*I. J. Computer Network and Information Security,* **2016, 1, 53-60**
Published Online January 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2016.01.07

# An Intrusion Detection and Prevention System based on Automatic Learning of Traffic Anomalies

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

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

***Abstract*—**The ever changing network traffic reveals new attack types, which represent a security threat that poses a serious risk for enterprise resources. Therefore, the security administrators are in a real need to employ efficient Intrusion Detection and Prevention Systems, IDPS. Such systems might be capable to learn from the network behavior. In this paper, we present an incremental Learnable Model for Anomaly Detection and Prevention of Zero-day attacks, LMAD/PZ. To facilitate the ability of learning from observations that can provide a reliable model for automatic prevention, a comparison has been carried out between supervised and unsupervised learning techniques.

Thus, in LMAD/PZ, the intrusion detection step is integrated with an intrusion prevention plan. To ensure that the prevention plan is dependable and automatic, it must be backed and sustained with robust and accurate detection process. Therefore, two incremental data mining techniques are deeply investigated and implemented on NSL-KDD'99 intrusion dataset. The first technique is the Algorithm Quasi-optimal (AQ), which is a supervised Attributional Rules Learner, ARL, while the second is the Cobweb; an unsupervised hierarchical conceptual clustering algorithm. These algorithms categorize the network connections as either normal or anomalous. The performance of AQ is compared to Cobweb, and the best performance result is integrated with the prevention plan, to afford a fully automated system. The experimental results showed that, the model automatically adapts its knowledge base from continuous network streams, in addition to offering the advantage of detecting novel and zero day attacks. Many experiments have verified that AQ performance outperforms the Cobweb clustering, in terms of accuracy, detection rate and false alarm rate.

***Index Terms*—**Incremental learning, Conceptual clustering, Attributional calculus, Intrusion detection, Intrusion prevention, Zero-day attack.

## 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 [1]. 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 adapts its knowledge base incrementally.

Various researchers [2]–[4] 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, decision rules, K-means clustering, hierarchical clustering and outlier detection. These algorithms can be further divided into supervised and unsupervised data mining techniques.

One of efficient and comprehensible supervised 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 [5]. Moreover, attributional rules are concise, generic and more accurate compared to normal decision rules such as rules generated from C4.5 algorithm [6] and RIPPER rule learner [7].

Among all rule learners, attributional rule learner tends to be very accurate and efficient when extracting useful patterns from large volumes of data. Moreover, they engage background knowledge (in the form of generalization rules) about the problem in order to compensate for the data limitations when deriving useful knowledge from poor, noisy and inconsistent data [8].

 

Algorithm quasi-optimal (*AQ*) [9]–[11], 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. The simplest form of generated rules from AQ is

CONDITION → DECISION

Where condition and decision are complexes; conjunctions of Attributional conditions, for example

[src_bytes = 30...170] ^[Service = telnet ∨ ftp] ^
[Protocol = tcp ]→[Activity = Anomalous]

Which means that an activity is anomalous if *src_bytes* ranges from [30-170], and *service* is in {telnet, ftp} and *protocol* in {tcp}. The algorithm has many features such as learning rules with exceptions, determining optimized sets of alternative hypotheses generalizing the same data, and to handle data with missing, irrelevant and/or not-applicable meta-values.

On the other hand, Cobweb is gaining attention due to its incremental nature, economical computations, and being one of few incremental clustering techniques in unsupervised learning arena. Cobweb is a conceptual clustering algorithm developed in the late 1980 by Fisher [12]. It processes each new instance as it appears, and creates hierarchical clustering. The algorithm carries out a hill-climbing search through a space of hierarchical classification schemes using operators that enable bidirectional travel through this space, and then it measures the clusters quality based on heuristic measure, which is the category utility function. In this paper, Cobweb is used to cluster all network connection records into normal or anomalous constellation, after that, any unseen record is classified according to the most similar cluster (a.k.a. classification via clustering).

Employing incremental attributional rules and conceptual clustering in LMAD/PZ produces a powerful real-time model, that's capable of detecting novel and zero-day attacks. At the same time, it saves relevant knowledge that had been learned, previously, from old network streams [13].

The rest of this paper is organized as follows: Section 2 highlights related work about current incremental IDSs. Section 3 describes the proposed detection model. Section 4 presents the proposed integration plan. Section 5 describes the implementation of the model and evaluation results, and section 6 concludes the proposed model.

## II. Related Work

A wide range of data mining techniques have been employed in anomaly detection domain including, Support vector machine [14], Artificial neural network [15], decision trees [16], Bayesian network [17] and many others [2], little of which employs incremental algorithms. Most researchers have concentrated on employing such techniques on intrusion detection using a well-known KDD99 [18] benchmark dataset to verify their IDS adaptivity. In 1999, Syed et al. [19] proposed the incremental SVM by partitioning huge data into small partitions and train SVM on each partition. Zhang et al. [20] extended the traditional SVM, Robust SVM and one-class SVM to be of online incremental forms. Baowen et al. [21] 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. Shafi et al. [22] proposed an Adaptive Rule based Intrusion Detection Architecture, which integrates a signature rules base with a Learning Classifier System (LCS) to produce interpretable rules. It allows learning new attack and normal behavior patterns by interacting with a security expert. Hassina et al. [13] 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. [23] proposed a new incremental SVM method that combines support vector machine with sequential k-means clustering algorithm. Zhang et al. [24] has introduced incremental IDS based on a special version of a decision tree, which is the Hoeffding tree. They achieved a detection rate of 84%. Leckie et al. [25] proposed a time varying of the standard clustering techniques to accommodate non-stationary traffic distribution. Shi-Jinn et al. [26] proposed and IDS that integrate BIRCH clustering algorithms with SVM, in which SVM is trained over fewer and highly qualified training data from BIRCH output. Nasr et al. [27] 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%.

Conceptual clustering also has been employed in intrusion detection systems. Panda et al. [28] proposed a hybrid clustering approach based on Cobweb and farthest first traversal clustering algorithms for classification of rare attacks, such as U2R and R2L. Julisch et al. [29] proposed a variant of Attribute Oriented Induction (AOI) conceptual clustering technique, to mine historical false alarm patterns from knowledge base, in order to handle future alarm more efficiently. Petrovic et al. [30] introduced a new cluster labeling techniques for attacks identification based on combination of Davies-Bouldin index of clustering and centroid diameter evaluation.

## III. Detection Model

Prior to the prevention plan, we need to state a strict and dependable methodology for detection process. The proposed methodology 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 [31]. Also the testing is accompanied by the same

 

percent of test data. This compact dataset was chosen as it consists of reasonable number of records which can be trained and tested by a moderate machine.

At the beginning, the AQ classifier and Cobweb clustering algorithms are trained on the training data set (batch training, offline mode), and the records are distinguished as either normal or anomalous (i.e. it's 2-class model). Once finished, the generated models are retained to be deployed online (Fig. 1).

Next, the models are evaluated online using prequential testing approach [32] (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. 2 illustrates the online mode, in which the generated models from offline mode are reloaded.

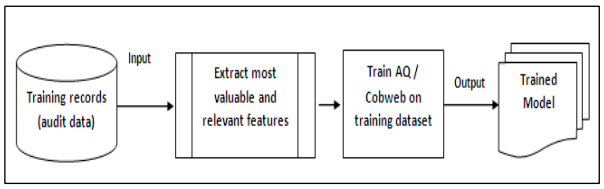


Fig.1. Offline mode of the proposed model

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 [33] to identify most features affecting the evaluation of KDD'99 dataset.

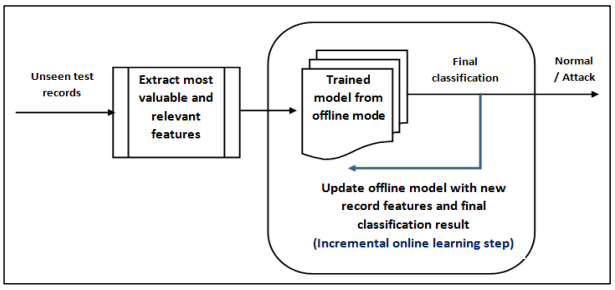


Fig.2. Online mode the proposed model

To simulate network stream, the testing data is read sequentially from the file system and fed one by one to the AQ/Cobweb (in online configuration), 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 underlying learning technique 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. It's important to note that Cobweb clustering algorithm produces hierarchical conceptual clusters that are either normal or anomalous (so that any new incoming traffic will belong to either concepts of the generated clusters).

The offline and online phase for the proposed methodology will be carried out separately on supervised AQ learner, and unsupervised Cobweb clustering, to figure out the efficacy of each technique. The final result will be concluded and the most accurate approach will be used in the generated system with the prevention plan. Moreover, the proposed system is pluggable, in sense of plugging either AQ or Cobweb models as requested; besides, the prevention plan can be changed dynamically.

## IV. Integrating the Detection Model with Prevention Plan

The Intrusion Prevention System (IPS), is a complementary component in LMAD/PZ analysis, and is considered as an extension of intrusion detection. Both components monitor network traffic and/or system activities for abnormal events. An IDS captures packets in real time, processes them, and can respond to threats, but works on copies of data traffic to detect suspicious activity by using anomaly detection techniques. This is called sniffing mode, where as an IPS works in line with the data stream to provide protection from malicious attacks in real-time. This is called inline mode. As a result, they form an integral part of a robust network defense. The main differences are, unlike intrusion detection systems, intrusion prevention systems are placed in-line and are able to actively prevent/block intrusions that are detected [34].

The active response of IPS can be categorized into two approaches: i-reactive response, which is activated and executed after intrusion has been detected, ii- proactive response, which is a set of preemptive actions to prevent an intended attack. Here, we employ the first response of IPS, where Fig. 5 presents the integration of the detection model with the prevention plan.

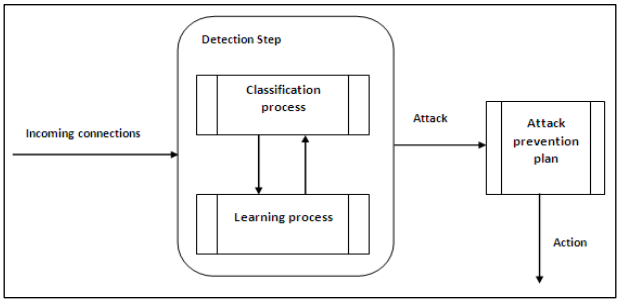


Fig.3. Integration or prevention plan with the detection model

Listing 1 illustrates a high level prevention process. The prevention plan is automated via the following steps:

First, IPS tries to mitigate attack effect by removing malicious or suspicious content if applicable (for example, by escaping characters, encoding contents, etc…); otherwise the next action will be triggered.

Second, the event correlation component of the IPS identifies the attack category by consulting the underlying classifier employed in the IPS (to be

explained later). If the attack category is beyond a predefined threshold, then the active response of IPS will be triggered.

Third, IPS will "*drop and reset connection*" in conjunction with *alarm* and *denial* of any connection originated from attacker address for a predefine period of time.

Listing 1: High Level prevention process

1. Mitigate attack effect if applicable
   **Otherwise**
2. Calculate attack category level by consulting prevention plan classifier
   **If** (category > THRESHOLD)
   **Do**
3. Drop current packet
4. Reset connection
5. Block and deny any future connections from this attacker address for a predefined period of time
6. Raise an alarm to administrators
7. Train the prevention classifier on this attack, and update statistics
   **Done**
8. Log attack incident, along with its statistics, into system log file.

It's important to note that the active response must not be over aggressive. This is because packets that are dropped based on false alarms can result in network disruption if the dropped packets are required for mission-critical applications downstream of the IPS system. Therefore, the false alarm rate for the prevention plan should be tuned automatically by analyzing historical alarm records, and digging out effective patterns that help in deciding future alarms. The tuning process is achieved by training a classifier on connection records that generated such alarms (for example naïve Bayses classifier for reactive response, or hidden Markov model for proactive response). Having done this, we ensure the automaticity of the plan execution

The final step of the plan is to log the incident into system log file for later auditing by network administrators.

## V. Implementation

To compare the AQ and Cobweb algorithms efficiency and accuracy for LMAD/PZ, an implementation has been carried out for both algorithms using Java programming language, with the aid of [35] and [36]. The implementation of the LMAD/PZ 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/clustering result will be normal or anomalous (regardless of attack type).

### *A. The Training Step*

To provide a means for detection of new and zero day attacks, both algorithms are trained on 5 different datasets, obtained from NSL-KDD training dataset, by excluding specific type of attack in training mode, and presenting the attack in testing mode. The datasets involved in this experiment is explained in table 1.

Table 1. Variation on NSL-KDD training dataset

| Dataset No. | Dataset Name | Description |
|---|---|---|
| 1 | All-Data | Represents NSL-KDD training dataset, without altering |
| 2 | No-DoS | Represents NSL-KDD training dataset, after removing all DoS attacks |
| 3 | No-Probe | Represents NSL-KDD training dataset, after removing all Probe attacks |
| 4 | No-R2L | Represents NSL-KDD training dataset, after removing all R2L attacks |
| 5 | 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 models (e.g. 5 AQ classifiers, or 5 Cobweb clustering models), each trained on specific dataset.

### *B. Performance Evaluation*

In this section, the evaluation of AQ and Cobweb, and a comparison of their performance 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 (a.k.a. false positive rate, FPR), which is the normal records, classified incorrectly as attack, over all normal.

Fig. 4 and 5 compare the accuracy graphs for AQ and Cobweb algorithms respectively. The best classification accuracy is obtained by training both algorithms on All-Data dataset without excluding any attack, at which the accuracy ranges from 87.5% to 93.3% for AQ, and from 65.5% to 77% for Cobweb.

Fig. 6 and 7 compare the detection rate of AQ and Cobweb respectively. It seems that for AQ, the detection rate is degraded when excluding R2L and U2R attack from training data, and presenting them in online mode, while detection rate remains similar for the other datasets. On the other hand, Cobweb shows similarity between different detection rates, except at All-Data dataset. Actually, these figures provide a means for detecting new and zero day attacks. The algorithms detection rate and their learning process -for unseen attack- are increasing with increasing the observed records

Fig. 8 and 9 compare the false alarm rate of AQ and Cobweb respectively. From the figure, training AQ on No-Probe and No-R2L gives the lowest false alarm rate, while keeping similar results for other datasets. On the other hand, Cobweb shows low false alarm rate, when trained on All-Data dataset.

Figure 10 compares the overall accuracy of AQ and Cobweb. From this figure, it's clear that AQ accuracy far exceeds the accuracy of Cobweb.

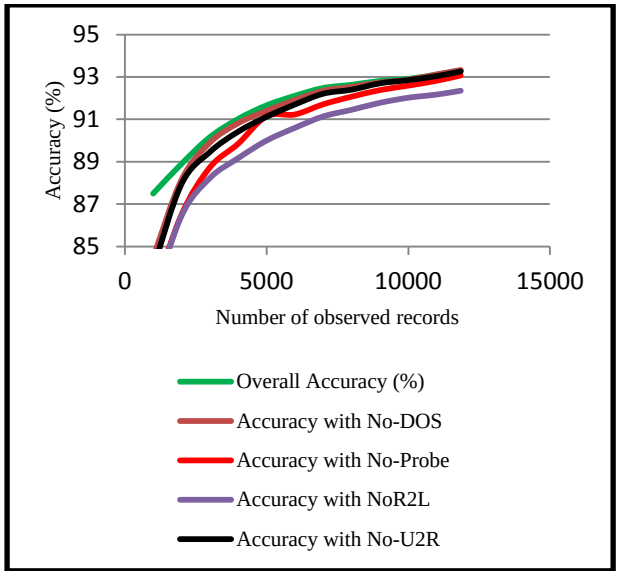


Fig.4. Accuracy graph for AQ

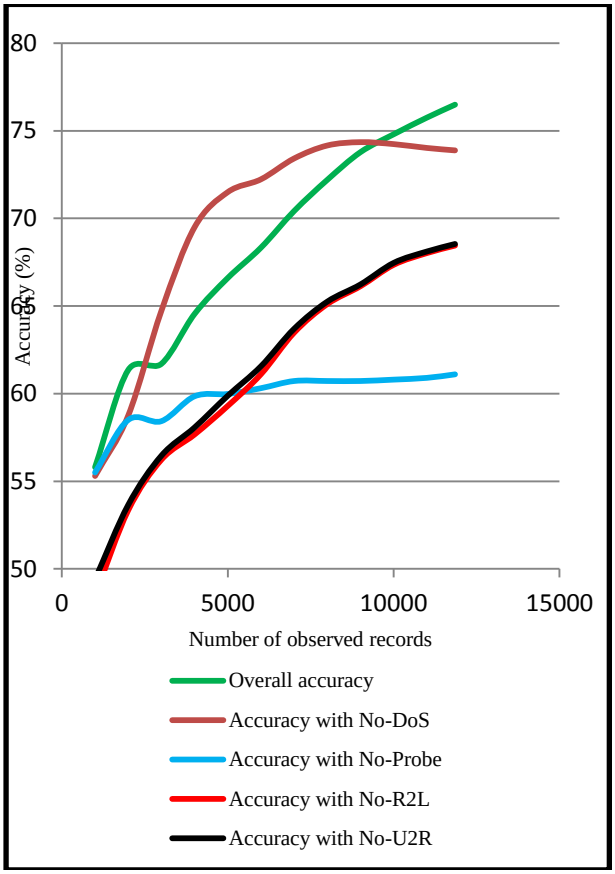


Fig.5. Accuracy graph for Cobweb

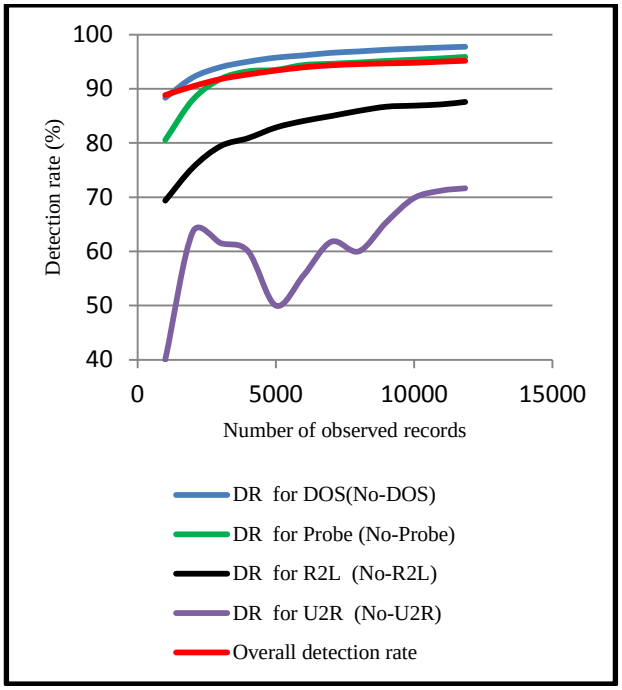


Fig.6. Detection rate graph for AQ

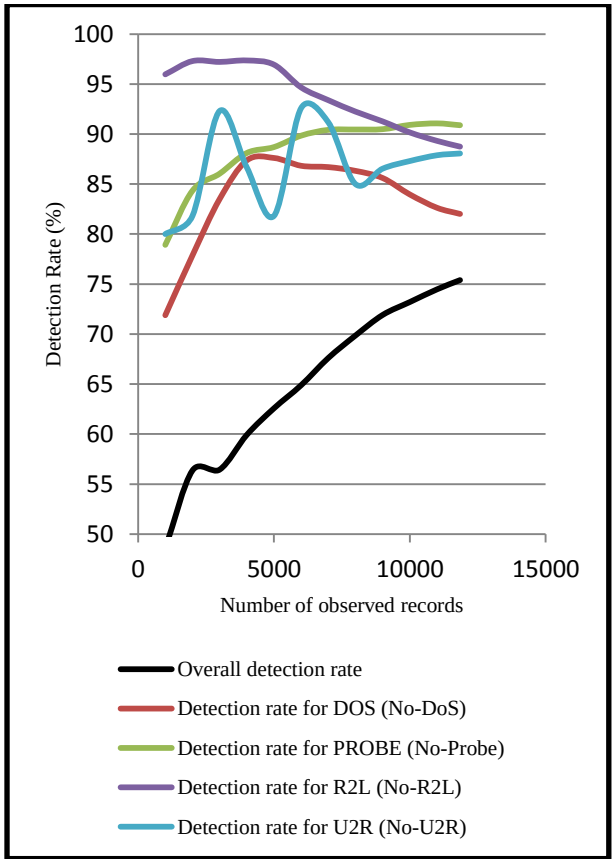


Fig.7. Detection rate graph for Cobweb

 

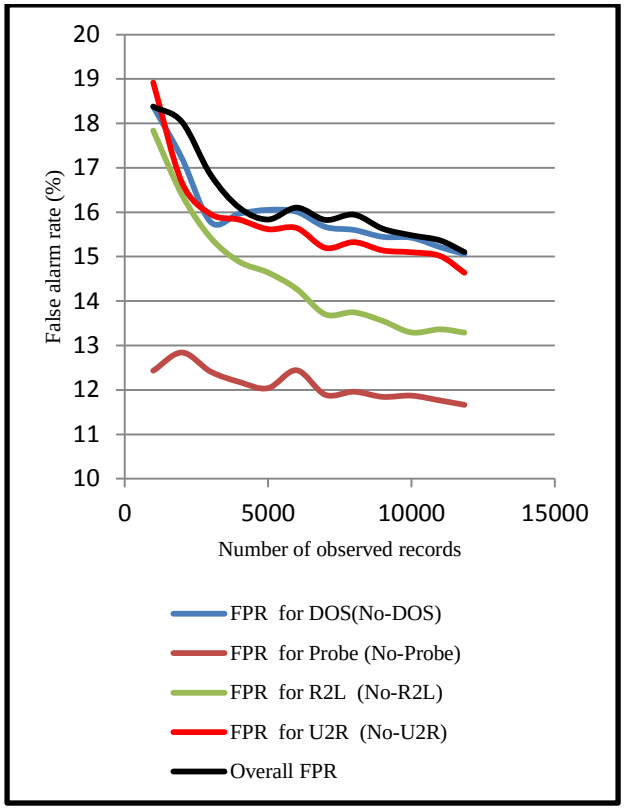


Fig.8. False alarm rate graph for AQ

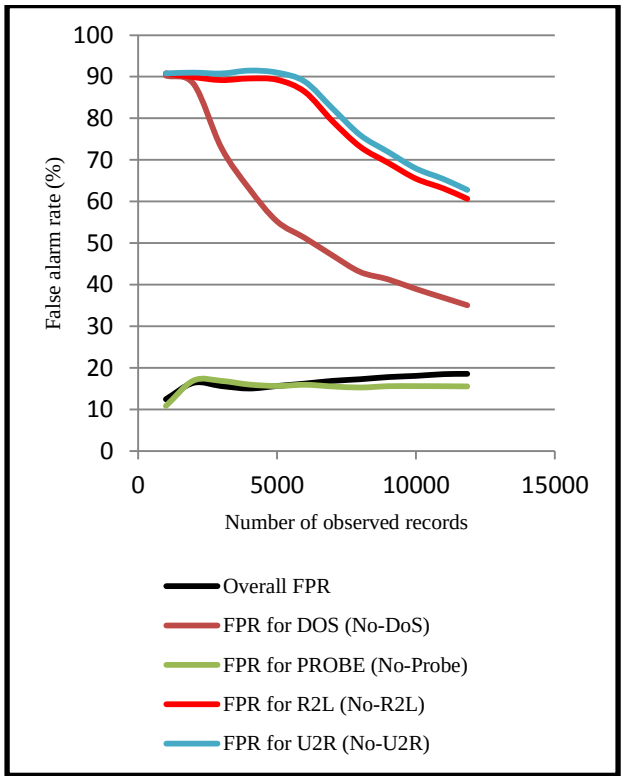


Fig.9. False alarm rate graph for Cobweb

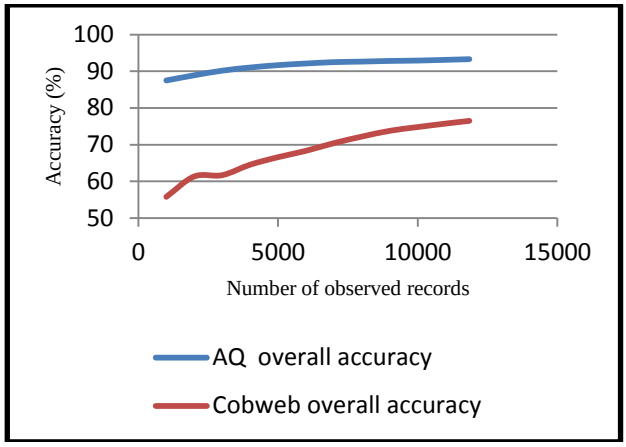


Fig.10. Accuracy comparison between AQ and Cobweb

The promising results of AQ algorithm in detecting anomalies have pioneered AQ over Cobweb to be plugged into the proposed IDPS system. This is due to the accurate identification of normal traffic that has been expressed using attributional rules. These rules are tabulated in table 2 and confirm the compactness, expressiveness and flexibility of attributional rules over decision rules.

Table 2. 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) |

From the above results, it's obvious that overall detection rate and accuracy of AQ algorithm far exceed their counterparts in Cobweb, whereas the overall false alarm rate is approximately equal. This is because AQ is a supervised learning algorithm compared to the unsupervised learning Cobweb.

## VI. Conclusion

In this paper, an incremental Learnable Model for Anomaly Detection and Prevention of Zero-day attacks, LMAD/PZ is presented. That model is based on making use of attributional rules and conceptual hierarchical clustering. Different scenarios have been taken into consideration during the model implementation. These scenarios are based on employing 2-class model and training each algorithm on different datasets. Based on these scenarios, the two algorithms are compared, and AQ algorithm showed dominant performance compared to Cobweb. The algorithm has verified its efficiency in detection of new and zero day attacks, and its capability of learning new attributional rules from the network stream, which makes the best fit for integration with the proposed prevention plan. The evaluation statistics of the system based on AQ algorithm, has shown enhancement in all evaluation criteria. Several experiments have confirmed that, the overall accuracy of LMAD/PZ reaches 93.3%.

## References


[1] J. H. Lee, J. H. Leet, S. G. Sohn, J. H. Ryu, and T. M. Chung, "Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection system," in *International Conference on Advanced Communication Technology, ICACT*, 2008, vol. 2, pp. 1170–1175.

[2] 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.

[3] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 11994–12000, Dec. 2009.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[5] R. S. Michalski, "Attributional calculus: A logic and representation language for natural induction," *Reports Mach. Learn. Inference Lab. MLI 04-2, Georg. Mason Univ.*, 2004.

[6] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993, p. 302.

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

[8] 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.

[9] J. Wojtusiak and R. S. Michalski, "The LEM3 System for Non-Darwinian Evolutionary Computation and Its Application to Complex Function Optimization," no. C, pp. 2005–2010, 2010.

[10] G. Cervone, P. Franzese, and A. P. K. Keesee, "Algorithm quasi-optimal (AQ) learning," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 2, pp. 218–236, Mar. 2010.

[11] 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.

[12] D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Mach. Learn.*, vol. 2, no. 2, pp. 139–172, 1987.

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

[14] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 42–57, Jan. 2013.

[15] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Networks*, vol. 51, no. 12, pp. 3448–3470, Aug. 2007.

[16] S. S. Sivatha Sindhu, S. Geetha, and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 129–141, Jan. 2012.

[17] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, no. 1–2, pp. 18–28, Feb. 2009.

[18] "KDD Cup 1999 Dataset." [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. [Accessed: 23-Jun-2015].

[19] 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.

[20] Z. Zhang and H. Shen, "Application of online-training SVMs for real-time intrusion detection with different considerations," *Comput. Commun.*, vol. 28, no. 12, pp. 1428–1442, Jul. 2005.

[21] 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.

[22] K. Shafi, H. A. Abbass, and W. Zhu, "An Adaptive Rule-based Intrusion Detection Architecture," *Secur. Technol. Conf. 5th Homel. Secur. Summit, Aust.*, pp. 345–355, 2006.

[23] 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.

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

[25] J. Oldmeadow, S. Ravinutala, and C. Leckie, "Adaptive clustering for network intrusion detection," in *Advances in Knowledge Discovery and Data Mining*, Springer, 2004, pp. 255–259.

[26] S.-J. Horng, M.-Y. Su, Y.-H. Chen, T.-W. Kao, R.-J. Chen, J.-L. Lai, and C. D. Perkasa, "A novel intrusion detection system based on hierarchical clustering and

support vector machines," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 306–313, 2011.
[27] 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.
[28] M. Panda and M. R. Patra, "A hybrid clustering approach for network intrusion detection using cobweb and FFT," *J. Intell. Syst.*, vol. 18, no. 3, pp. 229–246, 2009.
[29] K. Julisch and M. Dacier, "Mining intrusion detection alarms for actionable knowledge," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 366–375.
[30] S. Petrovic, G. Alvarez, A. Orfila, and J. Carbó, "Labelling clusters in an intrusion detection system using a combination of clustering evaluation techniques," in *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on*, 2006, vol. 6, p. 129b–129b.
[31] "The NSL-KDD Data Set." [Online]. Available: http://nsl.cs.unb.ca/NSL-KDD/. [Accessed: 24-Jun-2015].
[32] A. P. Dawid, "Present Position and Potential Developments: Some Personal Views: Statistical Theory: The Prequential Approach," *J. R. Stat. Soc. Ser. A*, vol. 147, no. 2, p. 278, 1984.
[33] M. Salem and U. Buehler, "Mining Techniques in Network Security to Enhance Intrusion Detection Systems," *CoRR*, p. 16, Dec. 2012.
[34] M. Whitman and H. Mattord, *Principles of information security*. Cengage Learning, 2011.
[35] "Weka 3 - Data Mining with Open Source Machine Learning Software in Java." [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed: 24-Jun-2015].
[36] "MOA Massive Online Analysis, Data Stream Analytics in Real Time." [Online]. Available: http://moa.cms.waikato.ac.nz/. [Accessed: 24-Jun-2015].

## 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,"An Intrusion Detection and Prevention System based on Automatic Learning of Traffic Anomalies", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.1, pp.53-60, 2016.DOI: 10.5815/ijcnis.2016.01.07

 