Hindawi Publishing Corporation
Mobile Information Systems
Volume 2015, Article ID 585432, 13 pages
http://dx.doi.org/10.1155/2015/585432

Research Article

Detecting Cyber-Attacks on Wireless Mobile Networks Using Multicriterion Fuzzy Classifier with Genetic Attribute Selection

El-Sayed M. El-Alfy[1] and Feras N. Al-Obeidat[2]

[1]*College of Computer Sciences and Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia*
[2]*IBM Research and Development Center, Markham, ON, Canada L3R 9Z7*

Correspondence should be addressed to El-Sayed M. El-Alfy; alfy@kfupm.edu.sa

Received 10 November 2014; Accepted 15 January 2015

Academic Editor: Zahoor Khan

With the proliferation of wireless and mobile network infrastructures and capabilities, a wide range of exploitable vulnerabilities emerges due to the use of multivendor and multidomain cross-network services for signaling and transport of Internet- and wireless-based data. Consequently, the rates and types of cyber-attacks have grown considerably and current security countermeasures for protecting information and communication may be no longer sufficient. In this paper, we investigate a novel methodology based on multicriterion decision making and fuzzy classification that can provide a viable second-line of defense for mitigating cyber-attacks. The proposed approach has the advantage of dealing with various types and sizes of attributes related to network traffic such as basic packet headers, content, and time. To increase the effectiveness and construct optimal models, we augmented the proposed approach with a genetic attribute selection strategy. This allows efficient and simpler models which can be replicated at various network components to cooperatively detect and report malicious behaviors. Using three datasets covering a variety of network attacks, the performance enhancements due to the proposed approach are manifested in terms of detection errors and model construction times.

1. Introduction

The number of wireless and mobile network subscribers is rapidly growing from day to day due to the flexibility of network access anywhere and anytime and the wide range of evolving capabilities that makes our lives easier. However, with these benefits a plethora of security threats also evolve as a result of the increased number of potentially exploitable vulnerabilities. The growth rate of malicious activities and botnets is jumping drastically to alarming levels according to recent security reports [1–3]. It is getting even worse for cross-network services with the emerging 4G/5G network technologies. The new era of information systems combines different environments including wireless ad hoc network, cloud computing, mobile applications, social networks, sensor networks, and smart grids [4].

There is a variety of passive and active cyber-attacks including eavesdropping or packet sniffing, attacks on wireless protocols, injection, port scanning, jamming and denial of service (DoS), fake authentication, address spoofing, session hijacking, man-in-the-middle, replay attacks, vulnerability exploits, traffic analysis, and unauthorized access [5–9].

To mitigate the anticipated risks resulting from various cyber-attacks on critical infrastructures and services, a number of algorithms and technologies have been proposed including encryption standards, digital signatures, antimalware packages, firewalls, and intrusion detection and prevention systems. These methods have been proven to be effective in securing privacy and integrity, controlling access to authorized users, and detecting malicious behaviors of known signatures. However, their performance fails to a great extent to handle sophisticated attacks, zero-day attacks, or attacks with varying signatures. A more flexible and adaptive set of approaches based on machine learning and data mining have been proposed to detect the stochastic deviation from normal behavior patterns. This category of methods is known as anomaly-based intrusion or outlier detection which provides a higher degree of automation and reduces the workload on

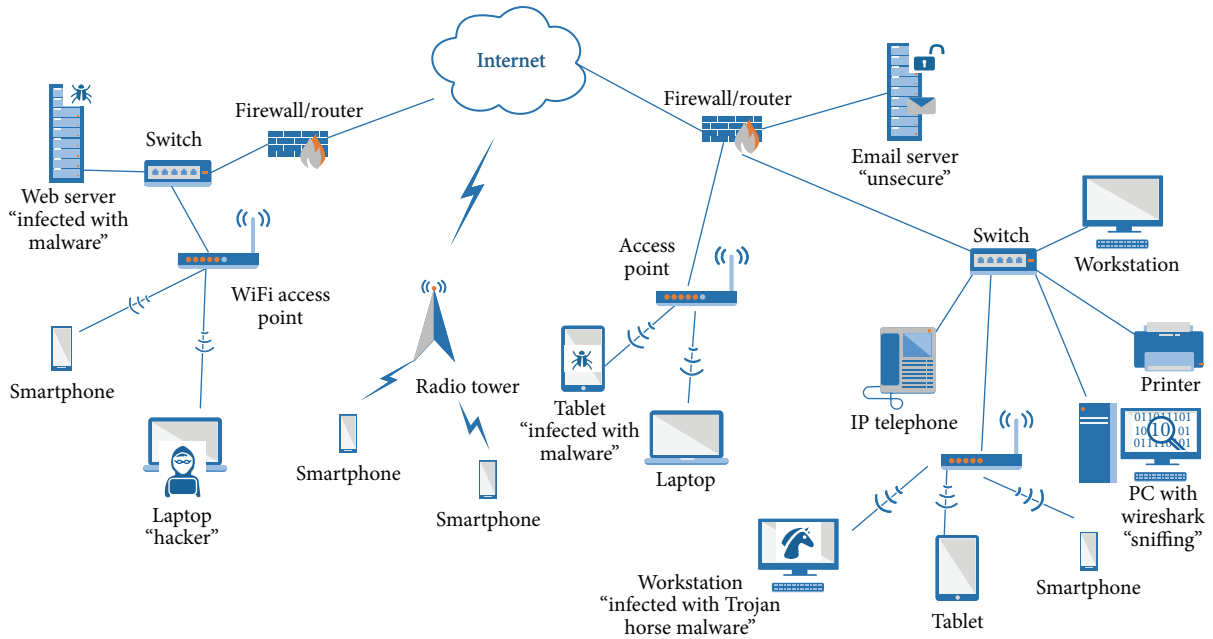

Figure 1: Illustration of a network topology with wireless and mobile devices where some devices are infected with malware or hacking.

security experts. Despite the variety of methods that have been proposed in the literature, the research on anomaly detection is still evolving to cope with uncertainties, improve the security, reduce false positive rate, and reduce computational costs [10, 11]. Additionally, since the performance to detect intrusive events is greatly influenced by type and number of attributes utilized [12], it is desirable to analyze and identify the most relevant and influential attributes from the large amount of available data.

Multicriterion decision making techniques were originally devised in the operations research field and have attracted attention of several researchers in various domains such as social psychology, business management, and health care [13, 14]. However, there is not much work done in the area of network security. In this paper, we investigate a new methodology for detecting cyber-attacks in wireless mobile networks based on multicriterion decision making fuzzy classification [15, 16]. The proposed approach is combined with an attribute selection strategy based on genetic algorithms [17]. With the minimum generalization error and the resulting simplicity and reduced computational complexity of the model, the proposed approach is practically feasible to be deployed in different network systems.

The rest of this paper is organized as follows. Section 2 gives a brief background on security in wireless and mobile information systems and Section 3 reviews related work. In Section 4, the proposed methodology is presented. Section 5 describes the adopted datasets and discusses the experimental evaluation and comparison of the proposed approach. Finally, Section 6 concludes the paper.

2. Background and Motives

In heterogeneous wireless mobile environments, there is no well-defined network perimeter; hence, the security administrator cannot enforce security policies even with the existence of firewalls and encryption. This can be attributed to its inherent nature resulting from device mobility, broadcast channels, pervasive use of multivender multidomain applications, and limited resources in wireless end-systems to implement sophisticated security countermeasures. Figure 1 illustrates a typical example of network topology where some machines are infected with malware and others are passively or actively hacking. Attackers only need to discover and exploit a single vulnerability to attack the entire system. Hence, the strength of the system security is as good as the strength of the least secure point in the system.

Wireless devices (such as smart phones, tablets, laptops, or sensors) can be communicating in an isolated environment or connected through a larger distribution network (such as a local area network, a wide area network, or the Internet) using access points. The former is called ad hoc network whereas the latter is known as infrastructure wireless network which is more common. Thus, cyber-attacks can target any of the software or hardware components in this environment including wireless end systems, wireless channels, access points, or the wired distribution network. It is highly important to detect and respond to these attacks to protect the entire system.

3. Related Work

Security of mobile information systems has been a core area in research and development. La Polla et al. in [18] surveyed

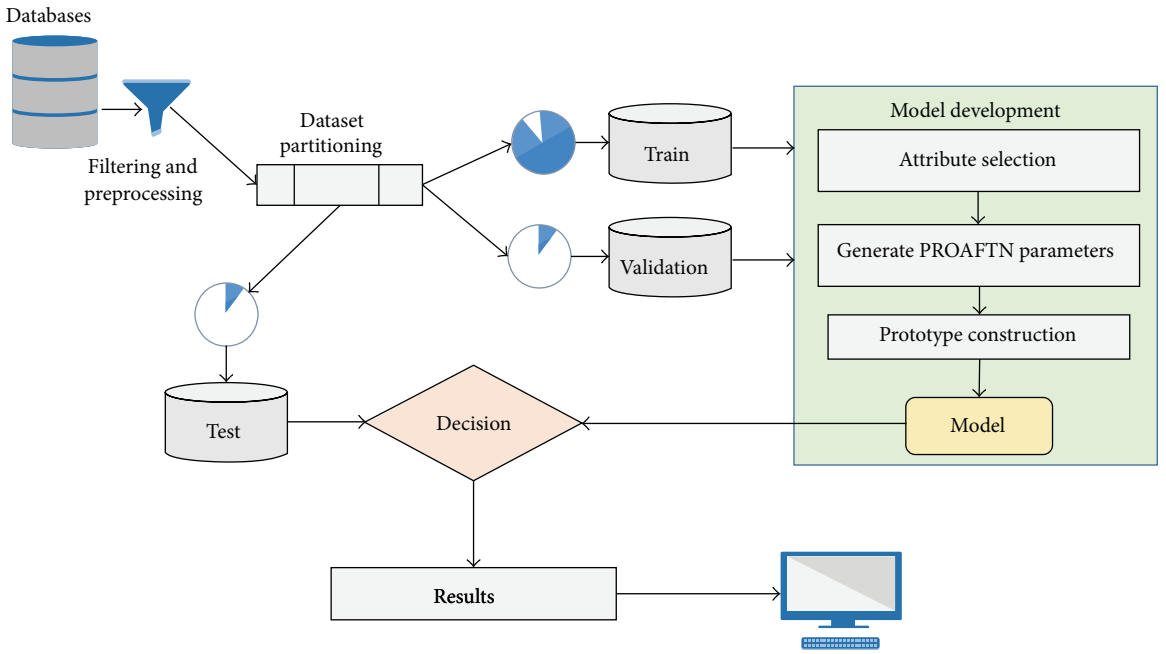

Figure 2: Block diagram for training and deploying the cyber-attack detection model.

the state of the art of high level attacks and vulnerabilities targeting mobile devices over the period from 2004 till 2011. They concisely reviewed and categorized known mobile malware including viruses, worms, rootkits, and botnets. They also discussed the proposed security solutions with focus on intrusion detection and trusted platforms. In [9], the authors reviewed the threats, vulnerabilities, and commonly available countermeasures for different components of a wireless network including clients, access points, and transmission medium.

Computational intelligence techniques have many characteristics such as adaption and fault tolerance that made them attractive for research on malware and intrusion detection. In [10], a review of 55 related studies between 2000 and 2007 is presented with focus on single, hybrid, and ensemble classifiers. Another extensive review is presented in [19]. Examples of these techniques include neural networks, fuzzy inference systems, evolutionary algorithms, artificial immune systems, and swarm intelligence. In [20], a naive Bayesian classifier is applied to identify potential intrusions. Trained on a small subset of KDD'99 dataset and tested on a larger subset, this approach showed superior identification rate. In [21], an evaluation of a number of existing machine learning classifiers is presented for dynamic Android malware detection. In [22], another approach for anomaly detection based on multicriterion fuzzy classification with greedy attribute selection is proposed and evaluated on KDD'99.

Combining security technologies can provide more solid multifaceted solutions against intrusion attempts [23]. A number of hybrid machine learning approaches have been proposed as well. For instance, in [24] a machine learning approach is introduced for classifying network activities as normal or abnormal. This approach combines support vector machines with clustering based on self-organized ant colony network. The authors demonstrated that this combination resulted in better classification rate and run time. Anomaly-based intrusion detection has attracted the interest of several researchers [10]. However, these methods can suffer from increased false positive rate. To gain advantage of misuse detection and anomaly detection, Depren et al. proposed a rule-based decision support system to combine the outcomes of decision tree for misuse detection and self-organizing map for modeling normal behavior [25].

Another important stage that can have significant impact on the accuracy and capability of intrusion detection systems is data preprocessing. A review of data preprocessing techniques for anomaly-based network intrusion detection is presented in [12]. During the preprocessing phase, various approaches can be applied such as discretization, normalization, and filtering of most relevant attributes. In [26], the impact of normalization techniques on the performance of support vector machines for intrusion detection is investigated. It has been found that min-max normalization leads to better results in terms of speed and accuracy than other normalization techniques. Another important related issue is attribute selection to reduce the high dimensionality and complexity [27].

Most of the work published in the literature is evaluated using the standard KDD Cup 99 dataset [20, 24, 26, 27]. Despite the fact that this dataset has some drawbacks, it is one of the largest datasets, covers a large number of attacks, and remains dominant to benchmark new techniques. Two more recent datasets have been recently collected and disclosed for the assessment of some attacks on IEEE 802.11 wireless channels [28].

4. Methodology

The overall block diagram for the cyber-attack detection system is shown in Figure 2. It starts with the database of

(1) i : prototype's index
(2) h: class index
(3) m: attribute's index
(4) Select *threshold* β for interval selection
(5) Generate intervals using a discretization technique
(6) Apply greedy hill climbing approach to select most relevant subsets
(7) **for** each class **do**
(8) **for** each attribute g **do**
(9) **for** every value in attribute r **do**
(10) Recursively check all values in the next attribute g_m
(11) **if** Frequency of values $\geqslant \beta$ **then**
(12) Choose intervals for prototype b_i^h
(13) **else**
(14) Discard interval and go next (i.e., $I_{g_2 h}^{r_2}$)
(15) **end if**
(16) **end for**
(17) **end for**
(18) **end for**

Algorithm 1: Composing of PROAFTN's prototypes (classification model).

captured traffic. After preprocessing and analyzing traffic records and log files, it performs feature extraction to represent each instance with a vector of relevant attributes. The dataset is then partitioned into train, validation, and test datasets. The train dataset is used to construct the detection model whereas the validation dataset is used during training to evaluate the model to avoid overfitting. The test dataset is used after training is over to evaluate the constructed model performance. The process of partitioning, training, and testing can be repeated if cross validation is required.

When datasets include attributes that are not relevant or may contain redundant attributes, this causes delay in building the classification model and accordingly degrades the classification accuracy. Hence, it is preferable to begin with selecting the most relevant attributes. In our case, we used a genetic algorithm attribute selection strategy. So, the target here is to reduce the hypothesis search space and improve the performance in terms of accuracy, scalability, and efficiency. The idea of genetic algorithms is to start with a random population of candidate solutions and then the population evolves by applying genetic operations, evaluation, and selection [17]. For attribute selection, each chromosome in the population is composed of a binary string with length equal to the total number of attributes where an attribute is selected if its corresponding bit is 1; otherwise, it is dropped. The fitness function depends on being "highly correlated with the class while having low intercorrelation" [29]. The evaluation function for a particular subset of attributes is defined mathematically as follows:

$$f(s) = \frac{k\overline{r}_{\mathrm{ca}}}{\sqrt{k + k(k-1)\,\overline{r}_{\mathrm{aa}}}}, \tag{1}$$

where k is the size of the subset s, $\overline{r}_{\mathrm{ca}}$ is the mean of attribute-class correlations, and $\overline{r}_{\mathrm{aa}}$ is the mean of the attribute-attribute correlations. This function will have lower values for attributes that are irrelevant (small value for the numerator) and/or redundant (large value for the denominator).

Once the most relevant attributes are identified, a multicriterion fuzzy classification approach is applied to construct a decision model that can assign unknown behavioral patterns to predefined classes. This type of decision problems requires a comparison between alternatives or patterns based on the scores of attributes using absolute evaluations [30]. In this case, the evaluation is performed by comparing the alternatives to different prototypes of classes, where the category or class is assigned to patterns based on the highest score value. Each prototype is described by a set of attributes and is considered to be a good representative of its class [31]. The complexity of this approach is a function of the number of attributes. Thus, utilizing the smallest subset of relevant attributes greatly improves the time complexity and accuracy of classification. A graphical illustration of the methodology is shown in Figure 3.

To explain how it works, assume the network behavioral pattern is described by a set of m attributes $\{g_1, g_2, \ldots, g_m\}$ and a label c identifying its category which belongs to the k classes $\Omega = \{C^1, C^2, \ldots, C^k\}$. Given a set of N historical patterns P, it is required to construct a classification model $f : P \rightarrow \Omega$ that can accurately predict the target class of each pattern. Once the model is built, it can be used to assign the most relevant class to new unseen behavioral patterns. The model parameters are automatically determined from the training data examples. Then, the constructed model is used for assigning a category to the unseen cases (testing data). This automatic data-driven approach is common to the learning procedures in other machine learning classifiers [32, 33]. Algorithm 1 explains the proposed induction approach through a recursive process to generate the classification model. The tree is constructed in a top-down recursive divide-and-conquer manner, where each branch represents the generated intervals for each attribute. The branches

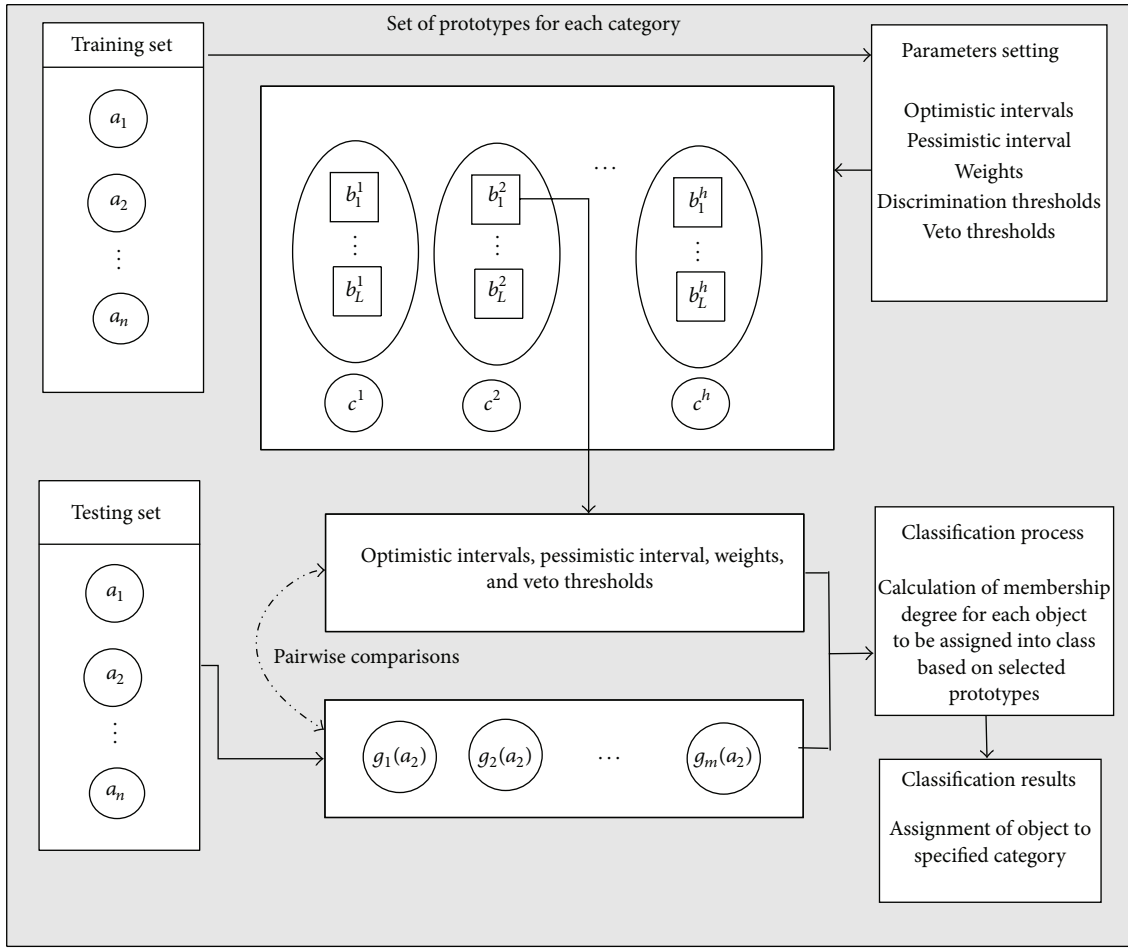

Figure 3: Graphical illustration of the multicriterion fuzzy classification procedure.

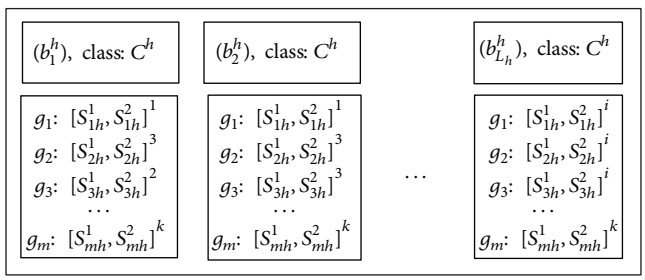

Figure 4: The prototype composition.

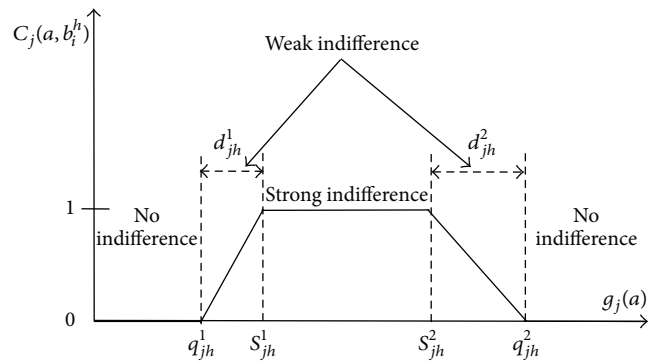

Figure 5: A typical example of the partial indifference fuzzy relation between the object a and the prototype b_i^h according to attribute g_j.

are selected recursively to compose the prototypes based on the proposed threshold. Using the generated tree from this algorithm, we can extract the prototypes and then the decision rules, respectively, to be used for classification. Figure 4 illustrates the prototypes' compositions process.

The learning strategy is based on utilizing the training set to compose a set of prototypes for each class. For class C^h, these prototypes are denoted by $B^h = \{b_1^h, b_2^h, \ldots, b_{L_h}^h\}$, where L_h is the number of prototypes for this class. For each prototype b_i^h and each attribute g_j, a fuzzy partial indifference relation $C_j(a, b_i^h)$ is defined to measure the degree of resemblance of patterns a to b_i^h according to g_j. This fuzzy relation is characterized by four parameters: the interval $[S_j^1(b_i^h), S_j^2(b_i^h)]$ where $S_j^2(b_i^h) \geq S_j^1(b_i^h)$ and the thresholds $d_j^1(b_i^h)$ and $d_j^2(b_i^h)$. Figure 5 shows a typical example of a fuzzy relation

Table 1: Some characteristics of the adopted datasets for evaluation.

Dataset	Number of traffic samples			Number of attributes	Number of attack types
	Normal	Malicious	Total		
KDD Cup 99	97278	396743	494021	41	22
WEP/WPA Dataset	15000	9200	24200	15	4
WPA2 Dataset	6000	4000	10000	16	4

with the four parameters illustrated to divide the range of values of g_j into three regions: strong indifference, weak indifference, and no indifference.

In this work, the supervised discretization technique introduced by Fayyad and Irani [34], which is based on the calculation of entropy, is utilized to generate the interval $[S_j^1(b_i^h), S_j^2(b_i^h)]$ for each class prototype and each attribute. To determine the values for $d_j^1(b_i^h)$ and $d_j^2(b_i^h)$, an adjustment/tuning is applied on $S_j^1(b_i^h)$ and $S_j^2(b_i^h)$ to allow more flexibility in assigning patterns to the closest classes. The intervals adjustment can be expressed mathematically as follows:

$$d_j^1\left(b_i^h\right) = \beta S_j^1\left(b_i^h\right), \quad d_j^2\left(b_i^h\right) = \beta S_j^2\left(b_i^h\right); \quad \beta \in [0,1]. \tag{2}$$

The prototypes in this study are constructed based on the frequency of combined values from all attributes in the dataset. After implementing the supervised discretization technique, each attribute will have a set of intervals and nominal values. The learning strategy starts from the first attribute in the list and selects the first interval or nominal value from list of values that belong to the attribute. Then, it proceeds to the next attribute and selects the first interval/nominal value and then counts the frequency of the occurrences for these combined values in each class. If the frequency exceeds the preselected threshold (e.g., more than 15%) then these values are added to the first prototype. The learning continues until all intervals and nominal values are examined by the above discussed strategy. The target is to reach all values for value-attribute from the first attribute to the last one.

To classify a pattern a to the class C^h, PROAFTN calculates the membership degree $\delta(a, C^h)$ as follows:

$$\delta\left(a, C^h\right) = \max\left\{I\left(a, b_1^h\right), I\left(a, b_2^h\right), \ldots, I\left(a, b_{L_h}^h\right)\right\}, \tag{3}$$

where $I(a, b_j^h)$ is the fuzzy indifference relation which is computed as a weighted sum of the partial indifference relations as given by

$$I\left(a, b_i^h\right) = \sum_{j=1}^{m} w_{jh} C_j\left(a, b_i^h\right), \tag{4}$$

where w_{jh} is the weight that measures the importance of a relevant attribute g_j of a specific class C^h:

$$w_{jh} \in [0,1], \quad \sum_{j=1}^{m} w_{jh} = 1. \tag{5}$$

The last step is to assign the pattern a to the class C^h that has the maximum resemblance according to the following decision rule:

$$a \in C^h \Longleftrightarrow \delta\left(a, C^h\right) = \max\left\{\delta\left(a, C^i\right) i \in \{1, \ldots, k\}\right\}. \tag{6}$$

5. Experimental Work

For the sake of evaluation of the proposed methodology, we adopted three datasets in our experimental work. Table 1 shows some of the characteristics of these datasets and more detailed description is provided in the following subsection. Then, we describe the conducted experiments and discuss the results.

5.1. Datasets Description

5.1.1. KDD Cup 99 (KDD'99) Dataset. This dataset consists of processed dump traffic portions of normal and attack connections to a local area network simulating a military network environment [35]. It was prepared from the raw dataset collected and managed by MIT Lincoln Labs as part of the 1998 DARPA Intrusion Detection Evaluation Program. Its first use was in the third International Knowledge Discovery and Data Mining Tools Competition in 1999. Since then, it has become very popular and widely used by most researchers to evaluate and benchmark their research work [20, 24, 26, 27]. The dataset has 494021 traffic samples belonging to 22 different attack types in addition to the normal traffic. These attacks fall into the following four categories: Denial of Service (DoS) such as Syn floods, unauthorized access from a remote machine (R2L) such as password guesses, unauthorized access to local root privileges (U2R) such as rootkits, and probing such as port scanning and nmap. Each connection is described with 41 attributes, as described in Table 2, and has a label identifying the traffic type to be normal or one of the attack types. Three attributes are symbolic and five attributes are binary, whereas the remaining 33 attributes are numeric. As shown in the table, these attributes are divided into four groups: basic attributes of individual connections (9 attributes), content attributes within a connection suggested by domain knowledge (13 attributes), time-based traffic attributes computed using a two-second time window (9 attributes), and host-based traffic attributes computed using a window of 100 connections to the same host (10 attributes).

5.1.2. WEP/WPA Dataset. The traffic samples in this dataset have been recently collected from a controlled wireless home network with enabled WEP/WPA [28]. The network topology

Table 2: Summary of various attributes: category, notation, name, type (numeric, categorical, and binary), statistics, and description.

Cat.	Not.	Name	Type	Statistics		Description
				Min	Max	
			Basic			
	a_1	Duration	Num.	0	58329	Connection length in seconds
	a_2	pro_type	Cat.	—	—	Prototype type which can be tcp, udp, or icmp
	a_3	srv	Cat.	—	—	Service on the destination; there are 67 potential values such as http, ftp, telnet, and domain
	a_4	Flag	Cat.	—	—	Normal or error status of the connection; there are 11 potential values, for example, rej, sh
	a_5	src_bytes	Num.	0	693 M	Num. of bytes from the source to the destination
	a_6	dst_bytes	Num.	0	52 M	Num. of bytes from the destination to the source
	a_7	Land	Binary	—	—	Whether conn. from/to same host/port or not
	a_8	wrng_frg	Num.	0	3	Number of wrong fragments
	a_9	urg	Num.	0	3	Number of urgent packets
			Content			
	a_{10}	Hot	Num.	0	30	Number of hot indicators
	a_{11}	n_failed_lgns	Num.	0	5	Number of failed login attempts
	a_{12}	logged_in	Binary	—	—	Whether successfully logged in or not
	a_{13}	n_cmprmsd	Num.	0	884	Number of compromised conditions
	a_{14}	rt_shell	Binary	—	—	Whether root shell is obtained or not
	a_{15}	su_attmptd	Num.	0	2	Number of "su root" commands attempted
	a_{16}	n_rt	Num.	0	993	Number of accesses to the root
	a_{17}	n_file_crte	Num.	0	28	Number of create-file operations
	a_{18}	n_shells	Num.	0	2	Number of shell prompts
	a_{19}	n_access_files	Num.	0	8	Number of operations on access control files
	a_{20}	n_obnd_cmds	Num.	0	0	Number of outbound commands in an ftp session
	a_{21}	is_hot_lgn	Binary	—	—	Whether login belongs to hot list or not
	a_{22}	is_guest_lgn	Binary	—	—	Whether login is guest or not
			t_traffic (using a window of 2 seconds)			
	a_{23}	cnt	Num.	0	511	Number of same-host connections as the current connection in the past 2 seconds
	a_{24}	srv_cnt	Num.	0	511	Num. of same-host conn. to the same service as the current connection in the past 2 seconds
	a_{25}	syn_err	Num.	0	1	Percentage of same-host conn. with syn errors
	a_{26}	srv_syn_err	Num.	0	1	Percentage of same-service conn. with syn errors
	a_{27}	rej_err	Num.	0	1	Percentage of same-host conn. with rej errors
	a_{28}	srv_rej_err	Num.	0	1	Percentage of same-service conn. with rej errors
	a_{29}	sm_srv_r	Num.	0	1	Percentage of same-host conn. to same service
	a_{30}	dff_srv_r	Num.	0	1	Percentage of same-host conn. to different services
	a_{31}	srv_dff_hst_r	Num.	0	1	Percentage of same-service conn. to different hosts
			h_traffic (using a window of 100 connections)			
	a_{32}	h_cnt	Num.	0	255	Number of same-host connections as the current connection in the past 100 connections
	a_{33}	h_srv_cnt	Num.	0	255	Num. of same-host conn. to the same service as the current connection in the past 100 connections
	a_{34}	h_sm_srv_r	Num.	0	1	Percentage of same-host conn. to same service
	a_{35}	h_dff_srv_r	Num.	0	1	Percentage of same-host conn. to different services
	a_{36}	h_sm_sr_prt_r	Num.	0	1	Percentage of same-service conn. to different hosts
	a_{37}	h_srv_dff_hst_r	Num.	0	1	Percentage of same-service conn. to different hosts
	a_{38}	h_syn_err	Num.	0	1	Percentage of same-host conn. with syn errors
	a_{39}	h_srv_syn_err	Num.	0	1	Percentage of same-service conn. with syn errors
	a_{40}	h_rej_err	Num.	0	1	Percentage of same-host conn. with rej errors
	a_{41}	h_srv_rej_err	Num.	0	1	Percentage of same-service conn. with rej errors

Table 3: Comparisons of accuracy for different approaches using 10-fold cross validation (results are approximated to two decimal digits). All model constructions have taken reasonable time except SVM and MLP.

Approach	KDD'99 dataset		WEP/WPA dataset		WPA2 dataset	
	Acc (%)	Time (sec)	Acc (%)	Time (sec)	Acc (%)	Time (sec)
With attribute selection						
Proposed	99.92	7	85.70	6	90.10	4
NB	94.57	3	77.38	2	68.82	2
SVM	97.61	27	83.23	21	80.24	19
MLP	96.15	31	84.32	29	88.46	23
Without attribute selection						
Proposed	99.20	10	84.89	8	91.82	6
NB	93.28	3.8	77.24	4	76.41	3
SVM	97.58	33	83.02	28	83.26	22
MLP	96.24	48	78.73	34	90.44	29

Table 4: The KDD'99 per-class performance of the proposed method with and without attribute selection (approximated to three decimal digits).

Normal/attack	Count	With attribute selection				Without attribute selection			
		Precision	Recall	F_1	AUC	Precision	Recall	F_1	AUC
Normal	97278	0.999	0.999	0.999	1	0.990	0.989	0.989	0.990
Back	2203	1	0.999	0.999	0.999	0.987	0.987	0.987	0.989
Buffer_overflow	30	1	0.692	0.818	0.846	0.723	0.678	0.700	0.848
ftp_write	8	0.5	0.5	0.5	0.75	0	0.000	0.000	0.573
Guess_passwd	53	0.909	0.952	0.93	0.976	0.933	0.933	0.933	0.962
imap	12	1	0.4	0.571	0.7	0.157	0.240	0.190	0.670
ipsweep	1247	0.892	0.988	0.938	0.994	0.985	0.983	0.984	0.989
Land	21	0.8	1	0.889	1	0.847	0.937	0.890	0.919
Loadmodule	9	0.333	0.2	0.25	0.6	0	0.000	0.000	0.766
Multihop	7	0.25	0.333	0.286	0.667	0.276	0.323	0.298	0.847
Neptune	107201	1	1	1	1	0.990	0.990	0.990	0.990
nmap	231	0.906	0.358	0.513	0.679	0.938	0.981	0.959	0.981
Perl	3	0.5	0.5	0.5	0.75	0.323	0.990	0.490	0.980
phf	4	0.25	1	0.4	1	0.990	0.657	0.790	0.990
pod	264	0.986	0.973	0.98	0.987	0.990	0.986	0.988	0.990
Portsweep	1040	0.981	0.976	0.979	0.988	0.977	0.982	0.979	0.987
Rootkit	10	0.5	0.333	0.4	0.667	0	0.000	0.000	0.662
Satan	1589	0.986	0.989	0.988	0.995	0.981	0.984	0.982	0.987
Smurf	280790	1	1	1	1	0.990	0.990	0.990	0.990
Spy	2	1	1	1	1	0	0.000	0.000	0.443
Teardrop	979	0.994	0.997	0.996	0.999	0.989	0.990	0.989	0.989
Warezclient	1020	0.997	0.982	0.99	0.991	0.968	0.982	0.975	0.988
Warezmaster	20	0.333	0.5	0.4	0.75	0.790	0.752	0.770	0.910

is a single basic service set (BSS) consisting of one access point (AP) connected to the Internet and three stations: one generating real HTTP and FTP traffic (STA1), one running Wireshark to monitor the network and capture traffic (STA2), and one for generating attacks (STA3). In addition to normal traffic, four types of attacks are reported: ChopChop, deauthentication, duration, and fragmentation. There are a total of 24200 traffic samples; 15000 of them belong to normal traffic whereas the rest are divided equally for each attack type. The captured traffic from normal and attack processes is preprocessed using Tshark to extract 15 attributes from the MAC headers.

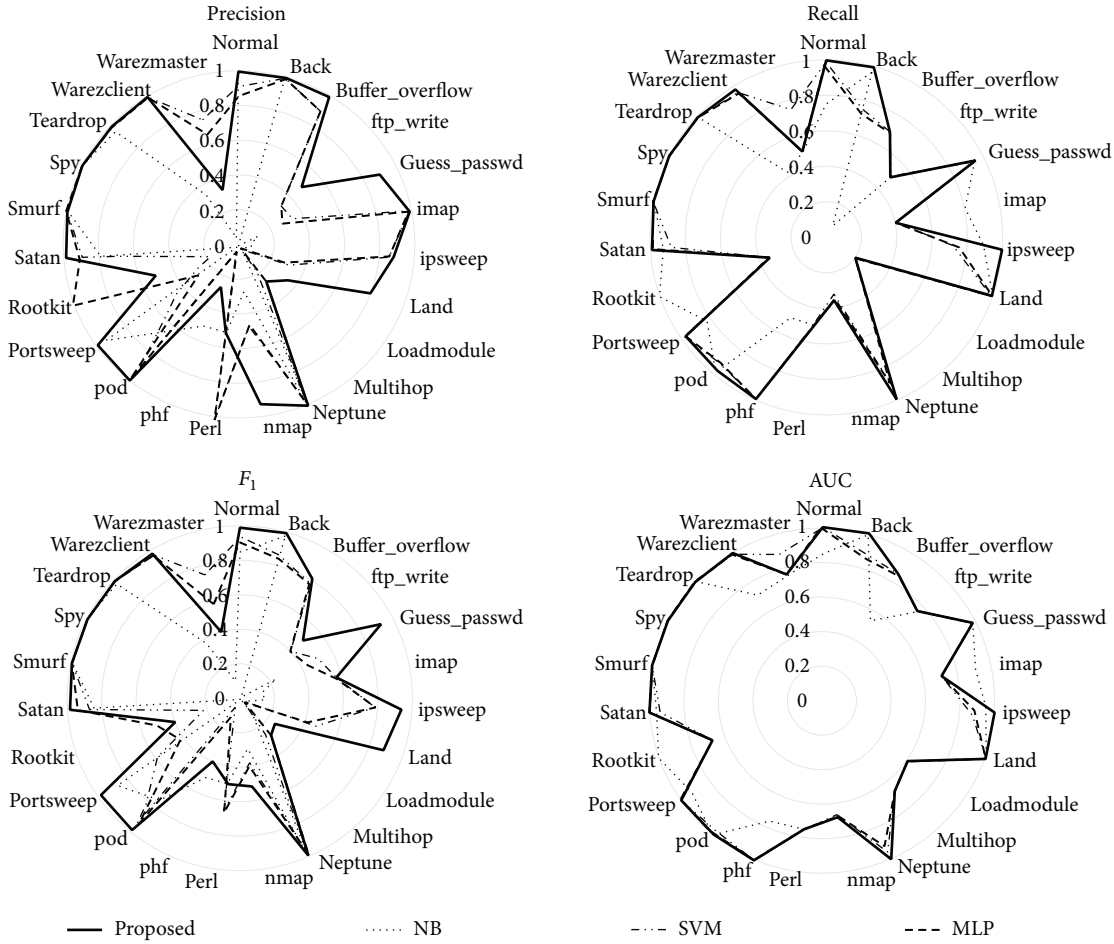

Figure 6: Comparing the per-class results for KDD'99 dataset using the reduced attribute vector (due to attribute selection) with various methods in terms of precision, recall, F_1 measure, and AUC.

5.1.3. WPA2 Dataset. The third dataset has been collected from a corporate network with enabled WPA2 encryption [28]. In this network, there are two access points connected to a local area network switch, which is connected to an authentication server (AS) and the Internet. In this scenario, there are five stations: three generating traffic, one monitoring the network, and one hacking. Here, there are four attack types: deauthentication, fake authentication, fake AP, and Syn flooding. The total number of traffic samples is 10000, where 6000 of them belong to normal traffic and the rest are distributed equally for each attack type. Each sample is processed as in the second dataset with Tshark and described with 16 attributes.

5.2. Performance Measures. We used 10-fold cross validation to evaluate and compare the performance of the proposed methodology. The performance is reported in terms of accuracy (Acc), recall (true positive rate), precision, and F_1 measure. These measures are computed as follows:

$$\text{Acc} = \frac{(\text{tp} + \text{tn})}{(\text{tp} + \text{tn} + \text{fp} + \text{fn})},$$

$$\text{Recall} = \frac{\text{tp}}{(\text{tp} + \text{fn})},$$

$$\text{Precision} = \frac{\text{tp}}{(\text{tp} + \text{fp})},$$

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{(\text{precision} + \text{recall})}, \tag{7}$$

where tp refers to true positive, tn refers to true negative, fp refers to false positive, and fn refers to false negative. We also compared the area under the receiver operating characteristic (ROC) curve (AUC) and the time to construct the attack detection model.

5.3. Experiments and Results. The proposed methodology was implemented in Java and ran in a Linux machine. We applied it to the datasets described above with and without attribute selection. For the first dataset, KDD'99, the application of the attribute selection strategy has resulted in only 17 out of the 41 attributes as relevant attributes. Referring to Table 2, the selected attributes are a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_{10},

Table 5: The WEP/WPA per-class performance of the proposed method with and without attribute selection (approximated to three decimal digits).

Normal/attack	Count	With attribute selection				Without attribute selection			
		Precision	Recall	F_1	AUC	Precision	Recall	F_1	AUC
Normal	15000	0.822	1	0.902	1	0.817	0.996	0.898	0.993
ChopChop	2300	1	0.326	0.491	0.627	0.855	0.130	0.226	0.521
Deauthentication	2300	0.945	0.971	0.958	0.984	0.938	0.981	0.959	0.989
Duration	2300	0.970	0.997	0.983	0.999	0.966	0.986	0.976	0.992
Fragmentation	2300	0.994	0.21	0.347	0.563	0.968	0.338	0.50	0.632

Table 6: The WPA2 per-class performance of the proposed method with and without attribute selection (approximated to three decimal digits).

Normal/attack	Count	With attribute selection				Without attribute selection			
		Precision	Recall	F_1	AUC	Precision	Recall	F_1	AUC
Normal	6000	0.906	0.935	0.920	0.916	0.906	0.970	0.937	0.961
Fake AP	1000	0.998	0.952	0.974	0.973	0.985	0.945	0.965	0.969
Fake authentication	1000	0.734	0.483	0.582	0.713	0.934	0.448	0.605	0.694
Deauthentication	1000	0.984	0.967	0.975	0.981	0.981	0.978	0.980	0988
Syn flooding	1000	0.822	0.997	0.901	0.998	0.874	0.997	0.931	0.998

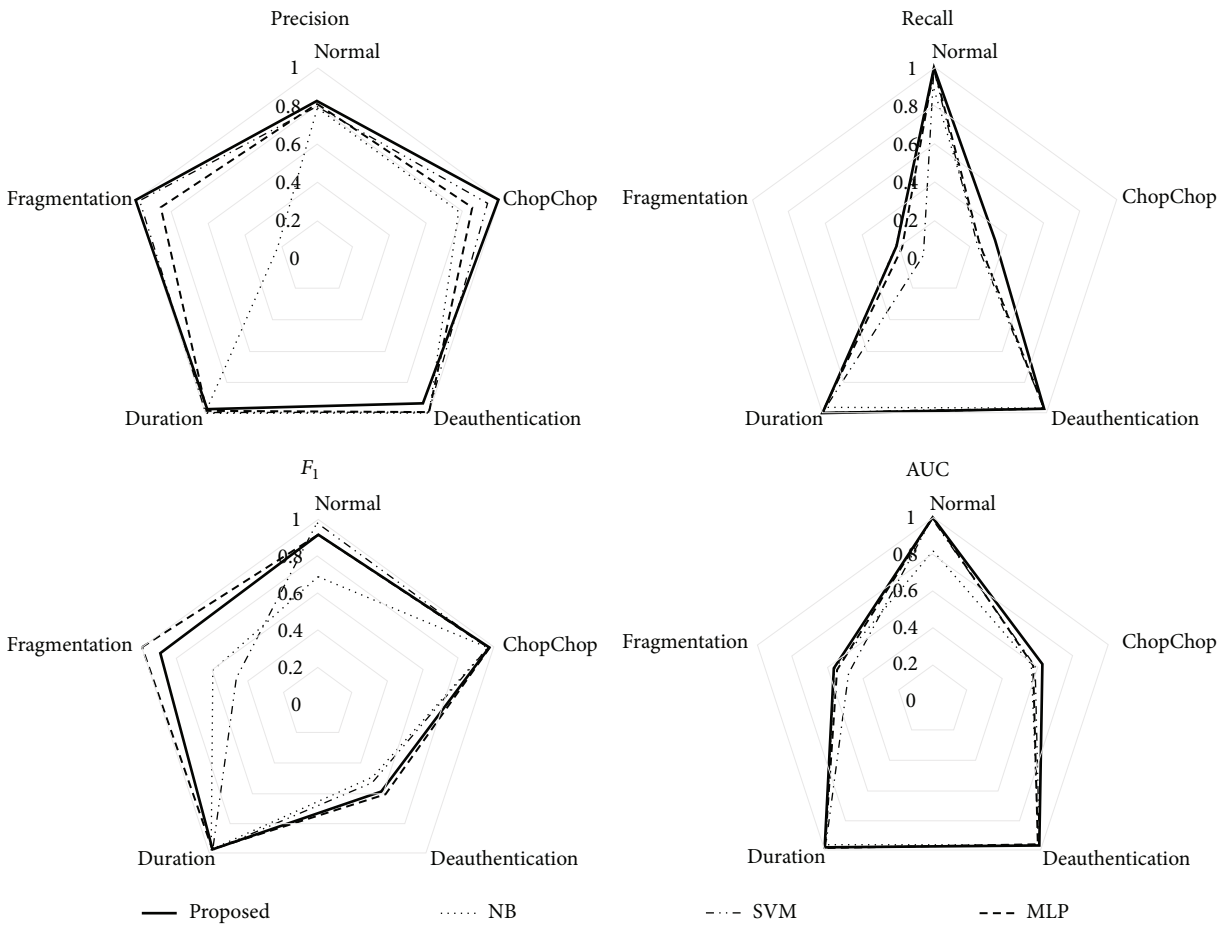

Figure 7: Comparing the per-class results for WEP/WPA dataset using the reduced attribute vector (due to attribute selection) with various methods in terms of precision, recall, F_1 measure, and AUC.

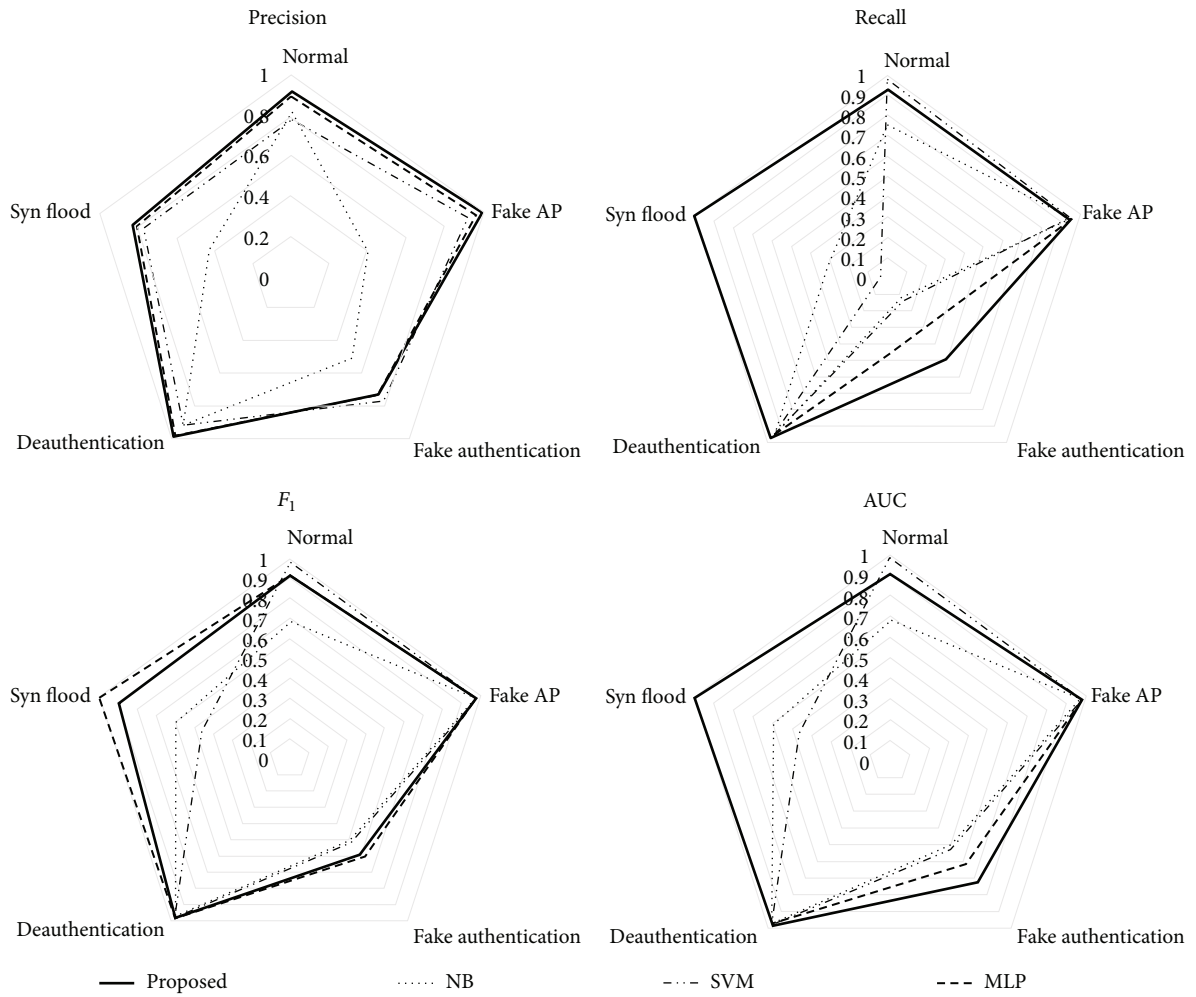

Figure 8: Comparing the per-class results for WPA2 dataset using the reduced attribute vector (due to attribute selection) with various methods in terms of precision, recall, F_1 measure, and AUC.

a_{12}, a_{19}, a_{23}, a_{29}, a_{30}, a_{31}, a_{33}, a_{34}, and a_{38}. For WEP/WPA dataset, only 7 attributes were selected whereas for the WPA2 dataset, only 5 attributes were selected.

We conducted a comparative study with three popular machine learning algorithms implemented in [36] with default settings using the stratified 10-fold cross validation. Table 3 summarizes the performance of the proposed method with and without attribute selection and compared it to the other classifiers: naive Bayes (NB), support vector machine (SVM), and multilayer perceptron (MLP). The reported time is the model construction time (in other words, it does not include the time for attribute selection). This table shows consistent results for the three considered datasets. All model constructions have taken reasonable times except for SVM and MLP. Although NB can take slightly less time than the proposed method, its accuracy is much lower. This demonstrates that the proposed methodology can outperform other techniques with improved accuracy and simpler models even with few selected attributes. In general, we observed that the performance for the KDD'99 dataset is much better than for the other datasets. This can be due to the size and nature of the dataset since KDD'99 has more samples and attributes covering larger parts of the search space.

For the proposed methodology, we also reported the performance for each class in the three datasets in terms of precision, recall, F_1 measure, and AUC. These results are shown in Tables 4, 5, and 6. For the first dataset, KDD'99, the distribution of traffic samples is skewed where some attacks are very rare. We can notice that the proposed methodology is very accurate when enough samples exist. For the other two datasets, the performance is very high except for two attack types. This can be attributed to incomplete attribute set to distinguish between all traffic types. The comparisons of the per-class performance with other methods are shown in Figures 6, 7, and 8. In these figures, it is desirable to cover larger area of the shape in each direction (class type). Similar conclusion can be drawn as above, where the proposed methodology is promising and can be effective for cyber-attack detection.

6. Conclusion

This paper presents a novel security mechanism for cyber-attack detection in wireless mobile networks. It uses historical data to build detection models with the most influential attributes. The proposed hybrid methodology is based on multicriterion fuzzy classification augmented with a meta-heuristic approach using a genetic algorithm for attribute selection strategy. The constructed predictive model is then deployed to classify unknown incoming traffic. After capturing, preprocessing, and analyzing traffic, the relevant attributes are then extracted and integrated with the model to decide whether the activity is normal or malicious. Three datasets with various natures and different cyber-attacks are utilized to evaluate and compare the effectiveness of the proposed methodology to detect cyber-attacks on different components of a mobile wireless network. Results showed that the proposed methodology behaved consistently for all datasets with promising detection accuracies and model construction times. In some attacks, the performance was relatively low. However, this can be due to the insufficient number of captured samples, imbalanced distribution of the dataset, or insufficient extracted attributes from the raw traffic. As future work, it is intended to explore more attacks and other datasets and subsequently improve our methodology further.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The first author would like to acknowledge the support provided by King Abdulaziz City for Science and Technology (KACST) through the Science & Technology Unit at King Fahd University of Petroleum & Minerals (KFUPM) for funding this work through Project no. 11-INF1658-04 as part of the National Science, Technology, and Innovation Plan.

References

[1] Cisco, *Cisco 2014 Annual Security Report*, 2014, https://www.cisco.com/web/offer/gist_ty2_asset/Cisco_2014_ASR.pdf.

[2] Sophos Security Threat Report 2014, http://www.sophos.com/en us/medialibrary/PDFs/other/sophos-security-threat-report-2014.pdf.

[3] *Mobile Security Review, AV-Comparatives*, 2014, http://www.av-comparatives.org/wp-content/uploads/2014/09/avc_mob_201407_en.pdf.

[4] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1294–1313, 2013.

[5] I. M. Chapman, S. P. Leblanc, and A. Partington, "Taxonomy of cyber attacks and simulation of their effects," in *Proceedings of the Military Modeling & Simulation Symposium*, pp. 73–80, Society for Computer Simulation International, 2011.

[6] G. Lehembre, "Wi-Fi security—WEP, WPA and WPA2," *Hackin9*, vol. 6, 2005.

[7] E. Tews and M. Beck, "Practical attacks against WEP and WPA," in *Proceedings of the 2nd ACM Conference on Wireless Network Security (WiSec '09)*, pp. 79–85, March 2009.

[8] A. H. Lashkari, M. M. S. Danesh, and B. Samadi, "A survey on wireless security protocols (WEP, WPA and WPA2/802.11i)," in *Proceedings of the IEEE International Conference on Computer Science and Information Technology*, pp. 48–52, Beijing, China, August 2009.

[9] M.-K. Choi, R. J. Robles, C.-H. Hong, and T.-H. Kim, "Wireless network security: vulnerabilities, threats and countermeasures," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 3, no. 3, pp. 77–86, 2008.

[10] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: a review," *Expert Systems with Applications*, vol. 36, no. 10, pp. 11994–12000, 2009.

[11] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1-2, pp. 18–28, 2009.

[12] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: a review," *Computers & Security*, vol. 30, no. 6-7, pp. 353–375, 2011.

[13] N. E. Fenton and W. Wang, "Risk and confidence analysis for fuzzy multicriteria decision making," *Knowledge-Based Systems*, vol. 19, no. 6, pp. 430–437, 2006.

[14] C. Zopounidis and M. Doumpos, "Multicriteria classification and sorting methods: a literature review," *European Journal of Operational Research*, vol. 138, no. 2, pp. 229–246, 2002.

[15] F. Al-Obeidat, N. Belacel, J. A. Carretero, and P. Mahanti, "An evolutionary framework using particle swarm optimization for classification method proaftn," *Applied Soft Computing Journal*, vol. 11, no. 8, pp. 4971–4980, 2011.

[16] N. Belacel, "Multicriteria assignment method PRO AFTN: methodology and medical application," *European Journal of Operational Research*, vol. 125, no. 1, pp. 175–183, 2000.

[17] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2-3, pp. 95–99, 1998.

[18] M. La Polla, F. Martinelli, and D. Sgandurra, "A survey on security for mobile devices," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 446–471, 2013.

[19] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: a review," *Applied Soft Computing Journal*, vol. 10, no. 1, pp. 1–35, 2010.

[20] H. Altwaijry and S. Algarny, "Bayesian based intrusion detection system," *Journal of King Saud University—Computer and Information Sciences*, vol. 24, no. 1, pp. 1–6, 2012.

[21] B. Amos, H. Turner, and J. White, "Applying machine learning classifiers to dynamic android malware detection at scale," in *Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC '13)*, pp. 1666–1671, Sardinia, Italy, July 2013.

[22] E.-S. M. El-Alfy and F. N. Al-Obeidat, "A multicriterion fuzzy classification method with greedy attribute selection for anomaly-based intrusion detection," *Procedia Computer Science*, vol. 34, pp. 55–62, 2014.

[23] K. Satpute, S. Agrawal, J. Agrawal, and S. Sharma, "A survey on anomaly detection in network intrusion detection system using particle swarm optimization based machine learning techniques," in *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA '13)*, pp. 441–452, Springer, 2013.

[24] W. Feng, Q. Zhang, G. Hu, and J. X. Huang, "Mining network data for intrusion detection through combining SVMs with ant colony networks," *Future Generation Computer Systems*, vol. 37, pp. 127–140, 2014.

[25] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks," *Expert Systems with Applications*, vol. 29, no. 4, pp. 713–722, 2005.

[26] W. Li and Z. Liu, "A method of SVM with normalization in intrusion detection," *Procedia Environmental Sciences*, vol. 11, pp. 256–262, 2011.

[27] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "Feature selection and classification in multiple class datasets: an application to KDD Cup 99 dataset," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5947–5957, 2011.

[28] D. W. F. L. Vilela, E. T. Ferreira, A. A. Shinoda, N. V. de Souza Araujo, R. de Oliveira, and V. E. Nascimento, "A dataset for evaluating intrusion detection systems in ieee 802.11 wireless networks," in *Proceedings of the IEEE Colombian Conference on Communications and Computing (COLCOM '14)*, pp. 1–5, Bogotá, Colombia, June 2014.

[29] M. A. Hall, *Correlation-based feature selection for machine learning [Ph.D. thesis]*, The University of Waikato, 1999.

[30] J. Léger and J.-M. Martel, "A multi-criteria assignment procedure for a nominal sorting problematic," *European Journal of Operational Research*, vol. 138, no. 2, pp. 349–364, 2002.

[31] K. Jabeur and A. Guitouni, "A generalized framework for concordance/discordance-based multi-criteria classification methods," in *Proceedings of the 10th International Conference on Information Fusion*, pp. 1–8, July 2007.

[32] P. W. Baim, "A method for attribute selection in inductive learning systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 888–896, 1988.

[33] D. M. Dutton and G. V. Conroy, "A review of machine learning," *The Knowledge Engineering Review*, vol. 12, no. 4, pp. 341–367, 1997.

[34] U. Fayyad and K. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI '93)*, pp. 1022–1029, 1993.

[35] KDD Cup 1999 dataset for network-based intrusion detection systems, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[36] H. Witten, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, 2005.