Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 34 (2014) 55 – 62

Procedia Computer Science

The 9th International Conference on Future Networks and Communications (FNC-2014)

A multicriterion fuzzy classification method with greedy attribute selection for anomaly-based intrusion detection

El-Sayed M. El-Alfy[a,*], Feras N. Al-Obeidat[b]

[a] *College of Computer Sciences and Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia*
[b] *Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada*

Abstract

Intrusion is widely recognized as a chronic and recurring problem of computer systems' security with the continual changes and increasing volume of hacking techniques. This paper explores a new countermeasure approach for anomaly-based intrusion detection using a multicriterion fuzzy classification method combined with a greedy attribute selection. The proposed approach has the advantage of dealing with various types of attributes including network traffic basic TCP/IP packet headers, as well as content-based, time-based and host-based attributes. At the same time, to reduce the dimensionality and increase the computational efficiency, the greedy attribute selection algorithm enables it to choose an optimal subset of attributes that is most relevant for detecting intrusive events. The simplicity of the constructed model allows it to be replicated at various network components in emerging open system infrastructures such as sensor networks, wireless ad hoc networks, cloud computing, and smart grids. The proposed approach is evaluated and compared on a commonly-used intrusion detection benchmark dataset. The results show more than 99.9% overall accuracy with high detection rates for various types of intrusions can be achieved with about 26% only of the available attributes.

Selection and peer-review under responsibility of Conference Program Chairs
Keywords: intrusion detection, machine learning, multicriterion fuzzy classification, attribute selection, dimensionality reduction.

1. Introduction

Numerous types of cyber attacks are launched against computer systems such as port scanning, traffic sniffing, denial of service, address spoofing, session hijacking, vulnerability exploits, unauthorized access and privilege escalation. The growth rate of intrusion is ever increasing to alarming levels as mentioned in recent security reports[1,2]. This problem is getting worse with the emerging network technologies and environments such as sensor networks, smart grids, wireless ad hoc network, cloud computing, mobile applications, and social networks. To provide reasonable protection for such critical infrastructures, a number of signature-based solutions and technologies are often deployed to detect misuse patterns and control access including ant-viruses, anti-spywares and packet-based firewall filters. These methods have proven their effectiveness in detecting attacks of known signatures but they fail to deal with zero-day attacks, attacks with slightly varying signatures, or sophisticated attacks. A more flexible and adaptive

* Corresponding author. Tel.: +966-13-860-1930.
E-mail address: alfy@kfupm.edu.sa

Selection and peer-review under responsibility of Conference Program Chairs
doi:10.1016/j.procs.2014.07.037

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 detection which provides a higher degree of automation and reduces the workload on 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[3,4]. Additionally, since the performance to detect intrusive events is greatly influenced by type and number of attributes utilized[5], it is desirable to consider various attributes during the model construction phase.

Multicriterion decision making techniques were originally devised in the operations research field and have attracted attention of several researchers in domains such as social psychology, business management, and health care[6,7]. However, there is not much work done in the area of network security. In this paper, we investigate a new methodology for anomaly-based intrusion detection based on multicriterion decision making fuzzy classification, known as PROAFTN[8,9], combined with a greedy hill-climbing search for attribute selection. With the minimum generalization error and the resulting simplicity and reduced computational complexity of the model, the proposed approach is practically feasible to be applied in the domain of network intrusion detection.

The rest of this paper is organized as follows. Section 2 briefly reviews related work. In Section 3, the proposed methodology is introduced. A description of the dataset and a discussion of the experimental work are provided in Section 4. Finally, conclusions are drawn in Section 5.

2. Related Work

Computational intelligence techniques have many characteristics such as adaption and fault tolerance that made them attractive for research on intrusion detection. In[3], 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[10]. Examples of these techniques include neural networks, fuzzy inference systems, evolutionary algorithms, artificial immune systems, and swarm intelligence. In[11], 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. A number of hybrid machine learning approaches have been proposed as well. For instance, in[12] 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[3]. 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[13].

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[5]. Data preprocessing covers various approaches such as normalization and selection of most relevant attributes. The impact of data normalization on the performance of support vector machines for intrusion detection is investigated in[14]. 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[15]. Most of the work published in the literature is evaluated using the standard KDD Cup 99 dataset[12,14,11,15]. Despite some critiques against the aging of this dataset, it remains a benchmark dataset especially for evaluating new approaches.

3. Methodology

The proposed methodology for anomaly-based intrusion detection consists of two major steps as explained in the following subsections.

3.1. Relevant Attribute Selection

When datasets include attributes that are not relevant or may contain redundant attributes, this causes delay in building the classification model and accordingly degrade the classification accuracy. Hence, we start with an attribute

subset selection approach to enhance the performance of the multicriterion fuzzy classification method. So, the target here is to reduce the hypothesis search space and improve the performance in terms of accuracy, scalability and efficiency.

Without loss of generality, we adopted a forward attribute selector that uses a correlation based heuristic to determine the usefulness of attributes, and evaluates its effectiveness with the PROAFTN method. The application of this approach as a preprocessing step allows selection of relevant attributes in reasonable time. The search for the best subset of attributes is conducted using a greedy hill-climbing augmented with a backtracking facility.

Selection of attributes is based on the hypothesis made by Hall [16] that "subsets of features that are highly correlated with the class while having low inter-correlation are preferred". It starts with an empty set of attributes and iteratively evaluates all single attribute additions and adds the attribute with the highest merit (i.e. highest predictive power and smallest degree of redundancy with other attributes in the set). The evaluation function for a particular subset of attributes is defined mathematically as follows [16]:

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

where k is the size of the subset s, $\bar{r}_{ca}$ is the mean of attribute-class correlations, and $\bar{r}_{aa}$ 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).

3.2. Multicriterion Fuzzy Classification

The anomaly-based intrusion detection problem can be solved by a multicriterion fuzzy classification approach which assigns 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 [17]. 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 [18]. 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.

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 [19,20]. The main steps of the classifier construction are outlined in Algorithm 1. 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 as $B^h = \{b_1^h, b_2^h, \ldots, b_{L_h}^h\}$ 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 the pattern 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 1 shows a typical example of a fuzzy relation 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 [21], which is based 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:

$$d_j^1(b_i^h) = \beta S_j^1(b_i^h), \text{and} \quad d_j^2(b_i^h) = \beta S_j^2(b_i^h); \;\; \beta \in [0, 1]$$

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

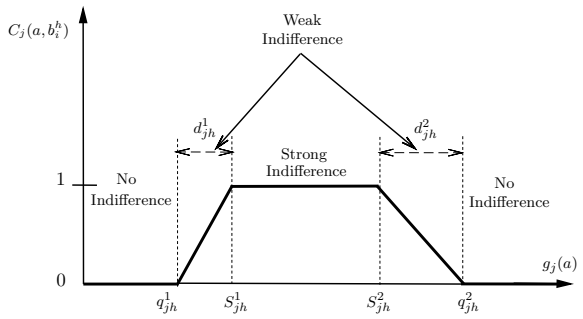

Fig. 1. A typical example of the partial indifference fuzzy relation between the object a and the prototype b_i^h according to attribute g_j.

nominal values. The learning strategy starts from the first attribute in the list and select 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 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.

Algorithm 1 Composing PROAFTN's prototypes (classification model)

```
i : prototype's index
h : class index
m : attribute's index
Select threshold β for interval selection
Generate intervals using a discretization technique
Apply greedy hill climbing approach to select most relevant subsets
for each class do
    for each attribute g do
        for every value in attribute r do
            Recursively check all values in the next attribute g_m
            if Frequency of values ⩾ β then
                Choose intervals for prototype b_i^h
            else
                Discard interval and go next (i.e., I_{g_2 h}^{r_2})
            end if
        end for
    end for
end for
```

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

$$\delta(a, C^h) = \max\{I(a, b_1^h), I(a, b_2^h), ..., I(a, b_{L_h}^h)\} \tag{2}$$

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:

$$I(a, b_i^h) = \sum_{j=1}^{m} w_{jh} C_j(a, b_i^h) \tag{3}$$

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

$$w_{jh} \in [0,1], \text{ and } \sum_{j=1}^{m} w_{jh} = 1 \tag{4}$$

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:

$$a \in C^h \Leftrightarrow \delta(a, C^h) = \max\{\delta(a, C^i) | i \in \{1, ..., k\}\} \tag{5}$$

4. Experimental Work

4.1. Dataset Description

The dataset used in our experimental work is adopted from the KDD Cup 99 (KDD'99) dataset[22]. This dataset is an adapted version of the dataset prepared and managed by MIT Lincoln Labs as part of the 1998 DARPA Intrusion Detection Evaluation Program. KDD'99 was first used for the third International Knowledge Discovery and Data Mining Tools Competition in 1999. Since then, KDD'99 became a dominant intrusion detection dataset which has been and still widely used by most researchers to evaluate and benchmark their related work[12,14,11,15]. The dataset consists of processed TCP dump portions of normal and attack connections to a local area network simulating a military network environment. There are 22 attacks in the dataset falling into four main categories, namely: denial of service (DoS) such as syn flood, unauthorized access from a remote machine (R2L) such as password guess, unauthorized access to local root privileges (U2R) such as rootkit, and probing such as port scan and nmap.

The dataset has 494021 connections; each connection is described with 41 attributes and has a label identifying the type as either normal or one of the attacks. Three attributes are symbolic, five attributes are binary, whereas the remaining 33 attributes are numeric. The 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). A summary of these attributes is provided in Table 1.

4.2. Performance Measures

The proposed method for anomaly-based intrusion detection is evaluated and compared with other approaches using stratified 10-fold cross validation and the performance is reported in terms of accuracy, recall (true positive rate), precision, and F_1 measure. These measures are computed as follows:

$$accuracy = (tp + tn)/(tp + tn + fp + fn) \tag{6}$$

$$recall = tp/(tp + fn) \tag{7}$$

$$precision = tp/(tp + fp) \tag{8}$$

$$F_1 = 2 \times precision \times recall/(precision + recall) \tag{9}$$

where tp refers to true positive, tn refers to true negative, fp refers to false positive, fn refers to false negative. We also compared the area under the Receiver Operating Characteristic (ROC) curve (AUC) and the time to construct the classification model.

4.3. Experiments and Results

The proposed learning methodology was implemented in Java and run in a Linux machine. We applied it to the network anomaly-based intrusion detection benchmark dataset without and with attribute selection. For attribute selection, we used the supervised correlation-based subset attribute selection with greedy hill-climbing. Out of the 41 attributes, only 11 attributes (approx. 26%) are returned by the attribute selector as follows: a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_{14}, a_{23}, a_{30}, a_{36} (their descriptions are as given in Table 1). We also tested another approach for attribute ranking and selection, but it returned 29 attributes and the performance of the classifier was worse and with higher time

Table 1. Summary of various attributes: category, notation, name, type (numeric, categorical, binary), statistics and description.

Cat.	Not.	Name	Type	Statistics Min	Max	Description
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, domain, etc.
	a_4	flag	cat.	–	–	Normal or error status of the connection; there are 11 potential values, e.g. rej, sh, etc.
	a_5	src_bytes	num.	0	693M	Num. of bytes from the source to the destination
	a_6	dst_bytes	num.	0	52M	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 guest login 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

complexity. So, we focused only on the greedy attribute selection. We conducted a comparative study with popular machine learning algorithms implemented in[23] with default settings using the stratified 10-fold cross-validation. Table 2 summarizes the performance comparisons of the proposed method with three other classifiers (decision tree ID3, support vector machine (SVM) and multi-layer perceptron (MLP)).

Table 2. Comparisons of accuracy versus time complexity for different approaches.

Approach	Accuracy (%)	Time (seconds)
Without attribute selection:		
Proposed	98.46	12
ID3	98.02	15
SVM	97.58	33
MLP	96.24	48
With attribute selection:		
Proposed	**99.96**	**7**
ID3	98.78	15
SVM	98.58	21
MLP	96.84	32

Table 3. The per-class performance of the proposed method without and with attribute selection (approx. to three digits).

		Without attribute selection				With attribute selection			
Normal/Attack	Count	Precision	Recall	F_1	AUC	Precision	Recall	F_1	AUC
normal	97278	0.990	0.989	0.989	0.990	1.000	1.000	1.000	1.000
back	2203	0.987	0.987	0.987	0.989	1.000	1.000	1.000	1.000
buffer_overflow	30	0.723	0.678	0.700	0.848	0.674	0.606	0.639	0.919
ftp_write	8	0.000	0.000	0.000	0.573	0.404	0.253	0.311	1.000
guess_passwd	53	0.933	0.933	0.933	0.962	0.990	0.935	0.961	0.990
imap	12	0.157	0.240	0.190	0.670	1.000	0.169	0.289	0.883
ipsweep	1247	0.985	0.983	0.984	0.989	0.928	0.993	0.959	1.000
land	21	0.847	0.937	0.890	0.919	0.960	0.914	0.937	0.962
loadmodule	9	0.000	0.000	0.000	0.766	0.336	0.112	0.169	0.876
multihop	7	0.276	0.323	0.298	0.847	0.253	0.144	0.184	0.933
neptune	107201	0.990	0.990	0.990	0.990	1.000	1.000	1.000	1.000
nmap	231	0.938	0.981	0.959	0.981	0.957	0.547	0.696	1.000
perl	3	0.323	0.990	0.490	0.980	1.000	0.336	0.505	0.981
phf	4	0.990	0.657	0.790	0.990	1.000	0.253	0.404	0.983
pod	264	0.990	0.986	0.988	0.990	1.000	1.000	1.000	1.000
portsweep	1040	0.977	0.982	0.979	0.987	0.997	1.000	1.000	1.000
rootkit	10	0.000	0.000	0.000	0.662	0.865	0.768	0.813	0.776
satan	1589	0.981	0.984	0.982	0.987	1.000	1.000	1.000	1.000
smurf	280790	0.990	0.990	0.990	0.990	1.000	1.000	1.000	1.000
spy	2	0.000	0.000	0.000	0.443	0.999	1.000	1.000	1.000
teardrop	979	0.989	0.990	0.989	0.989	1.000	1.000	1.000	1.000
warezclient	1020	0.968	0.982	0.975	0.988	1.000	0.986	0.994	1.000
warezmaster	20	0.790	0.752	0.770	0.910	0.842	0.758	0.797	0.902

Without attribute selection, the overall accuracy generated by PROAFTN is 98.46%. The decision tree ID3 approach achieved a classification accuracy of 98.02%; yet the size of the tree is so huge with many branches (812 branches with up to 694 leaves). The classification accuracy generated for SVM and MLP was 97.58 % and 96.24 %, respectively. Also it is worth noting that the time for building the PROAFTN classification model was reasonable compared with the aforementioned classifiers even though we are dealing with a big dataset. Comparing the time required to build the model, PROAFTN took 12 seconds whereas ID3 took 15 seconds. SVM and MLP were relatively time consuming; as each model required 33 and 48 seconds, respectively.

On the other hand, when greedy search attribute selection is used, the overall accuracy has improved for all classifiers. However, the proposed approach has significantly improved in terms of classification accuracy of 99.96% and training time of 7 seconds. In contrast, the decision tree ID3 approach achieved a classification accuracy of 98.78% but the size of the tree has increased with 886 branches and up to 784 leaves. Comparing with support vector machine (SVM) and multi-layer perceptron (MLP), the accuracy was 98.58 % and 96.84 %, respectively. The time required to build ID3, SVM and MLP models was relatively longer (15, 21 and 32 seconds, respectively). The detailed performance of PROAFTN for each class is summarized in Table 3.

5. Conclusion

A novel security countermeasure is proposed in this paper for anomaly-based intrusion detection. First a greedy search approach is applied for selecting the optimal subset of attributes. Then, a fuzzy-based multicriterion classification model is constructed and evaluated on a publicly available and widely used dataset. The results are very promising in terms of classification accuracy and model construction time. For instance, with only 11 attributes out of the 41 available attributes, the proposed classification model with attribute selection was able to yield more than 99.9% overall accuracy with very high detection rates for each attack type. As future work, it is intended to test the methodology on other datasets and feature selection methods.

Acknowledgements

The second 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 2014 Annual Security Report. Available on: `https://www.cisco.com/web/offer/gist_ty2_asset/Cisco_2014_ASR.pdf`
2. Sophos Security Threat Report 2014. Available on: `http://www.sophos.com/en-us/medialibrary/PDFs/other/sophos-security-threat-report-2014.pdf`
3. Tsai, C.F., Hsu, Y.F., Lin, C.Y., Lin, W.Y. Intrusion detection by machine learning: A review. *Expert Systems with Applications* 2009; **36**(10):11994 – 12000.
4. Garca-Teodoro, P., Daz-Verdejo, J., Maci-Fernndez, G., Vzquez, E. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security* 2009;**28**(12):18 – 28.
5. Davis, J.J., Clark, A.J. Data preprocessing for anomaly based network intrusion detection: A review. *Computers & Security* 2011; **30**(6):353–375.
6. Fenton, N.E., Wang, W. Risk and confidence analysis for fuzzy multicriteria decision making. *Knowledge-Based Sys.* 2006;**19**(6):430–437.
7. Zopounidis, C., Doumpos, M. Multicriteria classification and sorting methods: A literature review. *European Journal of Operational Research* 2002;**138**(2):229–246.
8. Al-Obeidat, F., Belacel, N., Carretero, J.A., Mahanti, P. An evolutionary framework using particle swarm optimization for classification method proaftn. *Applied Soft Computing* 2011;**11**(8):4971 – 4980.
9. Belacel, N.. Multicriteria assignment method proaftn: Methodology and medical application. *European Journal of Operational Research* 2000;**125**(1):175 – 183.
10. Wu, S.X., Banzhaf, W. The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing* 2010; **10**(1):1 – 35.
11. Altwaijry, H., Algarny, S. Bayesian based intrusion detection system. *J. King Saud University - Computer and Inf. Sc.* 2012;**24**(1):1 – 6.
12. Feng, W., Zhang, Q., Hu, G., Huang, J.X. Mining network data for intrusion detection through combining SVMs with ant colony networks. *Future Generation Computer Systems* 2013;.
13. Depren, O., Topallar, M., Anarim, E., Ciliz, M.K. An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks. *Expert systems with Applications* 2005;**29**(4):713–722.
14. li, W., Liu, Z. A method of SVM with normalization in intrusion detection. *Procedia Environmental Sciences* 2011;**11, Part A**:256 – 262.
15. Boln-Canedo, V., Snchez-Maroo, N., Alonso-Betanzos, A. Feature selection and classification in multiple class datasets: An application to KDD cup 99 dataset. *Expert Systems with Applications* 2011;**38**(5):5947 – 5957.
16. Hall, M.A. *Correlation-based feature selection for machine learning*. Ph.D. thesis; The University of Waikato; 1999.
17. Lger, J., Martel, J.M. A multi-criteria assignment procedure for a nominal sorting problematic. *European J. Operat. Res.* 2002;**138**:349–364.
18. Jabeur, K., Guitouni, A. A generalized framework for concordance/discordance-based multi-criteria classification methods. In: *The 10th International Conference on Information Fusion, 2007*. 2007, p. 1–8. doi:`10.1109/ICIF.2007.4408150`.
19. Baim, P. A method for attribute selection in inductive learning systems. *IEEE Trans. Pattern Analysis and Mach. Intell,* 1988;**10**(6):888–896.
20. Dutton, D., Conroy, G. A review of machine learning. *The Knowledge Engineering Review* 1996;**12:4**:341–367.
21. Fayyad, U., Irani, K. Multi-interval discretization of continuous-valued attributes for classification learning. In: *XIII International Joint Conference on Artificial Intelligence (IJCAI93)*. 1993, p. 1022–1029.
22. KDD Cup 1999 dataset for network-based intrusion detection systems. Available on: `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`
23. Witten, H. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems; 2005.

In [22], another approach for anomaly detection based on multicriterion fuzzy classification with greedy attribute selection is proposed and evaluated on KDDâ€™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.Â 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]. Index Termsâ€”Anomaly detection, deep learning, fuzzy logic, misuse detection. I. INTRODUCTION. In cybersecurity, the increasing dependence that companies have on their computer networks makes their protection from intrusion a critical issue.Â There are two main intrusion detection approaches: misuse and anomaly intrusion detection. Misuse intrusion detection is a rule-based approach that uses stored signatures of known intrusion instances to detect an attack. This approach is highly successful in detecting occurrences of previously known attacks. The main drawback of this approach is its inability to identify and characterise new attacks and to respond to them intelligently. Kernel based intrusion detection systems [Els00]. These are especially prevalent within Linux (LIDS, OpenWall). These systems examine the state of key operating system files and streams, preventing buffer overflow, blocking unusual interprocess communications, preventing an intruder from attacking the system. In addition, they can block a part of the actions undertaken by the super-user (restricting privileges). The HIDS reside on a particular computer and provide protection for a specific computer system. They are not only equipped with system monitoring facilities but also include other modu