Enhancing Wireless Intrusion Detection Using Machine Learning Classification with Reduced Attribute Sets

Razan Abdulhammed
School of Engineering
University of Bridgeport
Bridgeport, CT, USA
rabdulha@my.bridgeport.edu

Miad Faezipour*
School of Engineering
University of Bridgeport
Bridgeport, CT, USA
mfaezipo@bridgeport.edu

Abdelshakour Abuzneid
School of Engineering
University of Bridgeport
Bridgeport, CT, USA
abuzneid@bridgeport.edu

Ali Alessa
School of Engineering
University of Bridgeport
Bridgeport, CT, USA
aalessa@my.bridgeport.edu

***Abstract*—In cybersecurity, machine learning approaches can predict and detect threats before they result in major security incidents. The design and performance of an effective machine learning based Intrusion Detection System (IDS) depends upon the selected attributes and the classifier model. This paper considers multi-class classification for the Aegean Wi-Fi Intrusion Dataset (AWID) where classes represent 17 types of the IEEE 802.11 MAC Layer attacks. The proposed work extracts four attribute sets of 32, 10, 7 and 5 attributes, respectfully. The classifiers achieved high accuracy with minimum false positive rates, and the presented work outperforms previous related work in terms of number of classes, attributes and accuracy. The proposed work achieved maximum accuracy of 99.64% for Random Forest with supply test and 99.99% using the 10-fold cross validation approach for Random Forest and J48.**

***Index Terms*—AWID-ATK-R; multi-class; machine learning; WIDS.**

I. Introduction

As Wi-Fi usage and demand continues to grow, there is an increasing concern over security challenges, vulnerabilities and threats in Wi-Fi. Intrusion Detection Systems (IDS) can determine and possibly respond to a threat at the right time. In general, intrusion detection systems can be grouped into: signature-based detection and anomaly-based detection. Signature-based detection maintains a database of various known attack signatures. Thus, it reveals the use of patterns from prior known threats. On the other hand, anomaly-based detection methods examine and recognize intrusions behavior based on verifying the prior activities and checks for any deviations from the normal traffic behavior. For decades, IDS developers employed various approaches to build an IDS. One of these approaches is machine learning, and more specifically; classification to detect cyberattacks. This approach requires a dataset that encompasses training and testing data. Recently, in 2015, the Aegean Wi-Fi Intrusion Dataset (AWID) was released as a comprehensive 802.11 network dataset. AWID is a labeled dataset derived from real Wi-Fi traffic traces [1]. In this paper, we used AWID-ATK-R which is one of the subsets of the AWID [1]. The purpose of these experiments is to explore the attribute space of the AWID and various classifier models to enhance the performance of wireless intrusion detection in terms of accuracy and the number of utilized attributes. In our experiments, we trained 7 classifiers of different learning styles such as Rule-based, Meta Heuristics, Tree [2], Bayes [3], and function-based learning models. The classifiers we used in our design and experiments are OneR, Bagging [4], J48, Random Forest [2], Naive Bayes [3], Multi-Layer Perceptron CS and Simple Logistic from different learning styles. Furthermore, the performance of the trained classifiers were tested with four different number of selected Attribute Group Sets (AGS): 32-AGS, 10-AGS, 7-AGS, and 5-AGS. The selected attributes group sets were able to achieve high accuracy. The key contributions of this work include extracting a new subset of 5 attributes that produce high accuracy with minimum false positives (FP) and validated using the 10-folds cross validation approach.

The overall structure of the remainder of this paper is organized as follows. Section II provides an overview of the related work. Section III gives a brief review of the AWID dataset and further describes the methodology, experimental approach as well as the performance evaluation metrics. Section IV summarizes the principal findings of these experiments and discusses the results. Finally, we conclude our paper in Section V.

II. Related Work

The section that follows gives a brief synopsis of the relevant literature with an emphasis on prior related work that utilized the AWID. To start, Kaleem et al. [7] proposed an agent-based malicious detection model that uses Artificial Neural Network (ANN) for the detection process. The authors applied their model on the AWID-CLS-R subset to classify each instant either as normal or a threat. It has been shown that the model provides highly accurate results (99.3%). Thing [9] also used the AWID-CLS-R dataset, which has 4 classes. The author considered a multi-class classification utilizing a deep learning approach that achieved an overall accuracy of 98.67% using 154 attributes. Kolias et al. [1] applied 8 conventional supervised machine learning classifiers to perform the attack

978-1-5386-2070-0/18/$31.00 ©2018 IEEE

TABLE I
SUMMARY OF THE RELATED WORK

Study	Dataset	Approach	Attributes	Classes	Accuracy (%)
Kolias [1]	AWID-CLS-R	J48	20	4	96.2
Aminanto [5]	AWID-CLS-R	ANN	154	2	99.86
Aminanto [6]	AWID-CLS-R	SAE	154	4	97.7
Kaleem [7]	AWID-CLS-R	ANN	7	2	99.3
Thanthrige [8]	AWID-CLS-R	Random Forest	111	4	94.83
Thanthrige [8]	AWID-CLS-R	Random Tree	41	4	95.12
Thanthrige [8]	AWID-CLS-R	J48	10	4	92.44
Thing [9]	AWID-CLS-R	Deep learning	154	4	98.67
Thanthrige [10]	AWID-ATK-R	Random Tree	111	4	94.58
Thanthrige [10]	AWID-ATK-R	Random Forest	41	4	94.97
Thanthrige [10]	AWID-ATK-R	Random Forest	10	4	92.29

TABLE II
AWID-ATK-R CHARACTERISTICS [1]

Filename	Classes	size	Type	Total Records	Normal Records	Attack Records	Ratio
AWID-ATK-R-Trn	10	Reduced	Training	1795575	1633190	162385	3:2
AWID-ATK-R-Tst	15	Reduced	Test	575643	530785	44858	3:2

classification on the AWID-CLS-R subset. They used the AdaBoost, J48, Naive Bayes, OneR, Random Forest, Random Tree and ZeroR algorithms. The authors carried out manual attribute selections and incorporated 20 attributes to train the classifiers. The overall accuracy of their classifiers ranged from 89.43% to 96.2%. Aminanto et al. [6] proposed a framework for IDS using a Stacked Auto Encoder (SAE), which is an unsupervised learning method for attribute selection. The framework used regression layer, a supervised learning technique, with SoftMax activation function for the classification process. The highest accuracy was 97.7% with 4 classes and 154 attributes. Moreover, Aminanto et al. [5] employed three machine learning methods on AWID-CLS-R to select the best attributes to improve the detection process of only impersonation attacks. The authors have eliminated 2 classes out of 4 classes of the AWID-CLS-R and kept two classes which are impersonation attacks and normal traffic classes. They employed Support Vector Machine (SVM), Artificial Neural Network (ANN), and Decision Tree. ANN was used for the attack classification. The results show that their approach was 99.86% accurate in detecting impersonation attacks. Thanthrige et al. [8] applied 5 supervised machine learning classifier algorithms to perform the attack classification on the AWID-CLS-R and AWID-ATK-R subsets. The algorithms they used included AdaBoost, J48, OneR, Random Forest and Random Tree. The selected attributes were evaluated and ranked by using Information Gain and Chi-Square measures. The classifiers were applied on both datasets with different settings of attributes based on the attribute evaluation results. The highest achieved accuracy was 95.12% using Random Tree applied on AWID-CLS-R with a total number of 41 attributes. In addition, the highest achieved accuracy was 94.97% using the Random Forest classifier applied on AWID-ATK-R using the same number of attributes. The results show that reduced number of attributes (41 attributes) enhances the accuracy. However, the accuracy decreased when the authors further reduced the attributes to 10. Table I provides a summary of the related work.

III. METHODOLOGY AND EXPERIMENTAL PROCEDURES

This section provides a brief summary of the Aegean Wi-Fi Intrusion Dataset (AWID) with a focus on the AWID-ATK-R subset and also gives an overview of our methodology and how we carried out our experiments. The procedure mainly includes Preprocessing, Attributes selection and Classification (Figure 1).

A. AWID-ATK-R Structure

AWID-ATK-R, a subset of AWID, is a labeled dataset with a total number of 155 attributes. AWID-ATK-R was collected based on real traces of normal and intrusion activities of the 802.11 Wi-Fi network [11] and it has a finer grained class labeling corresponding to the attack name. The training set consists of 10 classes namely Amok, Arp, Authentication request, Beacon, Caffe latte, Deauthentication, Evil twin, Fragmentation, Probe response and Normal. The total number of records in the training set is 1,765,000. The normal traffic encompasses 1,633,190 records while the attacks records are 162,358. On the other hand, the total number of records in the test set is 575,643 records. The normal traffic comprises of 530,785 records. At the same time, the attacks records are 44,585. In addition, the test set has 15 classes with 7 different classes compared to the training set namely Chop-chop, CTS, Disassociation, Hirte, Power-saving, Probe request and RTS. Furthermore, Probe response and Authentication-request are included in the training set only. Table II highlights the characteristics of the AWID-ATK-R subset.

B. Preprocessing

Preprocessing is an important component in the machine learning experiments, and plays a key role in the classifier model as well as enhancing the overall classification performance. To begin this process, we determine the attribute types of the dataset which include numerical and non-numerical data. To be more specific, the service set identifier (SSID) attribute is a non-numerical attribute (string). Other attributes

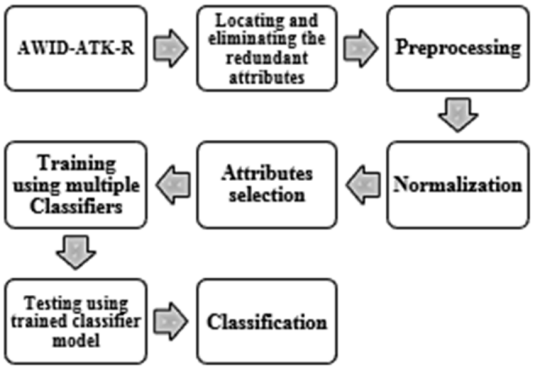

Fig. 1. Experimental Procedure

are numeric. Furthermore, some of the attributes are vital for intrusion detection, whereas other attributes may act as noise; causing a negative impact on the training speed and the accuracy. Thus, the first step in this process was to eliminate the unnecessary attributes before moving to the next step. This work has traditionally employed a simple preprocessing stage were the hexadecimal values in AWID are transformed into integer representation. After the preprocessing step, the attributes' scales within the dataset were heavily imbalanced. Thus, normalization was applied on the selected sets. The overall experimental procedure is shown in Figure 1.

C. Attribute Selection

The AWID-ATK-R-Trn dataset was used to train the machine learning techniques and the AWID-ATK-R-Tst dataset was used to evaluate the performance of the machine learning techniques. In the first step of our experiments, a total of 32 attributes were selected based upon reliable and accurate manual attribute selection and recommendation from previous related work. Cited work [12] ranked the most important attributes as those related to the MAC header. In addition, in previous studies of wireless intrusion detection systems (WIDS), certain attributes have been found to be related to the intrusion detection process [1], [8]. Table III shows the 32 attributes set (32-AGS).

In the second step, we used the Correlation Feature Selection (CFS) measure [13] to evaluate the attributes space along with the Best First Search method of forward direction [14], which is a heuristic search strategy to evaluate the attributes and select the highly class-correlated ones, yet uncorrelated to one another. Based on that, 10 attributes were selected. These attributes are listed as follows: frame.time.relative, frame.len, radiotap.channel.type.cck, wlan.fc.subtype, wlan.fc.dc, wlan.fc.pwrmgt, wlan.ta, wlan.seq, wlan.wep.iv, and data.len.

In the third step, we used the Harmony Search (HS) algorithm [15] which is one of the meta-heuristic algorithms that mimics the improvisation process of music players. One advantage of HS is that it avoids the problem of imposing complicated mathematical requirements and it is not sensitive to the initial value settings [15]. Here, we incorporate the HS algorithm with the Cost Sensitive Subset Evaluator (CostSensitiveSubsetEval), which is a meta subset evaluator that makes its base subset evaluator cost-sensitive [16]. The algorithm takes a cost matrix and a base evaluator. If the base evaluator can handle instance weights, then the training data is weighted according to the cost matrix, otherwise the training data is sampled according to the cost matrix. Based on this, a 7 Attributes Group Set (7-AGS) was selected that is consisted of radiotap.mactime, radiotap.channel.type.cck, wlan.fc.subtype, wlan.fc.pwrmgt, wlan.bssid, wlan.seq, and data.len.

Finally, we used the CFS algorithm along with the Harmony Search technique [15], based upon which, the 5 Attributes Group Set (5-AGS) was selected. The list of this 5-AGS are epoch.time, frame.len, wlan.duration, wlan.ra, and data.len.

D. Performance Evaluation Metrics

The metrics that were used to evaluate the performance of the chosen classifiers are the following:
(1) True Positive Rate (TPR) refers to the number of intrusions that are correctly classified as an attack by the classifier.
(2) False Positive Rate (FPR) or false alarm, is a common term which encompasses the number of normal instances incorrectly classified by the classifier as an attack. The FPR is calculated using Equation 1.

$$FPR = \frac{FP}{TN + FP} \tag{1}$$

(3) Accuracy (Acc) is defined as the ability measure of a classifier to correctly classify an object or data instance as either normal or attack. The Accuracy can be defined using Equation 2.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

(4) Precision represents the number of positive predictions divided by the total number of positive class values predicted. It is considered as a measure for the classifier exactness. A low value indicates large number of False Positives. The precision is calculated using Equation 3.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

(5) Recall is the number of True Positives divided by the number of True Positives and the number of False Negatives. Recall is considered as a measure of a classifier completeness such that a low value of recall realizes many False Negatives [17]. Recall is estimated through Equation 4.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

(6) The Receiver Operating Characteristic (ROC) Curve Area is another metric that can be used to evaluate the classifier test model. The Accuracy is evaluated through the area under the ROC curve. More specifically, an area closer to 1 exemplifies a perfect test, whereas, an area equal or less than 0.5 exemplifies a worthless test [17].

TABLE III
THE 32 ATTRIBUTES GROUP SET (32-AGS)

Attributes Name			
frame.time.relative	radiotap.channel.type.cck	wlan.ta	wlan.seq
frame.len	radiotap.dbm.antsignal	wlan.sa	wlan.gt.fixed.Listen.ival
wlan.fc.type.subtype	wlan.fc.type	wlan.bssid	epoch.time
wlan.duration	wlan.fc.dc	wlan.fc.subtype	Frame.time.delta
wlan.mgt.tim.dtim.period	frame.ignored	radiotap.mactime	wlan.fc.retry
wlan.ra	wlan.da	wlan.mgt.fixed.timestamp	wlan.mgt.fixed.beacon
wlan.mgt.tim.dtim.period	wlan.fc.pwrmgt	wlan.fc.protected	data.len
wlan.mgt.tagged.all	wlan.mgt.fixed.reason.code	wlan.wep.iv	wlan.wep.key

(7) Precision-Recall Curves (PRC) Area is an important measure to visualize the performance of a classifier. The PRC Area is especially important to observe the quality of the model if the dataset contains imbalanced classes. A PRC value closer to 1 indicates a good classifier model.
(8) The F-measure is a measure of a classifier's accuracy and is defined as the weighted harmonic mean of the precision and recall measures of the classifier. The F-Measure is calculated using Equation 5.

$$F\text{-}Measure = 2 \times \frac{PrecisionRecall}{Precision + Recall} \tag{5}$$

(9) MCC is the Matthews Correlation Coefficient and can be thought as a measure of the quality of classification [18]. MCC is calculated using Equation 6.

$$MCC = \frac{(TP \times TN)(FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{6}$$

(10) Time represents the required time to build the classifier model.

IV. RESULTS AND DISCUSSION

In this section, we present the principal findings of the current experiments. All the work for these experiments was carried out using Intel Core i7 3.30 GHz, 16 GB RAM system running Windows 10 and the Waikato Environment for Knowledge Analysis (Weka) software. The results obtained from the preliminary analysis of the performance of classifiers with 32-AGS, 10-AGS, 7-AGS and 5-AGS are summarized in Tables IV, V, VI and VII, respectively. What stands out from these Tables is that the highest result of accuracy with 32-AGS was 99.64% for the Random Forest algorithm, whereas, the lowest was 66.64% with 32-AGS for the Simple Logistic classifier. Here, the classifier fails to classify some of the attacks included in the AWID. This algorithm is a function based algorithm and changing the activation function may enhance the Simple Logistic's accuracy. In addition, the Multi-layer Perceptron CS, which is a function based algorithm, achieved an accuracy of 91.21% with the 10-AGS. This is while we were able to achieve a minimum accuracy of 94.93% with the Simple Logistic classifier using 7-AGS and a higher accuracy of 99.44% with the Bagging approach using 7-AGS.

Moreover, the area under ROC curve is another good metric to compare the performance of the classifiers. Here, higher values indicate better performance. The results, as shown in

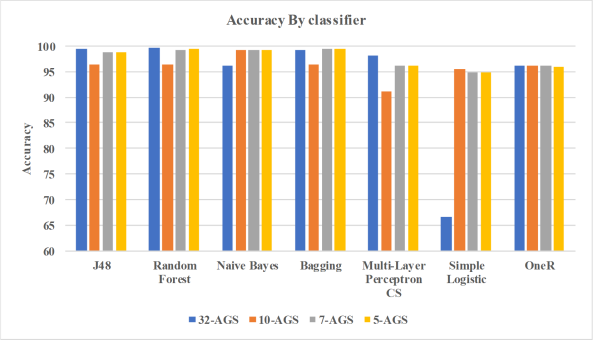

Fig. 2. Accuracy of the classifiers for different attribute sets/sizes

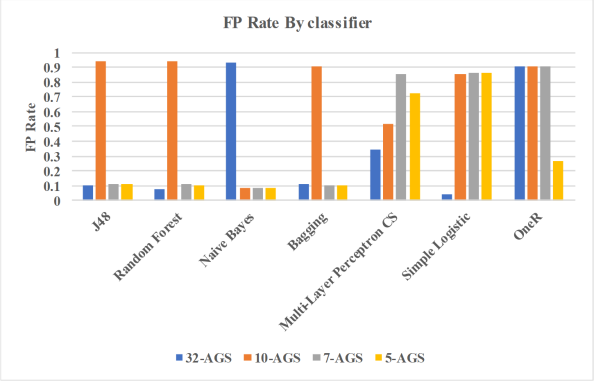

Fig. 3. FP of the classifiers for different attribute sets/sizes

Tables IV, V, VI and VII, indicate that the ROC curve area for OneR is lowest value of 0.512. Nonetheless, Random Forest shows high ROC curve area of 0.993 and 0.971 with 32-AGS and 5-AGS, respectively. Moreover, the Simple Logistic algorithm resulted in the highest ROC curve area for the 10-AGS, equal to 0.966. Figures 2 and 3 illustrate a comparison of the resulted accuracies and FP rates among the classifier models. As can be seen, 6 out of 7 classifiers achieved very good accuracy.

To further examine the 7-AGS and 5-AGS, we ran the experiments using the 10-fold cross validation approach to validate the obtained results. The cross validation results are set out in Table VIII and IX. These Tables illustrate that the maximum obtained accuracy was 99.99% using the Random Forest classifier. Furthermore, the resulted FP Rate for J48 and

TABLE IV
CLASSIFIERS EVALUATION OF AWID-ATK-R WITH 32-AGS

Classifier	Accuracy (%)	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Time
J48	99.42	0.994	0.1	0.994	0.994	0.994	0.926	0.959	0.991	469.91
OneR	96.17	0.962	0.909	0.93	0.962	0.946	0.149	0.512	0.891	11.13
Random Forest	99.64	0.966	0.077	0.995	0.966	0.995	0.956	0.993	0.998	2277.51
Naive Bayes	96.20	0.962	0.933	0.954	0.962	0.946	0.093	0.532	0.92	9.8
Bagging	99.15	0.992	0.111	0.991	0.992	0.991	0.889	0.943	0.988	1326.53
Multi-Layer Perceptron CS	98.05	0.981	0.343	0.976	0.981	0.977	0.788	0.895	0.98	17256
Simple Logistic	66.64	0.666	0.639	0.987	0.666	0.789	0.26	0.612	0.95	16830.26

TABLE V
CLASSIFIERS EVALUATION OF AWID-ATK-R WITH 10-AGS

Classifier	Accuracy (%)	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Time
J48	96.33	0.963	0.94	0.929	0.963	0.946	0.109	0.603	0.927	53.45
OneR	96.17	0.962	0.909	0.93	0.962	0.946	0.149	0.512	0.891	3.2
Random Forest	96.40	0.964	0.939	0.93	0.964	0.946	0.151	0.826	0.981	1197.08
Naive Bayes	99.14	0.991	0.083	0.989	0.991	0.99	0.882	0.927	0.989	3.24
Bagging	96.40	0.964	0.906	0.932	0.964	0.947	0.214	0.772	0.967	298.53
Multi-Layer Perceptron CS	91.21	0.912	0.514	0.944	0.912	0.928	0.296	0.886	0.956	4564.3
Simple Logistic	95.51	0.955	0.858	0.933	0.955	0.943	0.159	0.966	0.988	2049.28

TABLE VI
CLASSIFIERS EVALUATION OF AWID-ATK-R WITH 7-AGS

Classifier	Accuracy (%)	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Time
J48	98.82	0.988	0.109	0.984	0.988	0.986	0.913	0.908	0.974	37.61
OneR	96.17	0.962	0.909	0.93	0.962	0.946	0.149	0.512	0.891	2.52
Random Forest	99.31	0.993	0.107	0.993	0.993	0.992	0.921	0.971	0.991	582.63
Bagging	99.44	0.994	0.099	0.994	0.994	0.924	0.924	0.877	0.972	321.79
Multi-Layer Perceptron CS	96.09	0.963	0.858	0.932	0.963	0.947	0.272	0.88	0.948	3518.46
Simple Logistic	94.93	0.949	0.86	0.932	0.949	0.94	0.122	0.959	0.9	1501.04
Naive Bayes	99.3	0.993	0.083	0.991	0.993	0.992	0.902	0.967	0.99	2.18

TABLE VII
CLASSIFICATION EVALUATION OF AWID-ATK-R WITH 5-AGS

Classifier	Accuracy (%)	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Time
Bagging	99.44	0.994	0.099	0.994	0.994	0.994	0.924	0.877	0.972	260.2
Random Forest	99.35	0.993	0.106	0.993	0.993	0.993	0.922	0.971	0.991	538.82
Naive Bayes	99.30	0.993	0.083	0.991	0.993	0.992	0.902	0.965	0.99	1.52
J48	98.82	0.988	0.109	0.948	0.988	0.986	0.913	0.908	0.974	23.67
OneR	96.17	0.962	0.909	0.93	0.962	0.946	0.149	0.512	0.891	1.71
Multi-Layer Perceptron CS	96.09	0.961	0.727	0.936	0.961	0.948	0.415	0.934	0.957	2133.21
Simple Logistic	94.93	0.949	0.86	0.932	0.949	0.94	0.122	0.959	0.9	1516.93

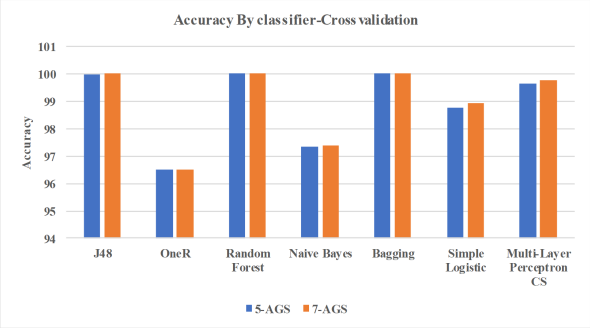

Fig. 4. Accuracy of classifiers for 7-AGS and 5-AGS using cross validation

Bagging was the lowest value of 0, which is the best result. Conversely, the OneR classifier achieved the highest FP Rate of 0.345. The Precision values range from a maximum value of 1 to a minimum value of 0.939. The training speed of the algorithms span from 0.59 seconds for Naive Bayes to 1603.38 seconds for the Multi-Layer Perceptron CS algorithm.

The following paragraph compares our experiments with previous related works that used AWID-ATK-R. Thanthrige et al. [8], [10] achieved highest accuracy of 94.97% using Random Forest with 41 attributes. In addition, when they reduced the attributes to 10, the results showed that the accuracy decreased from 94.97% to 92.29% for Random Forest. This is while our study achieved 99.99% accuracy using Random Forest with 7-AGS.

Figures 4 and 5 present a comparison of the resulted accuracies and FP rates among the classifiers in terms of attribute set size with the cross validation approach. As demonstrated from the results, the selected attributes impact positively upon our classifier models. To be specific, attributes such as epoch.time, frame.time.relative, and wlan.fc.pwrmgt are essential and effective to detect attacks such as Caffe latte, Hirte, Honeypot and Evil twin [11] . Moreover, large attribute sets may result in over-fitting problems, and optimal attribute set selection reduces the processing time, which is crucial for real-time applications.

TABLE VIII
PERFORMANCE EVALUATION OF THE CROSS VALIDATION APPROACH USING 5-AGS

Classifier	Accuracy (%)	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Time
J48	99.99	1	0	1	1	1	0.999	1	1	5.51
OneR	96.50	0.965	0.342	0.939	0.965	0.951	0.739	0.812	0.937	0.89
Random Forest	99.99	1	0	1	1	1	1	1	1	1
Naive Bayes	97.33	0.974	0.011	0.986	0.974	0.978	0.853	0.995	0.998	0.59
Bagging	99.99	1	0	1	1	1	1	1	1	298.41
Simple Logistic	98.76	0.988	0.08	0.983	0.988	0.985	0.917	0.995	0.991	1407.3
Multi-Layer Perceptron CS	99.65	0.997	0.021	0.997	0.997	0.996	0.981	0.996	0.997	1603.38

TABLE IX
PERFORMANCE EVALUATION OF THE CROSS VALIDATION APPROACH USING 7-AGS

Classifier	Accuracy (%)	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Time
Random Forest	99.99	1	0	1	1	1	1	1	1	178.53
J48	99.99	1	0	1	1	1	1	1	1	6.58
Simple Logistic	98.92	0.989	0.049	0.989	0.989	0.988	0.936	0.996	0.992	407.77
Bagging	99.99	1	0	1	1	1	1	1	1	41.81
Naive Bayes	97.37	0.974	0.11	0.986	0.974	0.978	0.853	0.995	0.998	0.72
OneR	96.49	0.965	0.345	0.939	0.956	0.951	0.739	0.812	0.937	0.83
Multi-Layer Perceptron CS	99.74	0.997	0.019	0.997	0.997	0.997	0.984	0.996	0.998	1202.98

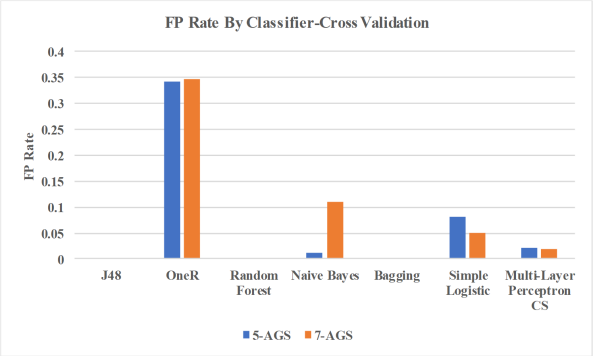

Fig. 5. FP rate of classifiers for 7-AGS and 5-AGS using cross validation

V. CONCLUSION

This paper examined various machine learning classifiers towards designing an efficient WIDS. One of the most challenging problems in this area is attribute selection since the selected attributes have an impact on the classification results. Seven well known classifier algorithms; which are J48, OneR, Naive Bayes, Random Forest, Simple Logistic, Bagging and Multi-Layer Perceptron CS; were evaluated through four different attribute sets of 32, 10,7 and 5 attributes. The results confirm that optimum attribute selection/reduction can lead to better results in terms of accuracy and processing time.

REFERENCES

[1] Kolias, Constantinos and Kambourakis, Georgios and Stavrou, Angelos and Gritzalis, Stefanos, "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, 2016.

[2] S. Sahu and B. M. Mehtre, "Network intrusion detection system using j48 decision tree," in *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on*. IEEE, 2015, pp. 2023–2026.

[3] S. Mukherjee and N. Sharma, "Intrusion detection using naive bayes classifier with feature reduction," *Procedia Technology*, vol. 4, pp. 119–128, 2012.

[4] J. Friedman, T. Hastie, R. Tibshirani *et al.*, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[5] M. E. Aminanto, H. Tanuwidjaja, P. Yoo, and K. Kim, "Weighted feature selection techniques for detecting impersonation attack in wi-fi networks," in *Proc. Symp. Cryptogr. Inf. Secur.(SCIS)*, 2017, pp. 1–8.

[6] M. E. Aminanto and K. Kim, "Detecting active attacks in wifi network by semi-supervised deep learning," in *Conference on Information Security and Cryptography 2017 Winter*, 2016.

[7] D. Kaleem and K. Ferens, "A cognitive multi-agent model to detect malicious threats," in *International Conference on Advances Applied Cognitive Computing ,2016*. CSREA, 2016, pp. 58–63.

[8] U. S. K. P. M. Thanthrige, J. Samarabandu, and X. Wang, "Machine learning techniques for intrusion detection on public dataset," in *Electrical and Computer Engineering (CCECE), 2016 IEEE Canadian Conference on*. IEEE, 2016, pp. 1–4.

[9] V. L. Thing, "Ieee 802.11 network anomaly detection and attack classification: A deep learning approach," in *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*. IEEE, 2017, pp. 1–6.

[10] U. S. K. P. M. Thanthrige, "Hidden markov model based intrusion alert prediction," Master's thesis, University of Western Ontario, 2016.

[11] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim, "Deep abstraction and weighted feature selection for wi-fi impersonation detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 621–636, March 2018.

[12] N. P. Neelakantan, C. Nagesh, and M. Tech, "Role of feature selection in intrusion detection systems for 802.11 networks," *International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN)*, vol. 1, no. 1, pp. 98–101, 2011.

[13] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, University of Waikato Hamilton, 1999.

[14] L. Xu, P. Yan, and T. Chang, "Best first strategy for feature selection," in *Pattern Recognition, 1988., 9th International Conference on*. IEEE, 1988, pp. 706–708.

[15] R. Diao and Q. Shen, "Feature selection with harmony search," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 6, pp. 1509–1523, 2012.

[16] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999, pp. 155–164.

[17] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.

[18] Y. Liu, J. Cheng, C. Yan, X. Wu, and F. Chen, "Research on the matthews correlation coefficients metrics of personalized recommendation algorithm evaluation," *Int. J. Hybrid Inf. Technol*, vol. 8, no. 1, pp. 163–172, 2015.