into three parts: training data, validation data and testing data with ratio: 70 %, 15 % and 15 %, respectively. Training data are presented to the network during training, and the network is adjusted according its error. Validation data are used to measure model generalization, and to stop training when generalization stops improving. Testing data prepared for an independent measure of the model performance after training. Optimized model is reached when we have the smallest average square error for validation dataset. Fig. 4 shows one example of validation process result during ANN classification task. we can see that at epoch 163, cross entropy error starts increasing, means that at the epoch 163, the model is optimized learning model. Although training data outputs decreasing error value after epoch 163, the model is not optimizing anymore since this decreasing error expresses overfitting problem.

For testing purposes, we leverage the unbalanced test dataset in order to mimics real Wi-Fi network. The unbalanced dataset contains 530,785 normal instances and 20,079 impersonation attack instances as shown in Table 1. We examine our proposed approach with our previous work [27] and Kolias [10] as shown in Table 5. Our proposed approach with three different weighted-feature selection method are outperforming the other two previous work, especially our proposed approach with SVM is able to classify impersonation attack with 99.8619 % while maintaining really low FAR, 0.3872 % only. In the same time, Kolias [10] and our previous work [27] are unable to achieve high detection

Table 3: Selected Features among All Methods

Method	Selected Features
Filter-based Methods	
CFS	5,38,70,71,154
Corr	71,67,50,51,47,68,73,82
Wrapper-based Methods	
ANN	77,118,82,94,38,107,7,4
SVM	47,154,107,82,122,108,64,94
C4.5	71,76,68,140,119,77,38,11,107,66,61

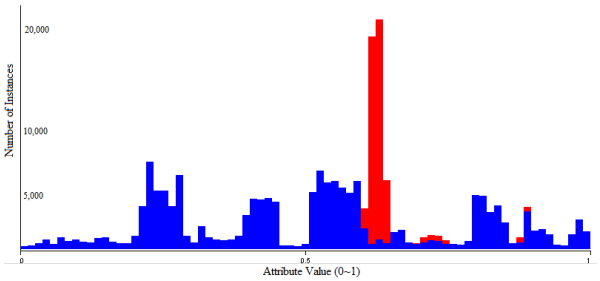

Figure 3: Feature Number 38 Characteristic

Table 4: Performance Evaluation between Feature Selection Methods

Method	Normal		Impersonation Attack		TBM (s)
	TN (%)	FP (%)	TP (%)	FN (%)	
CFS	96.6893	3.3107	94.8469	5.1531	80
Corr	99.6082	0.3918	92.0836	7.9164	2
ANN	99.5275	0.4725	99.7925	0.2075	150
SVM	99.6128	0.3872	99.8619	0.1381	10,780
C4.5	99.7708	0.2292	99.4283	0.571	1,294

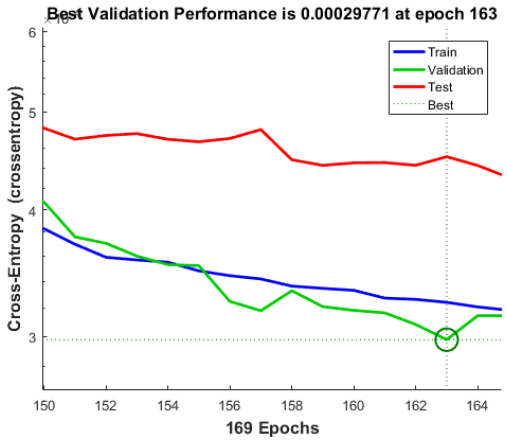

Figure 4: Best Validation Performance

Table 5: Comparison with Previous Work

Approach	DR (%)	FAR (%)
ANN	99.7925	0.4725
SVM	99.8619	0.3872
C4.5	99.4283	0.2292
Our Previous Work [27]	65.178	0.143
Kolias [10]	22.008	0.021

rate, with 22.008% and 65.178%, respectively. However, both previous works are able to achieve low and comparable FAR with our proposed approach.

5 Conclusion and Future Work

Thanks to the recent popularity of Internet-of-Things (IoT), Wi-Fi networks are getting more and more attentions. Pervasiveness of devices using Wi-Fi network is creating great opportunities for attackers to perform malicious activities. Impersonation attack in Wi-Fi network is becoming one of the most serious threats since an attacker can disguise itself so that difficult to be detected. Fortunately, the presence of IDS concept has allowed an opportunity to monitor the existing networks from any attacks including impersonation. Moreover, the recent advancement of machine learning technologies is helpful for IDS to learn and classify unknown attacks. However, the coverage and the speed of Wi-Fi networks improve daily. A method to reduce machine learning process while maintaining high accuracy and low false alarm rate is required. In this study, we presented a novel method of combining weighted-feature selection with a reliable impersonation attack detector in Wi-Fi network. High-dimensional original features are examined using weighted-feature selection method in order to eliminate redundant and unimportant features. We adopt ANN, SVM and C4.5 decision tree as a weighted-feature selection method. A few important features are sufficient to detect impersonation attack in large-scale Wi-Fi network with 99.8619% detection rate and 0.3872 % false alarm rate. In the near future, we are planning to incorporate recent advancement of machine learning methods such as deep learning, which is able to learn from complex and huge data. Therefore, the IDS model incorporating deep learning method suits Wi-Fi network property, which is large-scale data.

Acknowledgement

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-2015R1A-2A2A01006812)

References

[1] CISCO, "Cisco Visual Networking Index: Forecast and Methodology, 2015-2020," http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html, 2016, [Online; accessed 06-December-2016].

[2] C. Kolias, A. Stavrou, J. Voas, I. Bojanova, and R. Kuhn, "Learning internet-of-things security" hands-on"," *IEEE Security & Privacy*, vol. 14, no. 1, pp. 37–46, 2016.

[3] M. Barbeau, J. Hall, and E. Kranakis, "Detecting impersonation attacks in future wireless and mobile networks," *Proceeding of Secure Mobile Ad-hoc Networks and Sensors*, pp. 80–95, 2006.

[4] R. Beyah, S. Kangude, G. Yu, B. Strickland, and J. Copeland, "Rogue access point detection

using temporal traffic characteristics," *proceeding of Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, vol. 4, pp. 2271–2275, 2004.

[5] T. Shang and L. Y. Gui, "Identification and prevention of impersonation attack based on a new flag byte," *Proceeding of Computer Science and Network Technology (ICCSNT), 2015 4th International Conference on*, vol. 1, pp. 972–976, 2015.

[6] M. H. Yılmaz and H. Arslan, "Impersonation attack identification for secure communication," *IEEE Globecom Workshops (GC Wkshps)*, pp. 1275–1279, 2013.

[7] I. B. Lakshmi, B. S. Lakshmi, and R. Karthikeyan, "Detection and prevention of impersonation attack in wireless networks," 2014.

[8] C. Kolias, G. Kambourakis, and M. Maragoudakis, "Swarm intelligence in intrusion detection: A survey," *computers & security*, vol. 30, no. 8, pp. 625–642, 2011.

[9] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," *IEEE symposium on security and privacy*, pp. 305–316, 2010.

[10] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 184–208, 2015.

[11] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on kdd 99 intrusion detection datasets," in *Proceedings of the third annual conference on privacy, security and trust*. Citeseer, 2005.

[12] S. Zaman and F. Karray, "Lightweight ids based on features selection and ids classification scheme," *Proceeding of Computational Science and Engineering, 2009. CSE'09. International Conference on*, vol. 3, pp. 365–370, 2009.

[13] P. Louvieris, N. Clewley, and X. Liu, "Effects-based feature identification for network intrusion detection," *Neurocomputing*, vol. 121, pp. 265–273, 2013.

[14] V. Manekar and K. Waghmare, "Intrusion detection system using support vector machine (svm) and particle swarm optimization (pso)," *International Journal of Advanced Computer Research*, vol. 4, no. 3, p. 808, 2014.

[15] H. Saxena and V. Richariya, "Intrusion detection in kdd99 dataset using svm-pso and feature reduction with information gain," *International Journal of Computer Applications*, vol. 98, no. 6, 2014.

[16] E. Schaffernicht and H.-M. Gross, "Weighted mutual information for feature selection," *International Conference on Artificial Neural Networks*, pp. 181–188, 2011.

[17] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

[18] Z. Wang, "The applications of deep learning on traffic identification," 2015.

[19] M. Usha and P. Kavitha, "Anomaly based intrusion detection for 802.11 networks with optimal features using svm classifier," *Wireless Networks*, pp. 1–16, 2016.

[20] F. Palmieri, U. Fiore, and A. Castiglione, "A distributed approach to network anomaly detection based on independent component analysis," *Concurrency and Computation: Practice and Experience*, vol. 26, no. 5, pp. 1113–1129, 2014.

[21] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural networks*, vol. 6, no. 4, pp. 525–533, 1993.

[22] C. A. Ratanamahatana and D. Gunopulos, "Scaling up the naive bayesian classifier: Using decision trees for feature selection," 2002.

[23] J. R. Quinlan, "C4. 5: Programming for machine learning," *Morgan Kauffmann*, p. 38, 1993.

[24] M. Sabhnani and G. Serpen, "Application of machine learning algorithms to kdd intrusion detection dataset within misuse detection context." *Proceeding of MLMTA*, pp. 209–215, 2003.

[25] O. Y. Al-Jarrah, O. Alhussein, P. D. Yoo, S. Muhaidat, K. Taha, and K. Kim, "Data randomization and cluster-based partitioning for botnet intrusion detection," *IEEE Cybernetics*, 2015.

[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[27] M. E. Aminanto and K. Kim, "Detecting impersonation attack in wifi networks using deep learning approach," *Proceeding of WISA 2016*, 2016.

[28] M. A. Hall and L. A. Smith, "Practical feature subset selection for machine learning," *Proceedings of the 21st Australasian Computer Science Conference ACSC98*, 1998.

[29] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1, pp. 273–324, 1997.