An Adaptive Fuzzy Framework based on Optimized Fuzzy Contexts for Detecting Network Intrusions

HABIBULLAH BAIG*, MAHMOOD AHMAD SHEIKH*, AND FARRUKH KAMRAN*

RECEIVED ON 08.11.2008 ACCEPTED ON 01.06.2009

ABSTRACT

AIDS (Anomaly based Intrusion Detection System) is one of the key component of a reliable security infra-structure. Working at second line of defense, detection accuracy is the key objective that largely depends upon the precision of its normal profile. Due to existence of vague boundaries between normal and anomalous classes and dynamic network behavior, building accurate and generalize normal profile is very difficult. Based on the assumption that intruder's behavior can be grouped into different phases active at different times, this article proposes to evolve and use 'short-term fuzzy profiles/contexts' for each such individual intrusion phase resulting in enhanced detection accuracy for low-level attacks. The result is a context-driven, adaptable implementation framework based on a double layer hierarchy of fuzzy sensors. The framework adapts to network conditions by switching between different contexts, according to network traffic patterns, anomaly conditions and organization's security policies. These contexts are evolved in incremental fashion with GA (Genetic Algorithm) using real-time network traces. The framework is tested using DARPA 98/99 dataset showing accurate detection of low-level DoS attack.

Key Words: AIDS, Fuzzy Logic, GA, Fuzzy Context, Context Switching.

1. INTRODUCTION

The expansion of Internet and its usage in business and commerce has attracted many cyber criminals who try to exploit its power to achieve financial gains, personal motives and political objectives. These cyber criminals use sophisticated intrusion techniques including evolving programs and polymorphic codes to evade network security mechanisms. They even have created global cooperative networks to explore and exploit new vulnerabilities and to hide the ongoing criminal activity. Because of similar conditions and threats, network security has gained much more attention from the research community, system adminis-trators and state-level security organizations [1].

Defense mechanisms such as firewalls and intrusion detection systems have been proposed and imple-mented to detect and prevent malicious activities. Halme [2] classifies anti-intrusion techniques into six exclusive categories. These include preemption, prevention, deterrence, detection, deflection, and coun-termeasure techniques. Among the perimeter-based approaches, Intrusion Detection is most important be-cause new and normal-like malicious traffic often passes through perimeter preventive defenses.

Anomaly based Intrusion detection systems have proven their worth by detecting zero-age intrusions, but suffers

*Ph.D. Student, and **Professor,

Department of Computer Engineering, Center for Advanced Studies in Engineering (CASE) Islamabad, Pakistan

from large number of false alarms mainly because of imprecise definition of normal profile. Due to dynamics of network traffic and concept-drift phenomenon, constructing precise normal-profile is very difficult. Because of these and other such similar conditions, it is imperative to have a dynamic intrusion detection system with ability to change its detection profile according to network conditions and intrusion patterns.

Building an effective IDS and determining its normal profile is an enormous knowledge engineering process [3]. Researchers used statistical techniques [4-5], data mining approaches [6-9] and AI methods to construct normal-profiles [10-13]. These publications used offline standard data sets with single objective i.e. to enhance detection rate. Constructing a generalize profile for optimizing multiple conflicting objec-tives and adapting dynamic traffic conditions in an efficient manner is a hard problem.

The adaptive anomaly based fuzzy framework proposed in this paper is based on statistical variations, fuzzy logic and genetic algorithm. Inspired from well known multi-objective genetic algorithm: the SPEA (Strength Pareto Evolution Algorithm) [14] which uses multiple sets of solutions (population) we propose to construct two types solutions i.e. normal profiles; Generalized profile and Local profile. Generalized profile depends upon long-term history and is evolved using offline training data set with the objective of optimizing detection accuracy. The Local profile is a collection of short-term profiles each evolved using real time network traces over a limited time in the current history. These individual profiles are specialized for detecting local anomaly conditions and traffic variations. The Local profiles are evolved using GA and are used one at a time based on the network conditions, security policies and intrusion phase. Utilization of both Generalized and Local profiles result in improved detection performance, specifically for low-level attacks, as demonstrated in this paper.

The paper is organized as follows: Section 2 covers existing related work, Section 3 presents an overview of the proposed framework with its components; and Section 4 summaries the simulations and results. Conclusions are drawn in Section 5 with proposed future extensions in Section 6.

2. RELATED WORK

AI (Artificial Intelligence) techniques have proven their worth in learning classification rules from network data and hence automating the manual development of intrusion signatures. Among AI approaches fuzzy logic has been extensively used in intrusion detection systems, because of its ability to represent abstract and imprecise processes and to express quantitative attributes in qualitative terms. Traditionally fuzzy if-then rules are obtained from domain experts but evolutionary algorithms have been used to extract fuzzy rules. Dickerson, et. al. [15] first proposed FIRE (Fuzzy Intrusion and Recognition Engine) to detect network in-trusions using manual fuzzy rules. Botha [10] used fuzzy logic to model the user behavior classified into six exclusive phases. He used predefined rules and separate membership functions to model each phase that has inherent limitation in detecting new anomalies and adapting network condition. Gomez, et. al. [11], Yihua Liao [16], Saniee Abadeh [17] and Lee, et. al. [9] combined fuzzy logic and genetic algorithm to evolve fuzzy rules, optimize membership functions to detect new anomalies.

Gomez [11] used genetic algorithms with fuzzy inference engine to detect intrusions using mined fuzzy rules from DARPA data set. He also classified rules for normal and abnormal classes using intrusive and non-intrusive i.e. normal data-set [11]. Saniee Abadeh [17] and Chi-Ho Tsang [18] used multi-objective functions; high accuracy, interpretability of rules, and increase search capability to construct new rules for intrusion detection.

Yihua Liao [16] proposed an adaptive anomaly intrusion detection framework based on unsupervised evolution of connectionist system. The paper uses skewed class distribution for classification of normal and intrusive processes.

3. Overview of AAFF for Intrusion Detection

The proposed framework AAFF (Adaptable Anomaly Fuzzy Framework) as shown in Fig. 1 is divided into three main functional components; i.e. the data collection and organization module, the anomaly detection engine and the management in-terface. The data collection and organization module sniffs live traffic from the network physical interface and extracts various features. These features are arranged both in time independent as well as in temporal fashion (as described in the next section) to form various feature sets, which help detection engines in de-tecting a specific class of intrusions. The detection engine consists of two sub-sections; one specialized against network intrusions like port-scan, etc. while the other is optimized against DoS attacks. The man-agement interface and information-sharing module takes the results from the detection engine and presents a global picture of network security to the security professionals.

Following is a brief description of each of these individual modules.

3.1 Data Collection and Organization Module

Data collection and organization module sniffs live network traffic from network physical interface and extracts various features. These features are organized into two different feature sets based on their rele-vance in detecting different class of intrusions. Features from TCP (Transmission Control Protocol) packets containing RST flags and ICMP (Internet Control Message Protocol) packets, which help detection engines in detecting reconnaissance attacks, are arranged in a lexicographical order based upon the quadruplet (Net_ID, IP _ID, Dst_Port, Src_Port). Since this data organization is independent of the arrival time of a packet, the set is referred to as TIFS (Time Independent Feature Set) as defined in [19]. TIFS data presentation amplifies the scanning pattern irrespective of number of probes and their occurrence in a given time space. Features needed for detecting denial of service attacks are organized based on their arrival time in TFS (Temporal Feature Set). Features such as number of packets, number of SYN's, FIN, RSTs and the number of connected sessions collected in a particular time interval constitute temporal feature set TFS.

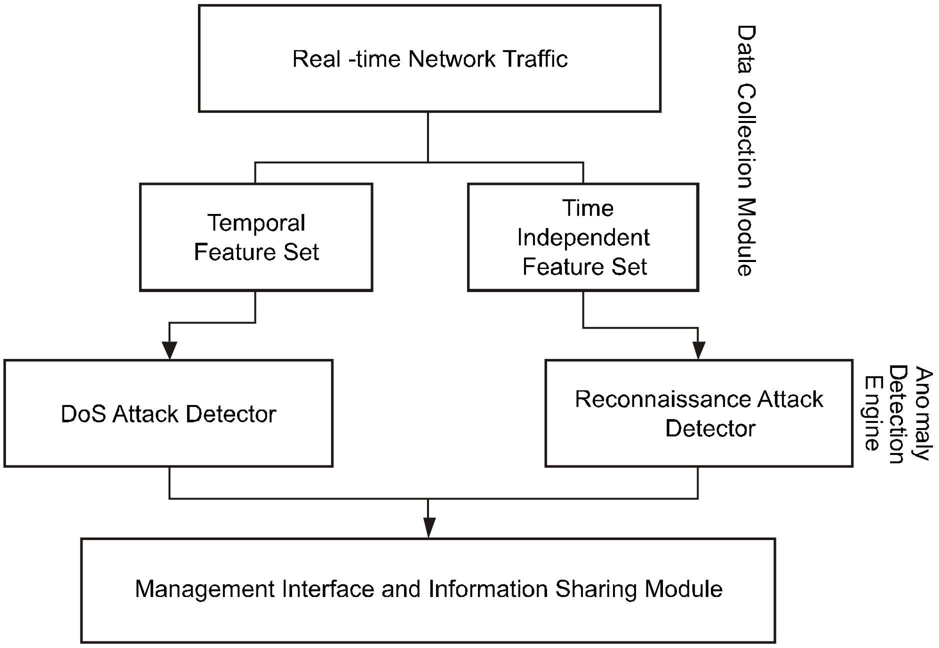

Real -time Network Traffic
Data Collection Module
Temporal Feature Set
Time Independent Feature Set
DoS Attack Detector
Reconnaissance Attack Detector
Anomaly Detection Engine
Management Interface and Information Sharing Module

FIG. 1. ADAPTIVE ANOMALY FUZZY FRAMEWORK (AAFF) ARCHITECTURE

3.2 The Anomaly Detection Engine

The detection engine is the central component of the proposed fuzzy framework. It consists of fuzzy agents, Genetic algorithm based Context Generation and Evolution Module and Context database and switching logic module shown in Fig. 2. Detection engine hosts multiple fuzzy agents. Each fuzzy agent uses dif-ferent normal profile depending upon type of phase of an attack. Context generation and evolution module constructs these normal profiles also called as Fuzzy Context. Detection engine activates/switches these profiles/contexts depending upon network conditions, intrusion phases and network security polices. Context Generation and Evolution module evolves various contexts, optimizing each for different objec-tives, such as high detection rate, low false positives or low detection time. Evolved contexts are stored in Context database and switched in real-time according to switching mechanism described in Section 3.2.3.

3.2.1 The Fuzzy Agent

Fuzzy agent is the basic element in the adaptive anomaly fuzzy framework, whose responsibility is to detect a specific attack or a particular phase of an attack. It consists of three components; fuzzy Context, expo-nential moving average module and fuzzy inference engine shown in Fig. 3. Fuzzy context represents the problem domain i.e. normal profile of network in reference to particular intrusion. Exponential moving average module adapts the fuzzy context according to current network conditions and traffic patterns, while fuzzy inference engine actually classifies an event using fuzzy knowledge base and real-time inputs.

Fuzzy context is a key component of the fuzzy agent, which consists of rules and membership functions. Context generation and evolution module constructs optimized rules and membership functions for current network and anomaly conditions described in Section 3.2.2. Fuzzy rules can be expressed in terms of simple if-then statements with higher interpretability score. If Number of SYN, Delta_SYN, Number of FIN, Numer of RST and Syn_Attack are linguistic variables with low, medium and high as fuzzy sets, then fuzzy rules can be expressed as

If Number of SYN is medium and Delta_SYN is medium then Syn_Attack is high (w_1=0.9, $S_{r,1}$=1)

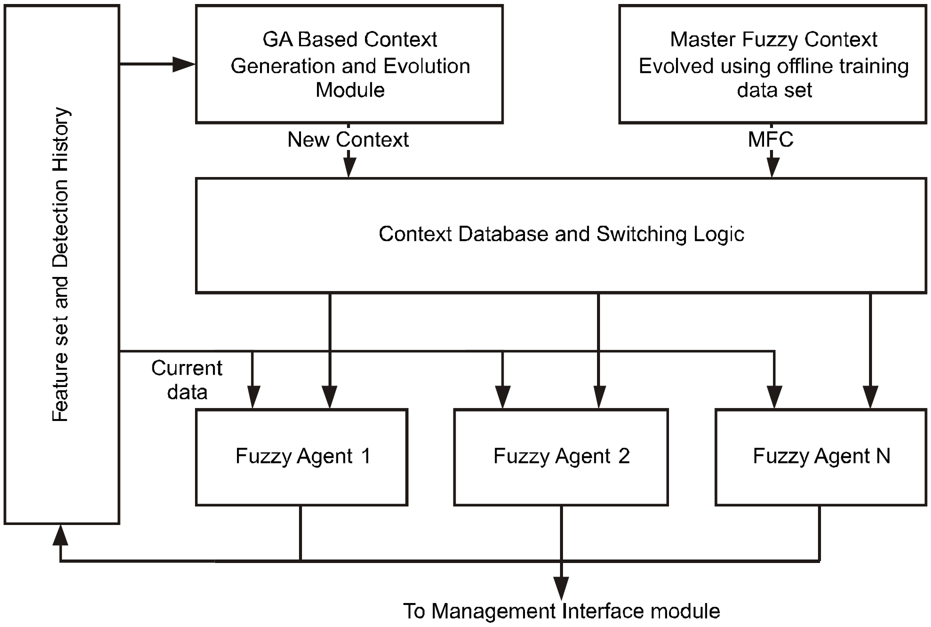

GA Based Context Generation and Evolution Module
Master Fuzzy Context Evolved using offline training data set
New Context
MFC
Context Database and Switching Logic
Feature set and Detection History
Current data
Fuzzy Agent 1
Fuzzy Agent 2
Fuzzy Agent N
To Management Interface module

FIG. 2. ARCHITECTURE OF A TYPICAL CLASS DETECTOR

If Number of SYN is medium and Delta_SYN is high and Number of FIN is medium and Number of RST is high then Syn_Attack is medium (w_{11}=0.6,$S_{r,11}$=1)

Here w_n is rule-weight of nth rule and S_r is n-bit real number, where each bit represents individual rule's status.

The membership functions, which are other component of fuzzy knowledge base, define membership de-gree of fuzzy linguistic sets over the input space, which are modeled using boundary parameters $\beta_{low1,2}$, $\beta_{medium1,2,3}$, $\beta_{high1,2}$ proposed by Habib, et. al. [19]. The fuzzy agent adapts the current network traffic and accommodates the concept drift phenomenon by varying boundary parameters consequently adjusting the membership functions.

Network traffic profile for specific protocol i.e. TCP, is computed by taking running average of key feature such as TCP connected-sessions. Exponential Moving average module, which is second component of fuzzy agent, computes the moving average $\overline{\Phi_{np}}$ of the selected feature. The modules output which can be termed as network profiler is described in Equation (1).

$$\overline{\Phi_{np}}(n) = \alpha\Phi_{cnp}(n) + (1-\alpha)\overline{\Phi_{np}}(n-1) \quad (1)$$

Where n is the interval number, α determine the weight of current record Φ_{cnp}(n) determine the weight of current record, is number of connected sessions in nth interval and $\overline{\Phi_{np}}$ is the modules output for previous interval i.e. history.

Let the fuzzy sets for fuzzy linguistic variables are low, medium and high. The membership functions of each linguistic fuzzy set in terms of boundary parameters are describe by Equations (2-4) [19]. The boundary parameters are functions of evolved parameters ω_{low1}...ω_{high2} and moving average modules output. Member-ship functions contract or expand linearly according to network history depending upon exponential moving average modules output. This helps in adjusting the attack threshold value at that particular interval while evolved parameters set the normal and not-normal class boundaries.

$$[\beta_{low1}, \beta_{low2}] = [\omega_{low1}, \omega_{low2}] * \overline{\Phi_{cnp}} + \gamma_1 \quad (2)$$

$$[\beta_{medium1}, \beta_{medium2}, \beta_{medium3}] = [\omega_{medium1}, \omega_{medium2}, \omega_{medium3}] \quad (3)$$

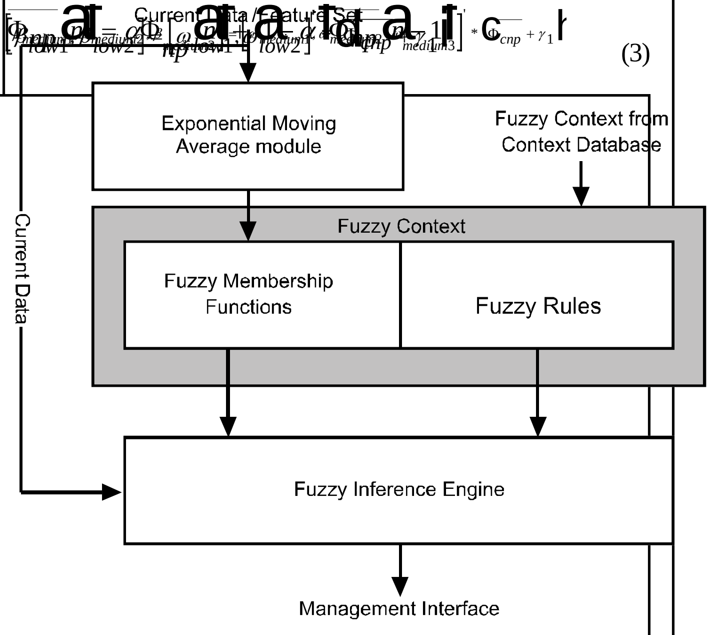

Current Data / Feature Set
Exponential Moving Average module
Fuzzy Context from Context Database
Fuzzy Context
Fuzzy Membership Functions
Fuzzy Rules
Current Data
Fuzzy Inference Engine
Management Interface

FIG 3. ARCHITECTURE OF FUZZY AGENT

$$\left[\beta_{high1},\beta_{high2}\right]=\left[\omega_{high1},\omega_{high2}\right]' * \mathbf{C}\overline{\Phi}_{cnp}+\gamma_1 \mathbf{h} \qquad (4)$$

The parameter γ_1 is constant, which represent the general relationship between fuzzy linguistic variables and feature used for network profiling i.e. TCP connected sessions. For example γ=2, if number-of-SYN is linguistic fuzzy set and TCP connected-sessions is used for modeling network conditions.

Fuzzy inference engine that is third component of fuzzy agent, classifies the real-time input as normal or malicious using fuzzy knowledge base. It basically accomplishes three functions (fuzzification, fuzzy in-ference, defuzzification) based on Mumdani principle [19]. In fuzzification, a crisp input i.e. a record from feature set is mapped to fuzzy sets to determine the membership degree. The inference engine evaluates applicable rules and their degree of matching to generate consequent rules. The defuzzification function aggregates the consequent rules and using centroid method, generates one crisp output, which determines the class of input record [19].

3.2.2 Context Generation and Evolution Module

Context generation and evolution module constructs fuzzy contexts using standard Genetic algorithm GA. It has three components; GA, dataset and fitness function. GA is a search technique used in computing to find exact or approximate solutions for optimization problems. Optimization problems are loosely modeled in terms of real numbers or bit strings (called chromosomes) abstractly representing real solutions (Individuals). The evolution usually starts from a population of randomly generated individuals and moves towards better solution in successive generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (using crossover and mutation operators) to form a new population. The new population is then used in the next iteration, thus moving towards better solution set.

The module evolves fuzzy context as candidate solution i.e. chromosome in terms of rules and mem-bership functions. Fuzzy Membership functions are encoded in terms of chromosome by membership boundary parameters ω_{low11},....ω_{high21},...ω_{low1n},...ω_{high2n} where each individual gene is representing by a real number.

The framework use pre-defined rule set with configurable rule-weights and rule-status. Domain expert initializes the rule-Set. In this framework a total 16 rules are added into rule-set. Context generation and evolution module constructs new rule-sets, selecting completely or part of predefine rules by changing there weights and status bits. Thus, new contexts are evolved optimizing low detection time (less number of rules) and high detection rate. Chromosome representation of fuzzy rules in terms of rule-weights and rules-status is [w_1,w_2,...w_n,S_r].

Phenotype representation of a chromosome in Matlab fuzzy toolbox is shown in Fig. 4. It is a snapshot of fuzzy context C1 at particular interval number 2000. It consists of two linguistic variables with total seven rules. Here only rule 5 has fired which specifies that if Number of SYN is medium and Delta_SYN is me-dium with other two variables, as don't cares, then set the output fuzzy-set to high. The record at particular interval number is classified as malicious with 80.4% confidence level.

The evolution module use incremental evolution by adding domain knowledge by initializing contexts with previous evolved contexts. Experimental results show that it helps in decreasing evolution time.

Each evolved chromosome is tested using a data set with known intrusions. Comparing the detection results of individuals with data-set gives the number of FP (False Positives), number of TD (True Detection), and the number of records which chromosome has not classified i.e. number of UC (Uncertain Records). Fitness of individual chromosome is computed by putting these parameters in cost function ff(n) defined in Equation (5). It is the sum of products of number of false positives and their associated weights w_{FP}, number of true detections and true detection weight factor w_{TD}, number of records classified as uncertain and their weight factor w_{UC}, number of rules N_{rules} and number of features N_f with respective feature cost F_c. Cost mini-mization is achieved by selecting those candidate solutions from population, which results into reduced number of FP, increased number of TD, reduced uncertain UC records and minimize the number of rules, thus moving towards a better fitness value.

$$ff(n)=\sum(FP*w_{FP}-TD*w_{TD}+UC*w_{UC})+N_{rules*}w_{rules}+N_f*F_c \tag{5}$$

By changing TD weight factor, FP weight factor and uncertain weight factor ω_{UC}, various contexts are evolved each optimized for either minimizing false positives, or maximizing detection rate.

The module evolves two types of contexts known as MFC (Master Fuzzy Context) and LFC (Local Fuzzy Context) based on two different datasets. MFC is evolved offline using standard training data set such as DARPA98/99 optimized for high accuracy, detecting generalize, and most frequent anomalous trends in network traffic. This context uses full features and all rules. It is complex and slow thus lags the real-time traffic. The second context is lightweight and optimized for detecting local anomaly conditions or a certain phase of an attack. Agent using this context is termed as Local fuzzy agent. It is evolved using a new data-set composed by network traces previously classified by Master fuzzy agent and respective local fuzzy agents. Only those records are used, which have been classified by both master and local agents with conforming results. With changing network conditions and intrusion patterns, new LFC are evolved using current network traces almost in near real-time. Evolved LFC are considered mature when they classify all the records or give better performance than the other stored/active contexts. These evolved contexts are stored in context database with their attributes and switched according to context switching mechanism.

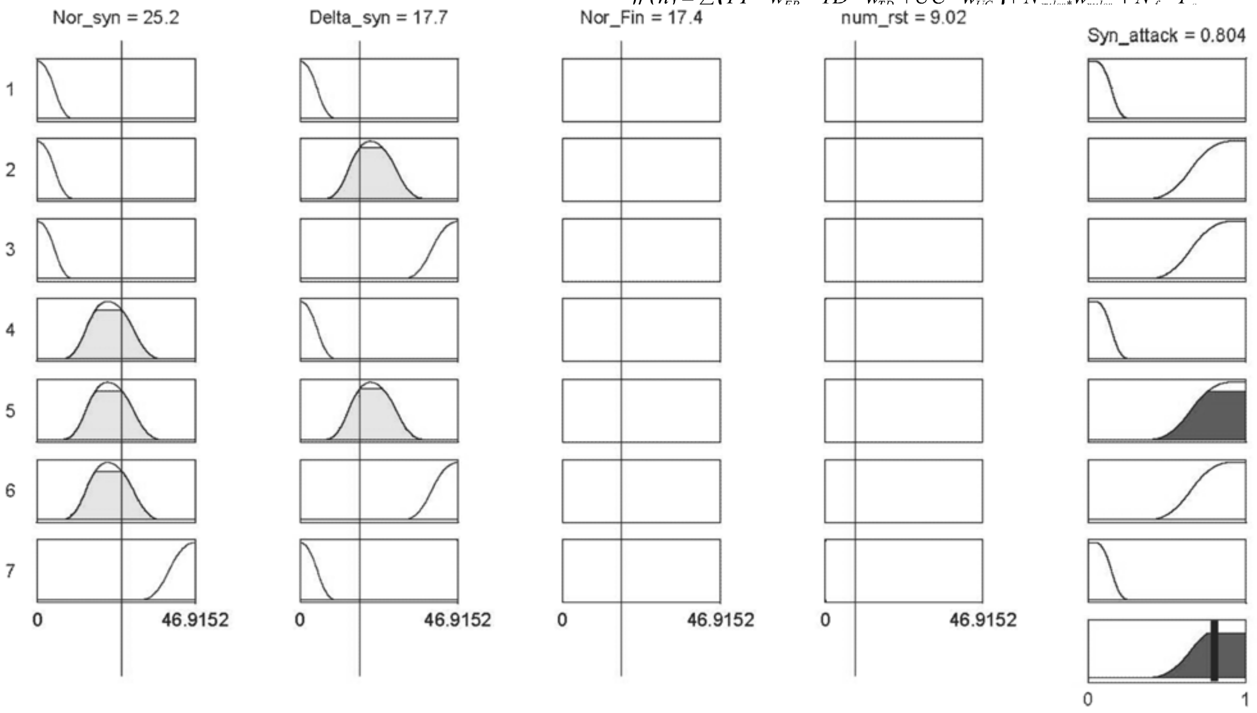


FIG. 4. FUZZY CONTEXT REPRESENTATION (SNAPSHOT OF C1 AT INTERVAL NUMBER 2000)

3.2.3 Context Database and Switching Logic

Evolved optimized contexts are stored in Context Database. Contexts are applied to fuzzy inference engine in real time depending upon the network conditions, intrusion phase or according to network security policies. These contexts are stored with attributes i.e. fitness values, associated number of rules, number of features, number of false positives and detection rate. Evolved contexts are switched according to organi-zation's security policies, current network condition, previous intrusion state and state of current active context. For example if the scanning activity is detected then next possible attack might be DDoS so a context optimized for detecting scanning and DDoS attacks will be used. Change in security setting will also result in context switching depending upon the security policies. Fig. 5 shows rules governing the context switching mechanism. Network domain expert defines the context switching rules according to security policies and current local or global security conditions.

Fig. 6 shows fuzzy context switching for DOS attack class. Context C0 is initial pre-computed context based on expert's domain knowledge. Membership function parameters for this context are initialized with 25%-overlapped values. Master fuzzy agent and context C0 are active and classifies the network traces from record zero. Genetic algorithm module evolves the context C1 using network traces classified by master fuzzy agent and local fuzzy agent (using the context C0). Initial context i.e. C0 is replaced within specific time (Time required to classify 200 records), if it is not switched by

If attack status is high directed to a target with different security settings
Switch to the context optimized for that security settings
If previous intrusion state is high and active LFC produces an UC
Switch to the context optimized for high detection rate
If Current context's classification differs from MFC's classification
Switch to the context optimized for no-false- positives.
If traffic profile (the output of the EMA module) has changed while no intrusion is detected
Change 'context' to high detection

FIG. 5. CONTEXT SWITCHING RULE SET

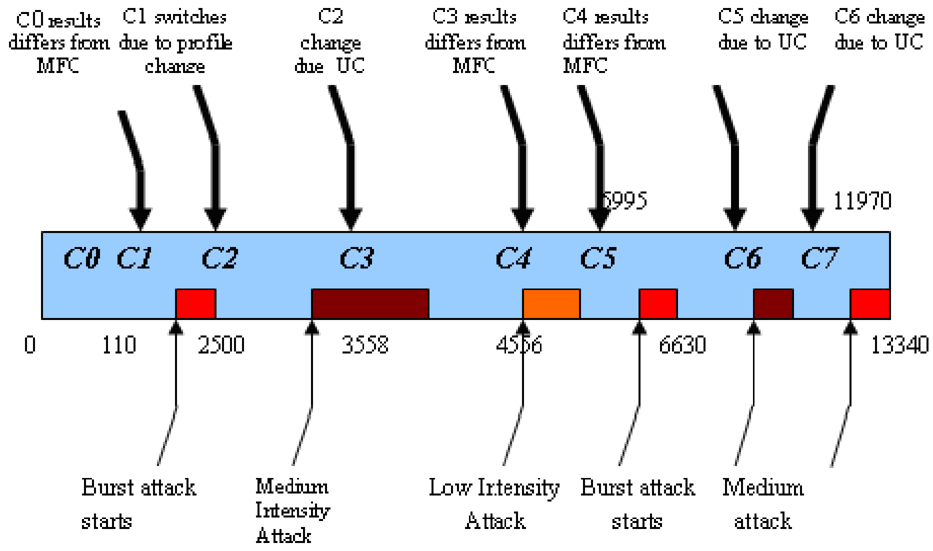


FIG 6. FUZZY CONTEXT SWITCHING AND INTRUSION PHASES

context selection algorithm. In Fig. 6, at record number 110, C0 is switched with C1 due to a false positive with reference to master fuzzy context, governed by context switching algorithm also shown in Fig. 8(a). Meanwhile Context C1 is evolved using records 0-109. It is switched due to change in intrusion phase at record number 2500, while C2 is switched when it fails to classify a record and results of both master and currently active local fuzzy agent differ. C4 is activated at record number 4556 due to change in intrusion phase and anomaly between LFA and MFA. Context selection algorithm discussed above governs all context switching.

3.3 Management Interface and Information Sharing Module

The management interface takes the results from each detection engine and presents a global picture current security state to the security professionals. It is also proposed that context evolved by context generation and evolution module can be shared through this module among other intrusion detection systems working under other autonomous system. This will help other IDS in earlier detection fast propagating worms.

4. Simulations Results

The DARPA Datasets 98/99 [20] has been widely used as the benchmark for evaluating Intrusion detection systems. The DARPA Datasets consists of Network traffic traces collected by simulating an Air Force network. Labeled attacks including denial of service, reconnaissance attack, U2R and R2U are embedded into simulated network traffic. The proposed framework is tested here for Neptune attack using TCP-dump data for Tuesday week 3, Thursday week5 of DARPA Dataset 98 and Wednesday of DARPA data set 99. The data set has three levels of Neptune attack namely; low, medium and burst or high intensity attack. Attacks with more than 300 SYN's per interval are classified as high intensity attacks, while less than or equal to 20 SYN's are termed as low profile SYN attack. There are three intrusion phases i.e., high intensity, medium intensity and low intensity attacks.

The fuzzy agents uses features such as number of connected sessions, number of SYN's, number of FIN's and number of RST, collected in a two-second intervals. These features including derived feature i.e. delta_SYN (number of SYNs minus number of connected-sessions) are mapped to fuzzy linguistic vari-ables. Master fuzzy agent uses all four features while local fuzzy agent uses only two features i.e. number of SYN,s and delta-SYN. Master and local fuzzy agents optimized for detecting Neptune attack are tested using same constructed data set. Fig. 7 shows the detection results of a master fuzzy agent. The results show that the detection rate for burst attack and medium intensity attack is one where low intensity attack detection rate is 0.7. It produced four false positives, one false negative and failed to classify three records. Fig. 8(a-h) and Table 1 shows the detection result of different contexts C0-C7 active in different time windows detecting different attack phases. These contexts are evolved for two objective; higher DR and minimizing number of rules. Each context detects single phase of intrusion.

C0 is pre-initialized context, which declared a normal record as anomalous shown in Fig. 8(a). Context C1 detects burst attack has detection rate equal to one shown in Fig. 8(b). Detection rate for low profile attack active during context C4 is 0.9 with single false positive. Medium and high intensity attacks active during C2-3, 6, and C1, 5, 7, respectively is greater than 0.9995. Although it has some uncertain records but the proposed approach produced better results than CUSUM algorithm specifically for low intensity Nep-tune attack [21].

Table 1 summaries the number of FP, NF negatives number of UC, context evolution time and generation-population product (G*P), number of rules, number of features used and fitness values for all context C0-C7 and MFC. It shows that MFC took an average of 1800 G*P product (number of Generation (G) * Population size (S)) and approximately 21hours of evolution time to achieve a solution with fitness value equal to 13. The time and G*S product increases exponentially for lower fitness values i.e. for better results. Comparing this with local context's C1-C7, which are all evolved using only 254 Generation-Population (G*P) product and in 78 seconds of evolution time, giving much better performance with overall fitness value equal to 6. It also shows that with almost same number G*P product, framework has constructed contexts by reducing 2-5 rules with overall fitness value equal to12. Context with minimum rules will help in detecting intrusion close to wire-speed.

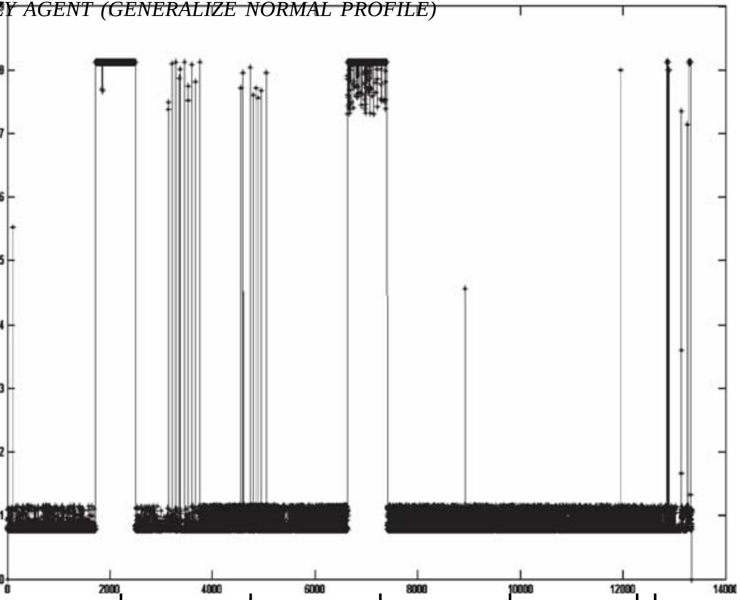
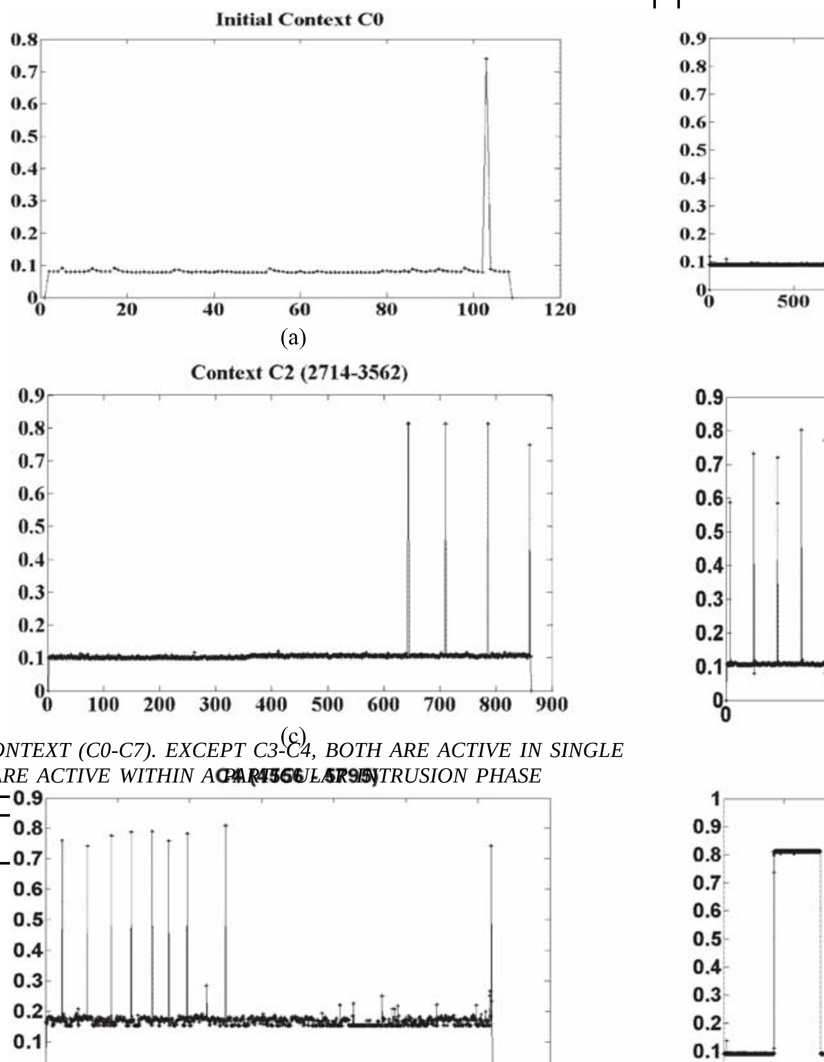


FIG. 7. DETECTION RESULT OF MASTER FUZZY AGENT (GENERALIZE NORMAL PROFILE)

TABLE 1 CONTEXT EVOLUTIO

Context/Opt		FP	FN	UC					
MFC		4	1	3					
C1	DR	0	0	0					
	Rules	0	0	0					
C2	DR	0	0	1					
	Rules	0	0	2					
C3	DR	0	0	0					
	Rules	0	2	0					
C4	DR	1	0	0					
	Rules*	0	1	2					
C5	DR	0	0	1					
	Rules	0	1	2					
C6	DR	1	0	0	0.9974	4(<1 sec)	12	2	2
	Rules*	0	0	0	1	16	8	2	0
C7	DR	0	0	0	1	27(9 sec)	12	2	0
	Rules	0	0	0	1	104	9	2	0

FP is Number of False Possitives, FN is Number of Flase Negatives, S is Population Size, DR is Detection Rate, and G is Number of Generation



(a)

(c)

FIG. 8(A-H). DETECTION RESULT FOR DIFFERENT FUZZY CONTEXT (C0-C7). EXCEPT C3-C4, BOTH ARE ACTIVE IN SINGLE INTRUSION PHASE AND C0 ALL OTHER CONTEXTS ARE ACTIVE WITHIN A PARTICULAR INTRUSION PHASE

5. CONCLUSIONS

Detection accuracy of anomaly based intrusion detection system depends upon correctness of their normal profile. Constructing precise and accurate normal profile for a network with dynamic network traffic and adapting this profile to accommodate the concept drift without introducing false positives is a key objective of AIDS. The article constructed two types of normal profiles; generalize or global normal profile and local normal profile. The results showed that generalize profile based on offline data-set helped in detecting global trends and the other local profile detects different phases of intrusions with less false alarms. The article constructs different local normal profiles specialized for detecting anomalies for particular network conditions, and switching these profiles according to network patterns, enhanced detection accuracy con-siderably. Furthermore it is proved that constructing normal-profiles for specific (limited) traffic pattern is easy comparing to a global, generalize normal profile. Experimental results show that the combined mu-tually inclusive (intersection) results of master normal profile and local profiles gave much better per-formance with no false positive but only uncertain records, which can be resolved with sufficient previous records.

ACKNOWLEDGEMENT

The authors acknowledge the support of HEC Pakistan for this research through its indigenous-300 S&T scholarship program.

REFERENCES

[1] Symantec, "Rise in Data Theft, Data Leakage, and Targeted Attacks Leading to Hackers", In Financial Gain news Release S. Reports, 2007.

[2] Halme, L.R., "AIN'T Misbehaving-A Taxonomy of Anti-Intrusion Techniques", Computers and Security, volume 40, No. 7, pp. 606, 1995.

[3] Lee, W. and Stolfo S.J., "A Framework for Constructing Features and Models for Intrusion Detection Sys-tems", ACM Transaction on Information and System Security, Volume 3, No. 4, pp. 227-261, 2000.

[4] Vasilios, A.S. and Papagalou F., "Application of anomaly detection algorithms for detecting SYN flooding attacks", Proceedings of IEEE communications Society Globecom, 2004.

[5] Wang, H., Zhang D., and Shin K.G., "Change Point monitoring for detection of Dos attacks", IEEE Trans-action of dependable and secure computing, volume 1, No. 4, 2004.

[6] Barbara, D., Couto J., Jajodia S., and Wu N., "Special section on data mining for intrusion detection and threat analysis: Adam: a test-bed for exploring the use of data mining in intrusion detection", ACM SIGMOD Record volume 30, pp. 15-24, 2001.

[7] Barbara, D., Wu N., and Jajodia S., "Detecting Novel Network Intrusions Using Bayes Estimators", Pro-ceedings of First SIAM International Conference on Data Mining, SDM 2001, Chicago, USA, 2001.

[8] Yoshida, K., "Entropy based intrusion detection", Proceedings of IEEE Pacific Rim Conference on Com-munications, Computers and Signal Processing (PACRIM2003), 2003.

[9] Lee, W., Stolfo S.J., and Mok K.W., "Mining audit data to build intrusion detection models", in Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, KDD '98, New York, NY, USA, 1998.

[10] Botha, M. and Solms R.V., "Utilizing fuzzy logic and trend analysis for effective intrusion detection", Computers & Security, Volume 22, No. 5, pp. 423-434, 2003.

[11] Gomez, J. and Dasgupta D., "Evolving fuzzy classifiers for intrusion detection", Proceedings of in Pro-ceedings of the 2002 IEEE Workshop on the Information Assurance, West Point, NY, USA, 2001.

[12] Pillai, M.M., J.H.P E., and H.S V., "An Approach to Implement a Network Intrusion Detection System using Genetic Algorithms", Proceedings of SAICSIT, 2004.

[13] Crosbie, M., "Applying genetic programming to intrusion detection", Proceedings of AAAI Fall Symposium series, 1995.

[14] Zitzler, E. and Thiele L., "Multi-objective Evolutionary Algorithms: A comparative Case Study and the Strength Pareto Approach", IEEE Transaction on Evolutionary Computation, volume 3, No. 4, pp. 257-271, 1999.

[15] Dickerson, J.E. and Dickerson J.A., "Fuzzy network profiling for intrusion detection", Proceedings of 19th International Conference of the North American Fuzzy Information Processing Society, Atlanta, USA, 2000.

[16] Liao, Y., Vemuri V. R., and Pasos A., "Adaptive anomaly detection with evolving connectionist systems", Network and Computers Applications, Volume 30, No. 2007, pp. 60-80, 2005.

[17] Abadeh, M.S., Habibi J., and Lucas C., "Intrusion detection using a fuzzy genetics-based learning algorithm", Journal of Network and Computer Applications, Volume 30, No. 2007, pp. 414-428, 2007.

[18] Tsang, C.-H., Lwong S., and Wang H., "Anomaly Intrusion Detection using Multi-Objective Genetic Fuzzy System and Agent-based Evolutionary Computation Framework", Proceedings of Fifth IEEE International Conference on Data Mining (ICDM'05), 2005.

[19] Yen, J. and Langari R., "Fuzzy Logic: Intelligence, Control and Information", Prentice Hall, Upper Saddle River NJ, 1999.

[20] DARPA, "Darpa 98/99 Data Set": MIT Lincoln Labs, 1998.

[21] Baig, H.U. and Kamran F., "Detection of Low Intensity DoS attacks using Fuzzy Intrusion Detection System", Proceedings of ICICE Conference, Dhaka Bangladesh, 2006.