I. J. Computer Network and Information Security, **2016, 11, 58-64**
Published Online November 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2016.11.07

A Learnable Anomaly Detection System using Attributional Rules

Abdurrahman A. Nasr
Al-Azhar University, System and Computer Engineering Dept., Cairo, 11651, Egypt
E-mail: anasr@azhar.edu.eg

Mohamed M. Ezz and Mohamed Z. Abdulmaged
Al-Azhar University, System and Computer Engineering Dept., Cairo, 11651, Egypt
E-mail: ezz.mohamed@azhar.edu.eg, azhar@eun.eg

Abstract—The continuous changing networks introduce new attacks, which represent an explicit problem that affects the security of enterprise resources. Thus, there is a real need to build up intelligent intrusion detection systems that can learn from the network behavior. In this paper, a learnable anomaly intrusion detection system based on attributional rules is presented. The proposed model is chosen with the advantages of being expressive, flexible and can operate in noisy and inconsistent environments. The system is a real-time intrusion detector that utilizes incremental supervised machine learning technique. Such technique makes use of the Algorithm Quasi-optimal (AQ) which is based on attributional calculus.

Here, an Algorithm Quasi-optimal for Intrusion Detection System (AQ4IDS) is exploited and implemented using attributional rules to discriminate between normal and anomalous network traffic. The behavior of AQ4IDS is tested, and to illustrate its superiority. The experimental results showed that, the model automatically accommodates new rules from continuous network stream. Many experiments have verified the fact that AQ4IDS can efficiently discriminate between normal and anomalous network traffic, in addition to offering the advantage of detecting novel and zero day attacks.

Index Terms—Intrusion detection, Algorithm Quasi-optimal, Attributional rules, data mining, Incremental learning, Real-time detection.

I. INTRODUCTION

IDS is one of the most essential component for security infrastructures in network environments, and it is widely used in detecting, identifying and tracking the intruders and safeguarding enterprise networks. The fundamental and foremost requirement in Intrusion Detection Systems (*IDSs*) is making the system intelligent enough to new information from the changing history of the network, such that it accommodates its knowledge-base incrementally.

Various researchers [1] have proposed different data mining techniques to learn the network behavior. Such techniques have been employed to build anomaly based intrusion detection systems. Example of such techniques is the support vector machine, artificial neural network, logistic regression, decision trees, association rules and decision rules. These techniques can be further divided in to black box (the former three algorithms) and white box (the latter three algorithms). Most closed box techniques suffer from the stability-plasticity dilemma when incrementally updated, while white box techniques usually don't.

One of efficient and comprehensible data mining techniques is the decision rules, which generates rule sets for discriminating between different classes in a dataset. Aside from decision rule are the attributional rules. Attributional rules are similar to normal decision rules, except that they employ a highly expressive representation language based on Attributional Calculus (*AC*) that combines aspects of propositional, predicate and multi-valued logic for the purpose of supporting pattern discovery and inductive learning. Moreover, attributional rules are concise, generic and more accurate compared to normal decision rules such as rules generated from C4.5 algorithm and RIPPER rule learner [2].

When employing data mining techniques for building adaptive incremental intrusion detection model; the main concerns about the algorithm are its efficiency, comprehensibility, justifiability and adaptivity. These criteria are partially achieved using rule learner algorithms, as they are human readable set of rules for discrimination between normal and anomalous behavior. Among all rule learners, attributional rule learner algorithms tend to be very accurate and efficient when extracting useful patterns from large volumes of poor, noisy or inconsistent data [3].

Algorithm quasi-optimal (*AQ*) [4], is a natural rule induction algorithm based on attributional rules. The algorithm almost fulfills the aforementioned concerns, by seeking different types of patterns in data and representing them in human-oriented forms resembling natural language descriptions. Moreover, it has the ability to adapt the generated rule sets so that no single rule covers both negative and positive examples at once.

In this paper, we focus on employing the power of attributional rules by utilizing AQ algorithm in adaptive incremental learning (*AIL*) of IDS. The model is endowed with a generalization capacity that covers new unknown attacks patterns. AQ4IDS is implemented with a two-class model implementation, which identifies network traffic as either normal or anomalous. This scenario represents a means for detection of new and zero day attacks. Moreover, the model is compared to other incremental methods to spotlight the efficacy of attributional rules, namely, the decision trees [5] and K* [6] algorithms.

II. Related Work

A wide range of data mining techniques have been employed in anomaly detection domain including, Support vector machine, Artificial neural network, decision trees, Bayesian network and many others [7], little of which employs incremental algorithms. Most researchers have concentrated on employing such techniques on intrusion detection using a well-known KDD99 benchmark dataset to verify their IDS adaptivity. In 1999, Syed et al. [8] proposed the incremental SVM by partitioning huge data into small partitions and train SVM on each partition. Baowen et al. [9] proposed an incremental algorithm for mining association rules. The algorithm considers not only adding new data into the knowledge base but also reducing old data from the knowledge base. Hassina et al. [10] proposed a new approach for IDS adaptability by integrating a Simple Connectionist Evolving System (SECOS) and a Winner-Takes-All (WTA) hierarchy of XCS (extended Classifier System). Hongle et al. [11] proposed a new incremental SVM method that combines support vector machine with clustering algorithm. Zhang et al. [12] has introduced incremental IDS based on a special version of a decision tree, which is the Hoeffding trees. They achieved a detection rate of 84%. Nasr et al. [13] proposed an incremental online pairwise model for intrusion detection that utilizes an ensemble of decision trees and AQ algorithms. Their overall model accuracy is 85%.

To improve the existing IDS models; AQ4IDS has been built as a learnable intrusion detection model on the basis of attributional rules. Also this model has been compared to other incremental learning models.

III. Illustrative Example

Algorithm Quasi-optimal (AQ) was introduced by Michalski in 1973 [4] and is based on attributional calculus, which is a highly expressive description language with well-defined syntax and semantics. In this section, we explain the AQ algorithm with illustrative example.

ATTRIBUTIONAL RULES FORMATION

AQ generates rules by an iterative process aimed at identifying generalizations of the positive examples with respect to the negative examples (i.e. rules are generated from examples and counterexamples). Listing 1 lists a high level AQ algorithm, where P represents positive examples (anomalous records in our case), and N represents negative examples (normal records in our case).

Listing 1: High Level AQ Algorithm

Required: |P| > 0 & |N| > 0

1. P* = P ; R = 0 //P* is a list of positive events to be covered
2. **While** |P*| > 1 **do**
3. Select random seed *p* from P*
4. r = STAR (*p*, N, maxstar) //find a rule that generalize the seed
5. P* = P* - [P* ∩ r]
6. R = R + r //increment the set of rules by new one
7. **end while**

The algorithm starts by selecting a random seed from positive event list P*, and then creates a STAR rule for that example, which is an iterative process aimed at generating a set of alternative general descriptions (rules) of the seed, that satisfy given constraints, for example, do not cover negative examples, do not contradict prior knowledge.

AQ algorithm learns incrementally by first classifying, then generalizing each new example to the best rule set generated previously. Before AQ generalizes a new example, it checks to see if there are any rule sets in the affected area of feature space that conflict with the proposed new rule set and of the opposite class. If so, the generalization is aborted and the record is stored verbatim.

ATTRIBUTIONAL RULES EXECUTION

To illustrate the AQ rule learner capabilities, we use a simple hypothetical problem. It should be noted, however, that the program can work with datasets containing thousands of instances. The following example is a simple one that has been inspired from [4]. Connection records (instances) are defined using the features described in table 1.

Suppose that our task is to determine strong patterns in examples for which the output attribute, *activity*, takes value "*Normal*". A generalized logic diagram (GLD) visualizing the representation space spanned over the input attributes and 22 input records is presented in Fig. 1 GLD is a technique for knowledge visualization in a compact view, in which features (attributes) are divided between rows and column of a rectangle.

Table 1. Common Features for the Hypothetical Example used in AQ Algorithm

Feature	Feature type	Feature values
Risk	Ordinal	{Risky, Probable, Safe}
Attack	Binary	{No, Yes}
Protocol	Nominal	{TCP, ICMP, UDP, HTTP}
Host-Type	Nominal	{Server, WorkStation}
Activity	Nominal	{Normal, Spam, PW_GUESS}

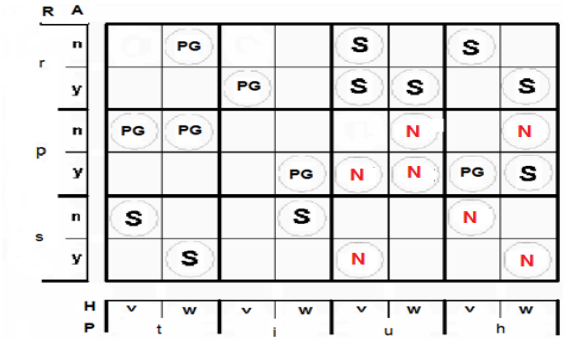

Fig.1. GLD Representation for Training Examples used in AQ Algorithm

The legend of the figure is as follows:

-**Decision class**: PG→ PW_GUESS, S→Spam, N→Normal.
-**Attributes**: R→Risk, A→Attack, H→Host-Type, P→Protocol.
-**Attribute values**: r→risky, p→probable, s→safe, n→no, y→yes, v→server, w→workstation, t→tcp, i→icmp, u→udp, h→http.

Next, we compare the AQ accuracy with well-known decision rule learner, which is RIPPER [2] to this dataset. The RIPPER program is applied to the same dataset, and the resultant rules are presented in below.

PW_GUESS:- protocol=tcp [3/5].
PW_GUESS:- protocol=icmp [2/3].
Normal:- risk=safe [3/6].
Normal:- risk=probable, attack=no [2/4].
Normal:- risk=probable, protocol=udp [3/3]
Default **Spam** (majority) [9/22]

Note that, these rules need to be evaluated sequentially, meaning that to obtain the hypothesis for activity "*Normal*", it is necessary to evaluate rules for activity "*PW_GUESS*". The number between square brackets represents rule coverage (support) / Accuracy (confidence).

Next, when AQ is applied to the same data set, only one string rule is presented below.

[Activity = *Normal*] ← [risk = probable **v** safe: (7,15)] & [protocol=udp **v** http: (7, 14)]: **pos**=7, **neg**=2, accuracy=0.77

This pattern consists of one rule stating that the activity is Normal, if the risk is probable or safe, and the protocol is udp or http. The rule covers 7 positive and 2 negative examples, and its accuracy is 0.77.

In conclusion, the RIPPER learner produced 3 rules for Normal activity that gave 5 errors out of 22 examples (23%) on the training dataset, whereas AQ produced one strong pattern (attributional rules) for the same activity, that covered all positive examples and 2 examples of other classes (9% error).

IV. The Proposed Model of AQ4IDS

AQ4IDS consists of two modes of operation: offline training and online testing. The training is carried out using a subset of 20% from NSL-KDD'99 dataset [14]. Also the testing is accompanied by the same percent of test data. The classification model distinguishes the input stream as either normal or anomalous (i.e. it's 2-class model).

Figs. 2a and 2b illustrate the training and testing datasets class statistics. This compact dataset was chosen as it consists of reasonable number of records which can be trained and tested by a moderate machine.

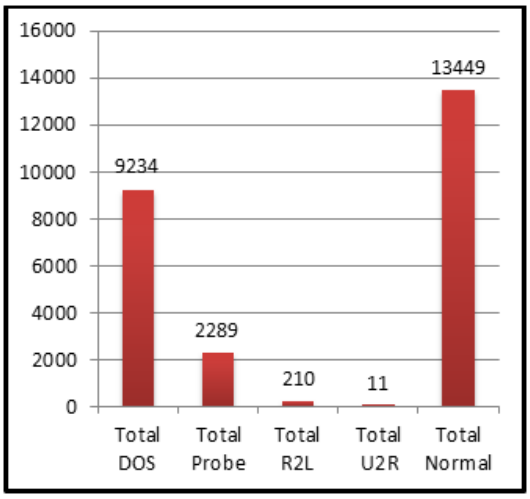

Fig.2a. NSL-KDD Training Dataset Statistics

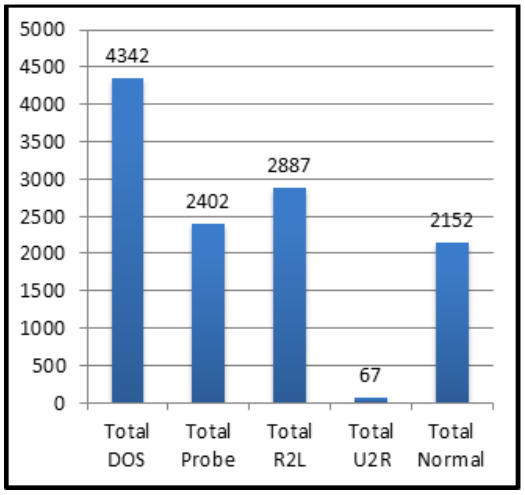

Fig.2b. NSL-KDD Testing Dataset Statistics

The model is evaluated online using prequential testing approach (a.k.a. Interleaved Test-Then-Train) by NSL-KDD test dataset. The prequential testing approach is an alternate scheme for evaluating data stream algorithms, in which each connection record is used to test the model before it is used for training it incrementally; and from this, the accuracy can be incrementally updated. When testing is performed in this order, the model is always being tested on a record it has not seen. Fig 3 & 4 illustrate the online and offline modes respectively.

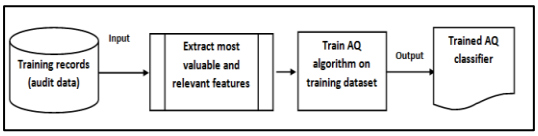

Fig.3. Offline mode of the AQ4IDS

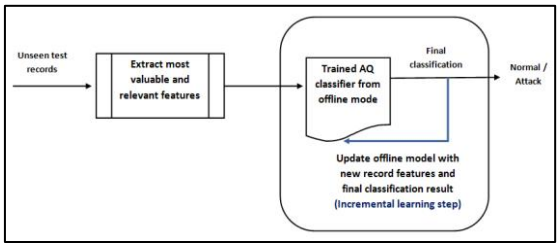

Fig.4. Online mode the AQ4IDS

Before training and testing the model, NSL-KDD'99 dataset are preprocessed to extract the 19 most valuable and relevant features (MVRF) based on the work done in [15] to identify most features affecting the evaluation of KDD'99 dataset.

To simulate network stream, the testing data is loaded in memory and fed sequentially to the model (in online configuration) one by one, and the model is prequentially tested.

Two steps are involved in online mode, the first is the classification step, which identify connection record as normal or anomalous, and the second is to incrementally update (learn) the AQ classifier with new information obtained from record features and class. This ensures the model adaptivity with the latest environment changes, yielding it adaptable to concept drift and ability to detect zero day attacks.

V. IMPLEMENTATION

To measure AQ algorithm efficacy and accuracy, an implementation has been carried out for AQ4IDS using Java programming language, with the aid of WEKA, which is an open source tool for machine learning algorithms and data mining tasks, and Massive Online Analysis (MOA) which is an open source framework for data stream mining and big data processing.

To provide a means for detection of zero day attack, AQ4IDS is trained on different datasets by excluding specific type of attack in training mode, and presenting the attack in testing mode.

AQ4IDS TWO-CLASS IMPLEMENTATION

Implementation of AQ4IDS classifies training examples in NSL-KDD dataset into normal or anomalous connection (2-class model). The dataset contains 5 main classes, namely, Normal, DoS, Probe, R2L and U2R. The final classification result of AQ4IDS will be normal or anomalous (regardless of attack type).

To ensure the model capability to detect zero day and new attacks, AQ4IDS is trained over 5 different datasets obtained from NSL-KDD by varying training data. The datasets involved in this experiment is explained in table 2.

Table 2. Variation on NSL-KDD Training Dataset

Dataset	Description
All-Data	Represents NSL-KDD training dataset, without altering
No-DoS	Represents NSL-KDD training dataset, after removing all DoS attacks
No-Probe	Represents NSL-KDD training dataset, after removing all Probe attacks
No-R2L	Represents NSL-KDD training dataset, after removing all R2L attacks
No-U2R	Represents NSL-KDD training dataset, after removing all U2R attacks

Removing specific attack from training phase and presenting it in test phase (online mode) simulates a real network situation, in which new attacks are emerged and concepts drift may occur. The output of this experiment is 5 AQ models, each trained on specific dataset.

AQ4IDS PERFORMANCE EVALUATION

In this section, the evaluation of AQ4IDS, and a comparison of its performance with different incremental learning algorithms are given. The evaluation is based on: (i) Accuracy, which is the correct classified records, over all records, (ii) detection rate, which the correctly classified attacks over all attacks, and (ii) false alarm rare, which is the normal records, classified incorrectly as attack, over all normal.

Fig. 5 compares the accuracy graphs for AQ algorithm, from the first experiment (i.e. the model is trained on different 2-class datasets by excluding specific attack type). The best classification accuracy is obtained by training the model on All-Data dataset without excluding any attack, at which the accuracy ranges from 87.5% to 93.3%.

Fig. 6 compares the detection rate of AQ from the first experiment. It seems that the detection rate is degraded when excluding Probe and U2R attack from training data, and presenting them in online mode, while detection rate remains similar for the other datasets. Actually, this figure provides a means for detecting new and zero day attacks. The model detection rate and its learning process -for unseen attack- are increasing with increasing the observed records.

Fig. 7 compares the false alarm rate of AQ from the first experiment. From the figure, training AQ4IDS on No-Probe and No-R2l gives the lowest false alarm rate, while keeping similar results for other datasets.

Fig. 8 compares the AQ algorithm to other classification algorithms. These algorithms include: (i)the K* algorithm [6], (ii)incremental decision trees [5], (iii)and the work done in [13], which employs an ensemble of incremental decision trees and AQ. From this figure, K* achieves best accuracy at the expense of high classification time (the average classification time is 0.68 ms, 1.61 ms, 799.59 ms for DT, AQ and K* respectively). On the long run, AQ reaches K* accuracy, Fig. 8, in addition to the fact that it's sufficiently fast to be deployed online.

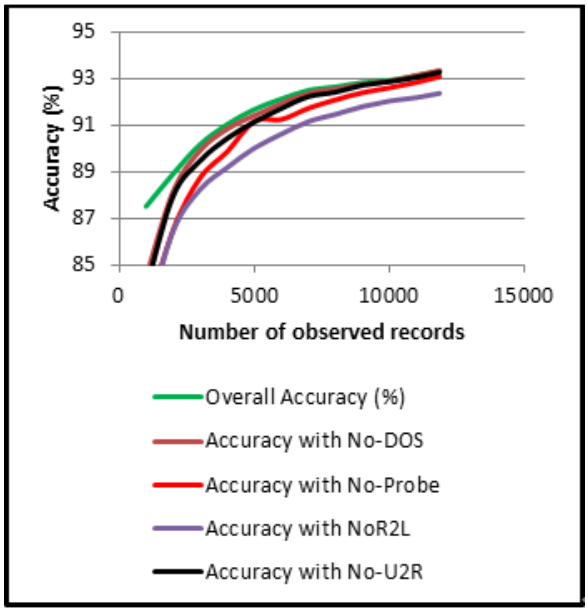

Fig.5. Accuracy Graph for AQ4IDS

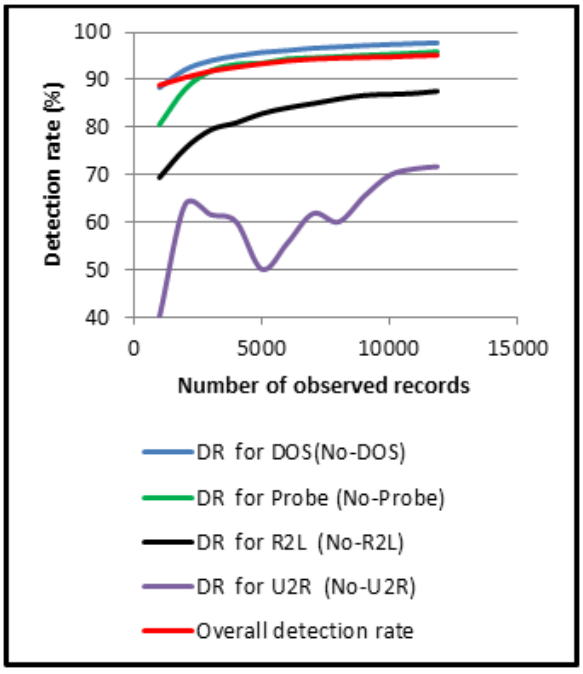

Fig.6. Detection rate Graph for AQ4IDS

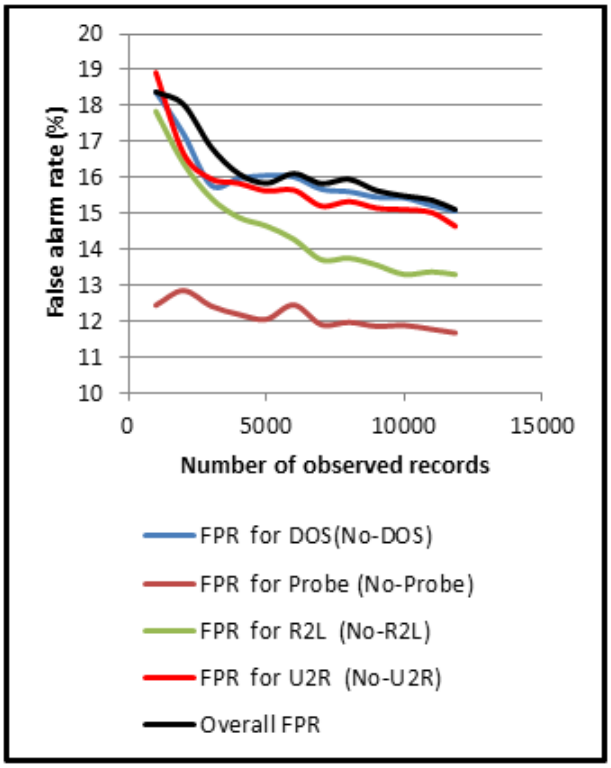

Fig.7. False Alarm Rate Graph for AQ4IDS

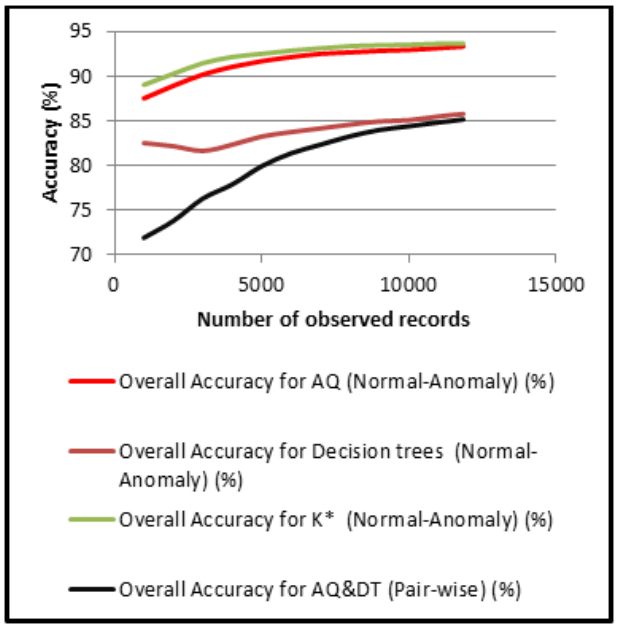

Fig.8. Model Comparison with Other Algorithms

The promising results of AQ4IDS in detecting anomalies are due to the accurate identification of normal traffic that has been expressed using attributional rules. An excerpt from these rules is tabulated in table 3 and confirms the compactness, expressiveness and flexibility of attributional rules over decision rules.

Table 3. Generated Attributional Rules for NORMAL Class

Generated rule		Record count
NORMAL IF:	protocol_type in {tcp} ^ service in {http} ^ 139.0<=src_bytes<=538.0 ^ wrong_fragment=0.0 ^ hot=0.0 ^ num_failed_logins=0.0 ^ logged_in=1.0 ^ num_compromised=0.0 ^ root_shell=0.0 ^ 0.0<=num_access_files<=1.0 ^ 0.0<=serror_rate<=1.0 ^ 0.0<=srv_serror_rate<=1.0 ^ 0.0<=rerror_rate<=0.17 ^ 0.0<=srv_rerror_rate<=0.67 ^ 0.5<=same_srv_rate<=1.0 ^ 0.0<=diff_srv_rate<=1.0 ^ 1.0<=dst_host_srv_count<=255.0 ^ dst_host_srv_diff_host_rate=0.0 ^ 0.0<=dst_host_serror_rate<=0.84	(2671)
NORMAL IF:	protocol_type in {icmp,udp} ^ service in {ntp_u,urh_i,other,domain_u} ^ 17.0<=src_bytes<=145.0 ^ wrong_fragment=0.0 ^ hot=0.0 ^ num_failed_logins=0.0 ^ logged_in=0.0 ^ num_compromised=0.0 ^ root_shell=0.0 ^ num_access_files=0.0 ^ serror_rate=0.0 ^ srv_serror_rate=0.0 ^ rerror_rate=0.0 ^ srv_rerror_rate=0.0 ^ 0.09<=same_srv_rate<=1.0 ^ 0.0<=diff_srv_rate<=0.67 ^ 3.0<=dst_host_srv_count<=255.0 ^ dst_host_srv_diff_host_rate=0.0 ^ 0.0<=dst_host_serror_rate<=0.01	(2338)

VI. Conclusion

In this paper, a learnable real-time model has been proposed for building up AQ4IDS to provide a learner for network anomalies using attributional calculus. The model is based on employing 2-class scenario for identification of network traffic as normal or anomalous. The model has verified its efficiency in detection of new and zero day attacks, and its capability of learning new attributional rules from the network stream. The overall accuracy of the model is 93.3% with overall detection rate 94.36%, and overall false alarm rate 15%.

References

[1] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion Detection System: A Comprehensive Review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, Jan. 2013.

[2] W. W. Cohen, "Fast Effective Rule Induction," in *Proceedings of the Twelfth International Conference on Machine Learning, Lake Tahoe, California*, 1995.

[3] J. Wojtusiak, R. S. Michalski, K. A. Kaufman, and J. Pietrzykowski, "The AQ21 natural induction program for pattern discovery: initial version and its novel features," in *Tools with Artificial Intelligence, 2006. ICTAI'06. 18th IEEE International Conference on*, 2006, pp. 523–526.

[4] J. Wojtusiak, R. S. Michalski, K. A. Kaufman, and J. Pietrzykowski, "Multitype Pattern Discovery via AQ21: A Brief Description of the Method and Its Novel Features," *Reports Mach. Learn. Inference Lab.*, vol. 1051, pp. 2–6, 2006.

[5] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2001, pp. 97–106.

[6] J. G. Cleary, L. E. Trigg, and others, "K*: An Instance-based Learner Using an Entropic Distance Measure," in *ICML*, 1995, pp. 108–114.

[7] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, Jan. 2013.

[8] N. A. Syed, H. Liu, and K. K. Sung, "Handling concept drifts in incremental learning with support vector machines," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '99*, 1999, pp. 317–321.

[9] B. Xu, T. Yi, F. Wu, and Z. Chen, "An incremental updating algorithm for mining association rules," *J. Electron.*, vol. 19, no. 4, pp. 403–407, Oct. 2002.

[10] H. Bensefia and N. Ghoualmi, "A New Approach for Adaptive Intrusion Detection," *2011 Seventh Int. Conf. Comput. Intell. Secur.*, pp. 983–987, Dec. 2011.

[11] H. Du, S. Teng, M. Yang, and Q. Zhu, "Intrusion detection system based on improved SVM incremental learning," in *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on*, 2009, vol. 1, pp. 23–28.

[12] X. Yun, L. Zhang, I. Security, and C. Network, "Using Incremental Learning Method For Adaptive Network," no. August, pp. 18–21, 2005.

[13] A. Nasr, M. Ezz, and M. Abdulmageed, "Use of Decision Trees and Attributional Rules in Incremental Learning of an Intrusion Detection Model," *Int. J. Comput. Networks Commun. Secur. IJCNCS*, vol. 2, no. 7, pp. 216 – 2 24, 2014.

[14] "The NSL-KDD Data Set." [Online]. Available: http://nsl.cs.unb.ca/NSL-KDD/. [Accessed: 24-Jun-2014].

[15] M. Salem and U. Buehler, "Mining Techniques in Network Security to Enhance Intrusion Detection Systems," *CoRR*, p. 16, Dec. 2012.

Authors' Profiles

Abdurrahman A. Nasr is a lecturer of software engineering, Computer and System Engineering Department, Faculty of Engineering, Al-Azhar University at Cairo. He received his M.Sc. and Ph.D. degrees in electrical engineering from Al-Azhar University in 2012, and 2014 respectively. His fields of interest include artificial intelligence, stochastic process, machine learning, data mining, mathematics and operating systems.

Mohamed M. Ezz is a lecturer of software engineering, Computer and System Engineering Department, Faculty of Engineering, Al-Azhar University at Cairo. He received his B.Sc., M.Sc. and Ph.D. degrees in electrical engineering degrees in electrical engineering from Al-Azhar University. His fields of interest include network security, and cryptography.

Mohamed Z. Abdulmageed is the professor of software engineering, Computer and System Engineering Department, Faculty of Engineering, Al-Azhar University at Cairo. He received his B.Sc. and M.Sc. degrees in electrical engineering from Cairo University in 1968 and 1973 respectively. He received his Ph. D. degrees in computer engineering from Warsaw Technical University, Poland in 1977. His fields of interest include artificial intelligence, soft computing, and distributed systems.

How to cite this paper: Abdurrahman A. Nasr, Mohamed M. Ezz, Mohamed Z. Abdulmaged,"A Learnable Anomaly Detection System using Attributional Rules", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.11, pp.58-64, 2016.DOI: 10.5815/ijcnis.2016.11.07

