Wi-Fi Intrusion Detection Using Weighted-Feature Selection for Neural Networks Classifier

Muhamad Erza Aminanto
School of Computing
KAIST
Daejeon, South Korea
Email: aminanto@kaist.ac.kr

Harry Chandra Tanuwidjaja
School of Computing
KAIST
Daejeon, South Korea
Email: elevantista@kaist.ac.kr

Paul D Yoo
Centre for Electronic Warfare
Information and Cyber
Defence Academy of
United Kingdom
Email: paul.d.yoo@ieee.org

Kwangjo Kim
School of Computing
KAIST
Daejeon, South Korea
Email: kkj@kaist.ac.kr

***Abstract*—Feature learning plays an important role in improving the learning capability of any machine learner by reducing the data complexity. As one of feature learning methods, feature selection has a crucial role for a machine learning with huge and complex input data. We examine the feature weighting methods in existing machine learners and look at how they could be used for the accurate selection of the important features. In order to validate our idea, we consider Wi-Fi networks since pervasive Internet-of-Things (IoT) devices create huge traffics and vulnerable at the same time. Detecting known and unknown attacks in Wi-Fi networks remains great challenging tasks. We test and validate the feasibility of the selected features using a common neural network. This study demonstrates that the proposed weighted-based machine learning model can outperform other filter-based feature selection models. The experimental results not only demonstrate the effectiveness of the proposed model, achieving 99.72% F_1 score, but also prove that combining a weight-based feature selection method with a light machine-learning classifier which leads to significantly *improved* performance, compared to the best result reported in the literature.**

***Keywords*—Intrusion detection system, Wi-Fi network, feature selection, artificial neural network, decision tree.**

I. Introduction

Feature learning can be characterized as a way to model the behavior of data using a small subset of attributes or instances [1]. Feature selection, one of feature learning methods, eliminates unnecessary features [2]. Feature selection can be divided to filter-based and wrapper-based depending on the deployed methodology. A filter-based (feature) selection measures the correlation and redundancy of each attribute without executing a learning algorithm. Therefore, the filter-based selection is lighter and faster than the wrapper-based selection. On the other hand, the wrapper-based selection considers the results of a learning algorithm, which makes the wrapper-based selection to fit the subset of features [3]. We propose a novel feature selection method that considers the weights of each feature coming from lightweight machine-learning models, namely Artificial Neural Network (ANN), and decision tree C4.5. Those models are capable of classifying each instance with relevant information from the data and lightweight compared to deep learning models. We select the most suitable features depending on the corresponding weight which expresses the importance of each feature. The small set of selected features is essential to a real-time detection. The proposed approach finally ends by leveraging the ANN as a classifier.

In order to verify our idea, we consider Wi-Fi networks which has been anticipated to be a major traffic in Internet traffic by 2020 [4]. As Wi-Fi networks have been widely deployed for high-speed local area connectivity, the number of attacks has grown exponentially [5]. However, there is no general model reported in the literature that is capable of detecting both known and unknown Wi-Fi attacks. An Intrusion Detection System (IDS) is one of the most common components for every network security infrastructure [6]. Machine learning techniques have been widely adopted as the main detection algorithm in the IDS due to their model-free properties [7]. We believe that leveraging recent machine-learning methods will bring significant improvements to the existing IDS models, particularly for detecting Wi-Fi attacks in large-scale networks.

We evaluate the proposed approach on the Aegean Wi-Fi Intrusion Dataset (AWID) [8]. They tested a number of existing machine-learning models on the dataset in a heuristic manner. Our proposed approach outperforms the state-of-the-art IDSs with 99.97% detection rate, 99.72% F_1 score and 0.41% false alarm rate. Clearly, the novel way of combining weighted-based feature selection with the ANN classifier improves the ability to detect impersonation attacks, and can be further generalized for different attack types, both known and unknown, in large-scale Wi-Fi networks.

The remainder of this paper is organized as follows: Section II reviews the related work. We describe our proposed approach in Section III. Section IV presents the experimental results and analysis. Finally, our conclusion and future work are described in Section V.

II. Related Work

Feature selection has been incorporated as a preprocessing step before classification task. Kayacik *et al.* [9] introduced the importance of feature selection during building IDS models. The relevance of each feature in the KDD'99 Dataset was investigated and the roles of information gain was discussed.

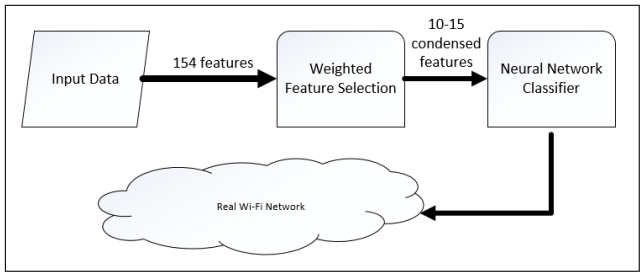

Fig. 1. Our proposed architecture consists of feature selection and classification tasks

Zaman and Karray [10] analyzed the Transmission Control Protocol/Internet Protocol (TCP/IP) network model to classify IDSs using a feature selection method called the Enhanced Support Vector Decision Function (ESVDF). Louvieris *et al.* [11] utilized naïve Bayes method in feature selection to propose an effects-based feature-identification IDS. Manekar and Waghmare [12] had an idea to combine Particle Swarm Organization (PSO) and Support Vector Machine (SVM). They applied PSO to do feature optimization, resulting in optimized feature. Subsequently, SVM carried out the classification task. Saxena and Richariya [13] also proposed a similar weighted feature selection approach while the original idea was introduced by Schaffernicht and Gross [14]. Guyon *et al.* [15] proposed feature selection method using SVM-based algorithm. They ranked the input features based on their importance during support vector learning process. Wang [16], suggested a unique related approach, using weights learned by an ANN to rank the input features. He showed that deep neural networks can be used to find useful features existing in raw network-flow data.

Several work have been done in Wi-Fi network attacks detection. Kolias *et al.* [8] 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, the impersonation attack detection was found to be the most unsatisfactory. Our goal is to improve overall detection rate including the impersonation attack detection. Recently, Usha and Kavitha [17] used the AWID dataset and successfully improved the overall detection rate. We implement different feature selection method in order to achieve near-perfect Wi-Fi attacks detection.

III. Our Approach

There are two main tasks in our proposed architecture: feature selection and classification tasks. Fig.1 shows our proposed architecture which begins with feature selection, and ends with classification task. Feature selection is performed to select several features from the raw feature space. New generated features are simply selected from the raw ones without transformation. Feature selection aims for the smaller number of new generated features than the raw ones. We leverage weighted feature-selection methods using ANN and C4.5. The ANN is also employed for classification task in the final step. Algorithm 1 explains the pseudocode of our proposed architecture. Further details of each method are explained in the following sub-sections.

Algorithm 1 Pseudocode of our proposed architecture

procedure Weighted Feature Selection
  function Dataset Normalization(Raw Dataset)
    return $InFeats$
  end function
  function Feature Selection($InFeats$)
    switch ANN **do**
      case Weighted-ANN($InFeats$)
        return $OutFeats$
      case Weighted-C4.5($OutFeats$)
        return $OutFeats$
  end function
  procedure Classification($OutFeats$)
    Training ANN
    Min $E = \frac{1}{\nu}\sum_{j=1}^{\nu}\sum_{i=1}^{\kappa}\left[z_{ij}\log y_{ij} + (1-z_{ij})\log\left(1-y_{ij}\right)\right]$
  end procedure
end procedure

A. ANN

We apply an ANN as weighted feature-selection method. By using ANN, our model is able to choose a subset of features which are of utmost significance in order to learn the attack model based on the heuristic weights from ANN learning. Fig. 2 shows the ANN model where b_1 and b_2 depict the bias values for each hidden layer, respectively.

We use the first hidden layer only for feature selection and then consider the weight values between the first two layers in order to select the important input features. The weight expresses the share of the input features to the first hidden layer. The values close to zero for W_{ij}, which means that the corresponding input feature X_j is meaningless for next hidden layer H_i. Therefore, the value of weights can be considered to measure the importance of a feature. One hidden layer is sufficient, since we consider the weights in the first hidden layer only. We define the important value of each input feature as expressed by Eq. (1):

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

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

Besides using an ANN for a weight-based feature selection, we also utilize an ANN as a classifier. The ANN is known to be one of the most popular pattern recognition algorithms. We use a supervised ANN and leverage it with a scaled conjugate gradient optimizer, which is suitable for an exhaustive problem [18].

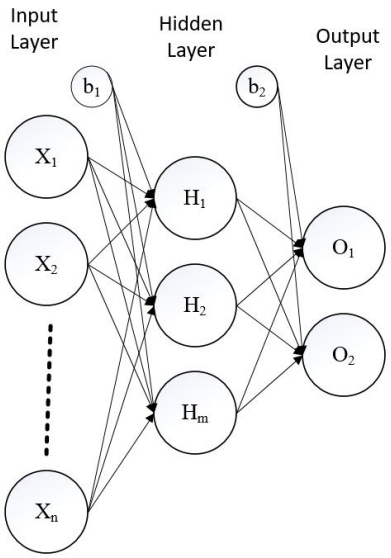

Fig. 2. ANN Model

B. Decision Tree

We adopt C4.5 decision tree [19], which is one of the most popular decision tree methods, as inductive reference. C4.5 decision tree is robust from noisy data and can learn disjunctive expressions. It has a tree structure, where each node inside the tree represents a test of several attributes from the input representation data. Every branch of the tree expresses possible values of features residing at that node and different test results. C4.5 uses a greedy algorithm to construct a tree in a top-down recursive divide-and-conquer approach [20]. The first step of C4.5 algorithm is to select the best attribute that results in important information for classification and to generate a test node for corresponding attributes. For the next step, it divides the data based on their values according to test attributes that reside in the parent node. The algorithm finishes when all data has been grouped in the same class, or the process of adding additional separations does not improve the classification performance, based on some predefined threshold. Subsequently, feature selection phase starts. It begins with selecting top-three level nodes, then removes the equal nodes and updates the list of selected features.

IV. Evaluation

Our proposed approach is evaluated in several experiments. First, we verify two feature selection approaches: filter-based and wrapper-based methods, which are implemented in the Waikato Environment for Knowledge Analysis (WEKA) [21]. Second, we implement the ANN classifier using MATLAB R2016a running on an Intel Xeon E-3-1230v3 CPU @3.30 GHz with 32 GB RAM. We validate our proposed approach using the unbalanced dataset, in order to show that our proposed approach can run in a real Wi-Fi network. We also compare our proposed approach with the state-of-the-art IDS in Wi-Fi networks.

TABLE I
Distribution of each attack class in AWID dataset

Class	Training	Test
Impersonation	48,522	20,079
Flooding	48,484	8,097
Injection	65,379	16,682
Total	162,385	44,858

A. Dataset

We evaluate our methods on the AWID Dataset [8] 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 the common benchmark dataset for Wi-Fi network-related researches. We use AWID-CLS-R-Trn and AWID-CLS-R-tst for training and test dataset, respectively. There are 1,795,575 instances in the training dataset with 1,633,190 and 162,385 normal and attack instances, respectively. While the test dataset contains 575,643 instances with 530,785 and 44,858 normal and attack instances, respectively. Table I shows the distribution of attack classes for both training and test datasets.

The dataset expresses the nature of a network which normal instances significantly outnumber attack instances [8]. The ratio between normal and attack instances are 10:1 and 11:1 for unbalanced training and test datasets, respectively. This may cause bias to the training model, and reduce classification accuracy. In order to avoid this problem, we balance the dataset in advance. The ratio between normal and attack instances after the balancing process is 1:1 for both balanced training and test datasets. The normal instances are randomly reduced into 163,319 data instances for training dataset while 53,078 data instances for test dataset. We train our proposed approach using the balanced dataset and verify the trained model using the unbalanced dataset.

Preprocessing should be conducted in advance since the AWID dataset [8] has diverse value data types. We use two main steps for preprocessing: mapping symbolic-valued attributes to numeric values, and the normalizing step. Symbolic attributes would be mapped to integer values with a minimum value of 1 and a maximum value of N, where N is the number of symbols. Some attributes that have a hexadecimal data type need to be casted into integer values, as well. Also, there are some attributes left with a continues data type. In addition, the dataset also contains the question mark ("?") for unavailable values for the corresponding attributes. The question marks are assigned to zero value [22]. After all attribute values are casted into integer values, each of the attributes is linearly normalized between zero and one. Eq. (2) shows the normalizing formula:

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)}, \tag{2}$$

where z_i expresses the normalized value, x_i denotes to the corresponding attribute value, and $min(x)$ and $max(x)$ are the

TABLE II
FEATURE SET OBTAINED AMONG ALL SELECTION METHODS

Method	Selected Features
CFS	4,8,47,68,71
Corr	67,50,51,47,75,71,9,8,154,145
ANN	131,134,93,70,120,83,79,136,94,90,75,140,142,64,66, 73,67,38,118,82,112,107,68,7,4
C4.5	67,75,78,38,79,110,8,76,119,70,1,107,80,4,111,77, 61,2,3,112,108,76,109,82

minimum and maximum values of the attribute x, respectively.

B. Evaluation Metrics

The different measures commonly used [23] are invoked to evaluate the performance of our approach: Detection Rate (DR), False Alarm Rate (FAR), time to build the model (TBM), and time to test the model (TT). Intuitively, our goal is to achieve a higher DR and F_1 score, and at the same time, maintain lower FAR, TBM and TT. 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.

DR refers to the number of attacks detected divided by the total number of attack instances in the test dataset as expressed by Eq. (3).

$$DR = \frac{TP}{TP + FN}, \tag{3}$$

FAR is the number of normal instances classified as an attack divided by the total number of normal instances in the test dataset as shown by Eq. (4).

$$FAR = \frac{FP}{TN + FP}, \tag{4}$$

F_1 score measures a harmonic mean of precision and recall as expressed by Eq. (5).

$$F_1 = \frac{2TP}{2TP + FP + FN}, \tag{5}$$

C. Experimental Result

We compare our ANN and C4.5 weighted-feature selection methods, which belong to wrapper-based feature selection methods, with CFS and Corr filter-based feature selection methods. CFS (CfsSubsetEval) [24] considers the predictive ability of each feature individually and the degree of redundancy between them, in order to evaluate 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. While Corr (Correlation) measures the correlation between the feature and the class in order to evaluate the importance of a subset of features.

TABLE III
MODEL COMPARISONS ON SELECTED FEATURES

Method	DR (%)	FAR (%)	F_1 (%)	TBM (s)	TT (s)
CFS	93.40	3.22	94.72	108	**12**
Corr	98.19	2.05	97.89	**4**	22
ANN	**99.97**	1.46	99.13	661	25
C4.5	99.92	**0.41**	**99.72**	142	38

TABLE IV
ANN MODEL COMPARISONS ON NUMBER OF FEATURES

Feats	DR (%)	FAR (%)	F_1 (%)	TBM (s)	TT (s)
5	78.52	3.24	86.12	661	**7**
10	99.53	2.71	98.19	661	24
15	99.23	1.88	98.52	661	20
25	**99.97**	**1.46**	**99.13**	661	25

We select a subset of features using wrapper method considering each feature weight. For ANN, we first set a threshold weight value. The subset of features with higher weight value than a predefined threshold value are then selected. Similarly, C4.5 produces a binary tree with several level depths. We select the features that belong to the top three levels in the tree. CFS produces a fixed number of selected features and Corr provides a correlated feature list. Table II shows all feature set obtained among all selection methods.

Table III shows model comparisons on selected features. Best performance results are in bold. ANN achieved the highest DR (99.97%). However, C4.5 is the best performer model since it achieved highest F_1 score (99.72%) and FAR (0.41%). As expected, filter-based methods (CFS and Corr) built their models quickly, which is $4s$ only by Corr. However, CFS attained the lowest performance among all models. Filter-based feature selection methods take much shorter time compared to the time taken by weighted feature selection. However, weighted feature selection improves the filter-based feature selections performance significantly.

We observe the impact of different amount of features involved during testing experiments. Table IV describes the weighted-ANN model comparisons, while Table V describes the weighted-C4.5 model comparisons with respect to the number of features. The time taken to build the model is always the same due to the time taken during feature selection task. The best performer models for both weighted-ANN

TABLE V
C4.5 MODEL COMPARISONS ON NUMBER OF FEATURES

Feats	DR (%)	FAR (%)	F_1 (%)	TBM (s)	TT (s)
5	95.52	2.16	96.45	143	18
10	98.52	1.70	98.26	143	**12**
15	99.11	0.82	99.08	143	34
25	**99.92**	**0.41**	**99.72**	143	38

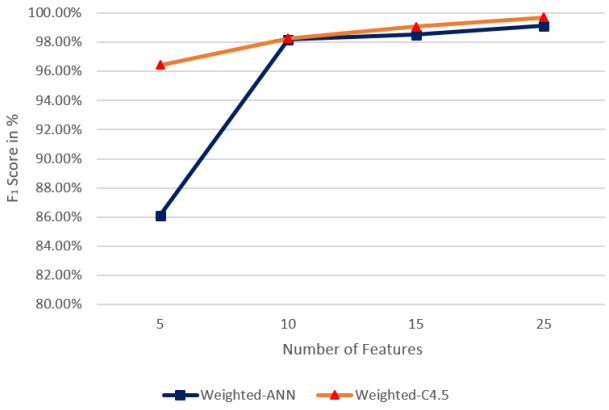

Fig. 3. Model performance comparison between Weighted-ANN and Weighted-C4.5 in terms of F_1 score

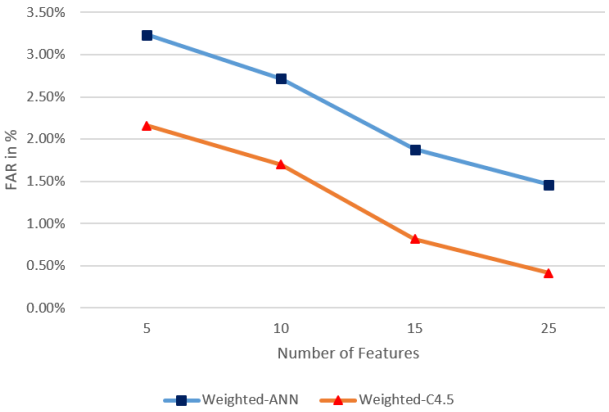

Fig. 4. Model performance comparison between Weighted-ANN and Weighted-C4.5 in terms of FAR

and weighted-C4.5 are the tests which involved 25 features achieving the highest DR and F_1 score while maintaining the lowest FAR. However, at the point of 25 features, the longest time was taken to test both models.

We compare the weighted-ANN and weighted-C4.5 with respect to number of features involved during the training as depicted in Figs. 3, 4, and 5 in terms of F_1 score, FAR and TT, respectively. The square-dotted line shows the performance of the proposed weighted-ANN method, while the triangle-dotted line represents weighted-C4.5. Fig. 3 shows that both methods can classify the attack instances with a few selected features only. Weighted-C4.5 achieved high F_1 scores since the point of 5 features only that involved during the test, since the previously selected features are more informative. While weighted-ANN has low F_1 score at the point of five features only, however, we achieved comparable F_1 scores at the point of 10 features afterwards. Fig. 4 shows the same pattern for both weighted-ANN and weighted-C4.5. However, weighted-C4.5 always achieved the lower FAR than weighted-ANN. As expected, the smaller the number of features involved, the faster time taken for testing experiments as shown in Fig. 5 by the smallest number of TT at 5 features for weighted-ANN.

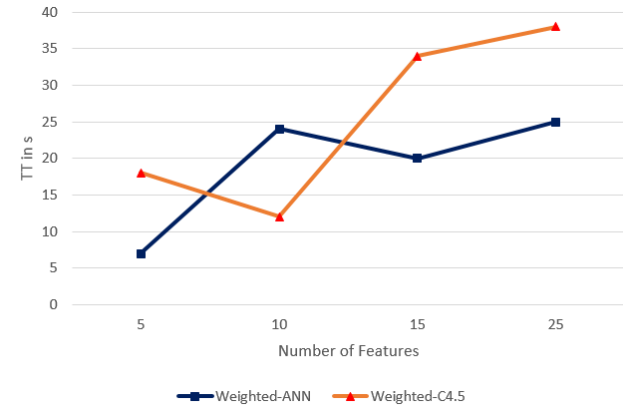

Fig. 5. Model performance comparison between Weighted-ANN and Weighted-C4.5 in terms of TT

TABLE VI
COMPARISONS WITH OTHER WORK

Approach	DR (%)	FAR (%)	F_1 (%)	TBM (s)
Weighted-ANN	**99.97**	1.46	99.13	661
Weighted-C4.5	99.92	0.41	**99.72**	**143**
SAE [25]	72.00	**0.15**	82.94	3,600
J48 [8]	96.30	3.99	94.80	569
NMI [17]	99.20	2.00	98.00	7,200

For a sake of the fairness, we leverage the unbalanced test dataset in order to mimic real Wi-Fi networks. The unbalanced dataset contains 530,785 normal instances and 44,858 attack instances. We examine our proposed approach against the state-of-the-art of IDSs in Wi-Fi networks, namely, Stacked Auto Encoder (SAE) classifier by our previous work [25], J48 classifier by Kolias *et al.* [8], and a Normalized gain based IDS for MAC Intrusions (NMI) by Usha *et al.* as shown in Table VI. Our proposed approach with weighted-feature selection methods outperforms the other three previous studies. In particular, our weighted-ANN is able to classify Wi-Fi attacks with the highest detection rate, 99.97%. While our proposed approach with weighted-C4.5 achieved the best performance with 99.72% F_1 score. We believe C4.5 achieved better performance than ANN because C4.5 is able to separate features based on the information gain value coming from test attributes. The lowest FAR has been achieved by our SAE classifier [25], since SAE adopts unsupervised learning. Thus, the classification performance excelled for one particular class only, which is normal class. Therefore, the SAE [25] achieved low attack detection as expressed by low DR. In summary, we observe the advantage of leveraging weighted-feature selection using ANN and C4.5, which is shown by high DR and F_1 score while maintaining low FAR.

V. Conclusion and Future Work

We discussed a novel method of combining weighted-feature selection with a reliable Wi-Fi attacks detector in wireless networks. High-dimensional original features are examined using a weighted-feature selection method in order to eliminate

redundant and unimportant features. We adopt ANN and C4.5 as a weighted-feature selection method. A condensed important features are sufficient to detect Wi-Fi attacks in a large-scale Wi-Fi network, with a 99.72% F_1 score and a 0.41% false alarm rate. In the near future, we plan to incorporate recent advancements in machine learning methods, such as deep learning, which is able to learn from complex and huge amounts of data. Therefore, the IDS model incorporating a deep learning method suits the Wi-Fi networks property, which is large-scale data.

Acknowledgements

This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (2013-0-00396, Research on Communication Technology using Bio-Inspired Algorithm) and National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2015R1A-2A2A01006812)

References

[1] 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, Wiley Online Library, 2014.

[2] H. Motoda and H. Liu, "Feature selection, extraction and construction," *Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol*, vol. 5, pp. 67–72, 2002.

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

[4] CISCO, "Cisco Visual Networking Index: Forecast and Methodology, 2015-2020," http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-net-working-index-vni/complete-white-paper-c11-481360.html, 2016, accessed December 6, 2016.

[5] 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, IEEE, 2016.

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

[7] 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, IEEE, 2010.

[8] 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, IEEE, 2015.

[9] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on KDD99 intrusion detection datasets," *Proceedings of the Third Annual Conference on Privacy, Security and Trust*, Citeseer, 2005.

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

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

[12] 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, Accents, 2014.

[13] 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, Foundation of Computer Science, 2014.

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

[15] 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, Springer, 2002.

[16] Z. Wang, "The applications of deep learning on traffic identification," *Blackhat USA 2015*, 2015.

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

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

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

[20] C. A. Ratanamahatana and D. Gunopulos, "Scaling up the naive Bayesian classifier: Using decision trees for feature selection," *IEEE International Conference on Data Mining (ICDM 2002)*, IEEE, 2002.

[21] 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, ACM, 2009.

[22] D. T. Larose, "Data preprocessing," *Discovering Knowledge in Data: an Introduction to Data Mining*, pp. 27–40, John Wiley & Sons, 2014.

[23] 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 Transactions on Cybernetics*, pp. 1796–1806, IEEE, 2015.

[24] M. A. Hall and L. A. Smith, "Practical feature subset selection for machine learning," *Proceedings of the 21st Australasian Computer Science Conference ACSC 1998*, pp. 181–191, Springer, 1998.

[25] M. E. Aminanto and K. Kim, "Detecting impersonation attack in Wi-Fi networks using deep learning approach," *Proceeding of WISA 2016*, Springer, 2016.