Efficient FSM Techniques for IDS to Reduce the System Attacks

Dr.M.Sadiq Ali Khan
Department of Computer Science
University of Karachi

Dr.S.M.Aqil Burney
Meritorious Professor
Department of Computer Science
University of Karachi

ABSTRACT
The main purpose of this research is to introduce the new techniques of Finite State Machine (FSM), mainly DFA and PDA to filter the error attacks of an Intrusion Detection System. The main purpose of implementing the PDA model in attacks is to minimize the space required which is important issue in network when attacks are push and slow down the network traffic. The data transfer from Intrusion Detection Model in a certain time on random basis, may the system will hang due to huge amount of data but if we will use Time Management Automata we can easily remove such problem.

Keywords: Finite State Machine; Deterministic Finite Automata; Push Down Automata; Intrusion Detection System

1. INTRODUCTION
Network Security whether in a commercial organization or in a critically important research network, is a major issue of concern with the increasing use of web even the personal information in under threat. Efficient network intrusion detection system is only solution to such threats [1]. IDS is a monitoring system of networks to control / avoid / secure the networks from cyber terrorist or it is the process of examine the events occurring in a network or computer system and detecting the signs of incidents which are the threats of computer security Policies [2]. KDD-99 data set usually used for investigating the nature of attack. The data set has 41 features listed. Here we discuss the Finite State Machine model to minimize the error rates in proposed IDS system. The proposed system deals with the attacks of error may be known or unknown in nature as shown in Figure 1 below

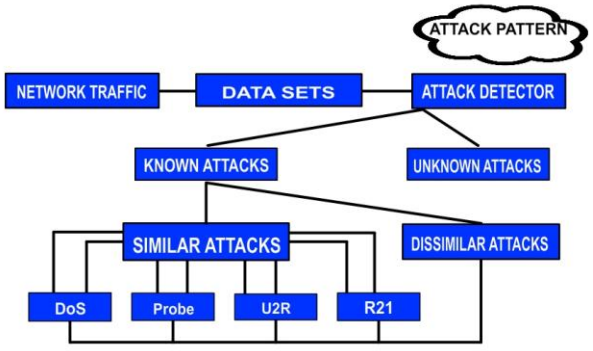


Figure 1: Intrusion Detection Model

The proposed system monitoring and identifying all the four general types of error as the types of identified which are *DoS, Probe, U2R and R2L*.

2. BACKGROUND
As network requirements increases, degree to major the vulnerability to attack becomes more important towards automation [3]. While considering the security of the network, isolated vulnerabilities consideration is not quite enough [4]. Network at large scale having multiple platforms specifically have some definite loop holes. So we should have such a secure monitoring system that efficiently analysis network traffic for any vulnerability. When we model a system working in irregular surroundings, certain transitions in the model represent the system's response to changes in the surroundings [5, 6]. When no observed information is on hand about the comparative probability of such surroundings-driven transitions, we can model them as nondeterministic. We specify the network model as FSM, in which every state transition corresponds to a single tiny attack by the attacker. If an attack occurs the system transition from one given state to another state where the effects of such attack holds [7]. Attackers are trying to penetrate in the network by following some sequence of state transitions.

2.1 Finite State Machine
An FSM acts as a model of behavior comprises of states, transitions and actions. State stores information and reflects the input changes; a transition refers the state change depending on the input and an action is what so ever a system performed in response to the input. It is Deterministic Finite Automata; it is type of graph / network diagram/tree used to recognize certain errors [8,9]. DFA machine is basically based on five tuples machine which are $(Q, \Sigma, \delta, q_0, F)$. There is no doubt in navigating nodes in DFA and there are a preset and identified number of nodes and branches. In this the transitions from one end to other is solely dependent upon the input value. The automata are deterministic, that is, once their state and input are given, their evolution is uniquely determined.

2.2 Non Deterministic FA
The formal definition of NFA is defined as (Q, I, δ, q_0, F) where $Q \rightarrow$ No. of states; $I \rightarrow$ Symbols of input ; $\delta \rightarrow$ incomplete switching function; δ: $Q \times I \rightarrow P(Q)$, where $P(Q)$ is the set of all subsets of Q; $q_0 \rightarrow$ starting state ; $F \rightarrow$ final states set.

The above description states that a NFA can reveal numerous dissimilar shifting sequences for a specified node and a known input. In NFA we have more than one path available, successful path is that which leads us to a final state.

2.3 Push Down Automata

These are abstract devices defined in automata theory, and have access to a potentially unlimited amount of memory in the form of a single stack as comparing to Finite Automata[10].It exist in deterministic or non deterministic varieties and accepts a formal language. A Push Down Automata (PDA) is a seven tuple model P= (Q, q_0, q_f, I, Z_0, δ, F) where Q →set of states; q_0→initial state; q_f→final state; I→input;Z_0→top of stack; Γ→state input; δ → transition function

Transition mapping function Q x (I U{ Є }) x Γ to finite subsets of Q x (Γ U{ Є })*, q_0 Є Q is the initial state which is unique in nature, Z_0 Є F is the starting of the stack. Three types of transition may be referring.

(1) Є-transition
(2) Push- transition
(3) Pop – transition

Є -transition is performing when stack is empty. Push transition is perform when data in enter (either useful data or error). Similarly pop transition is filter out all errors from useful data.

3. METHODOLOGY

The Finite Automata for the proposed system is

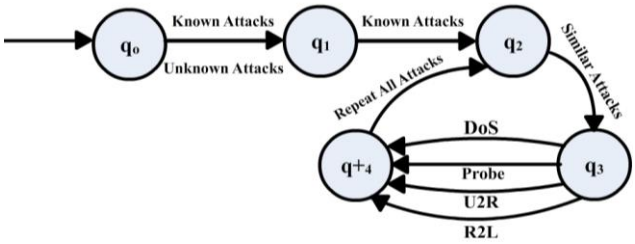


Figure 2 a: Similar Attack Model.

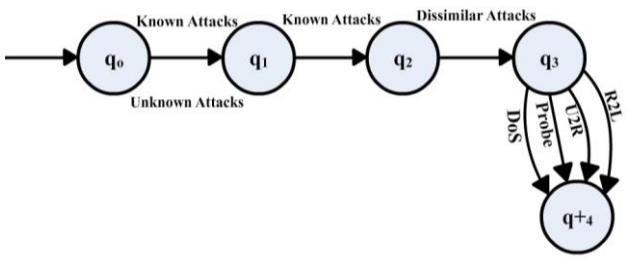


Figure 2 b: Dissimilar Attacks Model

In proposed IDS System the probability of attacking errors is 50% for each known and unknown attacks, so the Finite State Machine for IDS will be NFA model and NFA model is not efficient model for the implementation. According to theory of the Finite State Machine every NFA model has equivalent DFA model using the same concept [11, 12]. We can easily convert given NFA model into minimize DFA to make model efficient and faster. We can easily convert given NFA model into minimize DFA to make model more efficient and faster. The purpose of joining is to minimize the states and removes replicate states.

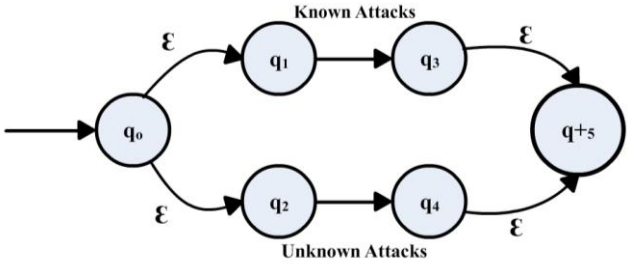


Figure 3: Combined FSM

The transition table for given machine will be as follows:

Table 1: Transition table for Combined FSM

	Known Attacks	Unknown Attacks
q_o	q_3 q_5	q_4 q_5
q_1	$q_3 q_5$	⊥
q_2	⊥	q_4 q_5
q_3	⊥	⊥
q_4	⊥	⊥
q_5	⊥	⊥
{q_3 q_5}	⊥	⊥
{q_4 q_5}	⊥	⊥

The attacks may be of four types further classified into Similar or Dissimilar attacks. Now these attacks will be added in minimized DFA as shown above. The NFA- **Є** will be as follows in Figure 4.

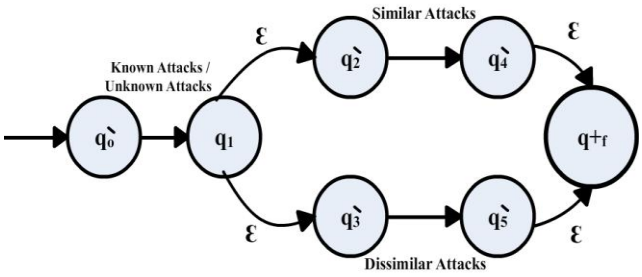


Figure 4: NFA-Є machine

By transition mapping function four new states are formed. After minimization following DFA will be formed as in Figure 5.

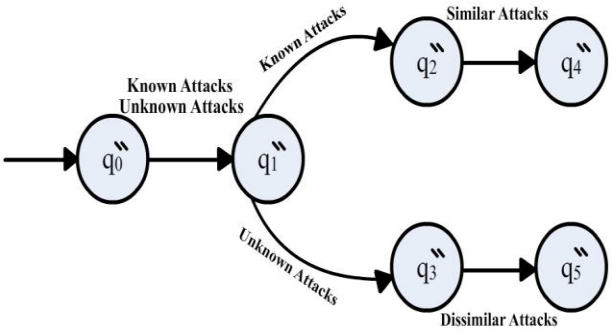


Figure 5: Minimized DFA machine

As similar and dissimilar attacks may be of four types. The NFA will be as follows in Figure 6

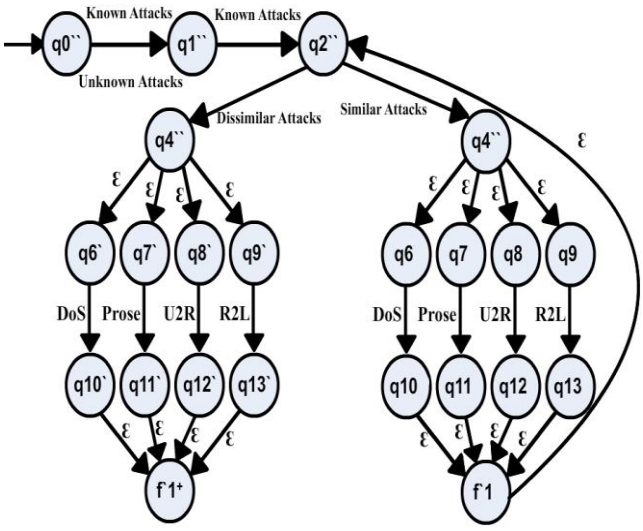


Figure 6: NFA Model

Using the similar algorithm of NFA-Є with all combination of Figure 6 consists of 23 states will be reduced in five states which may detect the intrusions quickly as shown in proposed model (Figure 2). To store the records of incoming errors in DFA machine we can use Pushdown Automata to store the record of all attacks and their types in stack.

PDA model can easily be implemented to increase the efficiency of IDS. Following is the general model of proposed system.

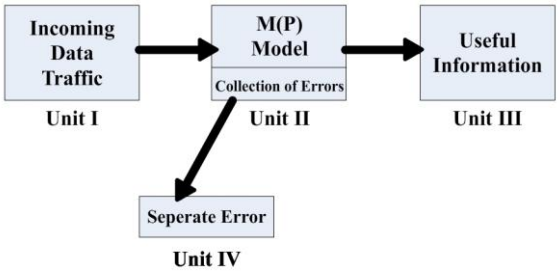


Figure 7: Generalized Proposed PDA model

The monitoring system extracts the relevant information and separates the errors by using PDA. In proposed system model (as in Figure 7) incoming data from Unit-I is push on and Unit-II i.e M(P), generates a sequence of useful data along with errors and finally filter or pop all errors separately. As the Unit-II based on stack model implementation. These errors are easily identified and separate from sequence of data.

Let take configuration of data send to Unit–II (MP) after processing the sequence. The new state of the system is succ(C,a) can be define as

Succ(C,a) → {Dn`\| Є 3c C,c => and}

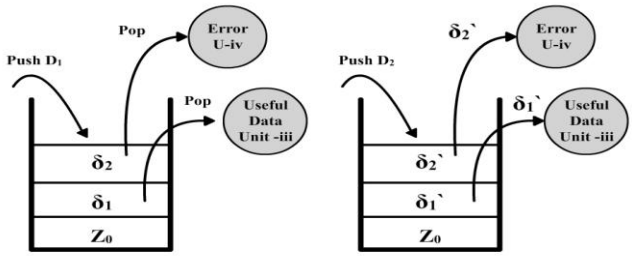
Suppose Dn is a sequence of incoming data

Dn → D_1,D_2,D_3……..

D1 → $\delta_1 + \delta_2$

δ_1 → is useful data

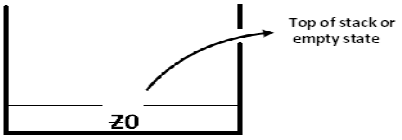
δ_2 → error



Minimization process will be defined as follows by using PDA

Step 1:

In first step the stack of PDA is empty i-e 'Z_0', the attacks are inserted into the top of the stack 'A'. Attack **A** => 'known' or 'Unknown'



Step 2:

When stack is empty A_0 or A_1 has equal probability to push into stack, the probability of A_0 or A_1 is also equal pushing as A_0, A_1. Complete transitions are mentioned in Table 2.

Table 2: Complete Transition Table

	Known Attack (A_0)	**Unknown Attack (A_1)**
Z_0	**Push(Z_0 ,A_0)**	**Push(z_0,A)**
A	**Push(A_0,A_0)**	**Push(A_0,A_1)**
A	**Push(A_1, A_0)**	**Push(A_1,A)**

Table 3: PDA for attacks

	DoS	Probe	U2R	R2L
Z_0	Pop(Push(Z_0 , DoS))	Pop(Push(, probe))	Pop(Push(Z_0, U2R))	Pop(Push(Z_0, R2L))
A_0	Pop(Push(A_0 , DoS))	Pop(Push(A_0 , probe))	Pop(Push(A_0, U2R))	Pop(Push(A_0, R2L))
A_1	Pop(Push(A_1, DoS))	Pop(Push(A_1, probe))	Pop(Push(A_1, U2R))	Pop(Push(A_1, R2L))
A_2	Pop(Push(A_2 , DoS))	Pop(Push(A_2 , probe))	Pop(Push(A_2, U2R))	Pop(Push(A_2, R2L))
A_3	Pop(Push(A_3, DoS))	Pop(Push(, probe))	Pop(Push(A_3, U2R))	Pop(Push(A_3, R2L))

The main purpose of implementing the PDA model in attacks is to minimize the space required which is important issue in network when attacks are push and slow down the network traffic. By using this model efficiency of network model can be increased. This can be further extended by using the double stack or n^{th} stack model as in Figure 8.

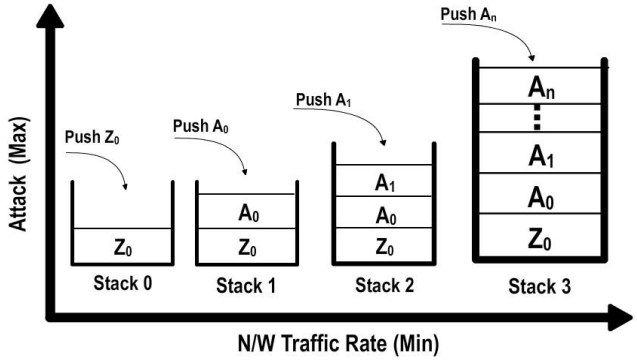


Figure 8: Double Stack or n^{th} Stack Model

To represent an idealized machine a computational model designed which refers as a Deterministic Finite Automaton (DFA)/PDFA. Particularly, DFAs are considered to identify member strings of a particular language. Formally they recognize regular languages, in our case we want to recognize the language of DoS attacks. PDFA /DFAs bear two significant properties. First, they represent a *finite state.* Second, they are Deterministic, meaning that there next state will be determined by the previous state and transition must be unique. DFAs are usually represented by State-Transition Diagrams. The different automaton's states are identified as vertex and unidirectional arrows represent inputs that may permit transitions between different states [10].

Usually final states which refer as accepting state for the input string are double circled. Also, for every state, there must be a one outgoing edge for every input alphabet. Figure 10 refers as a simple PDFA/DFA which accepts inputs of the alphabet {a, b} having the string "aba". In this illustration, q1 is the initial state and q4 is the accepting state. If the PDFA/DFA is in state q1 and takes "a" as an input, then the PDFA/DFA jumps to next state q2. If while in state q2 the PDFA/DFA receives input "b", then it moves to next state q3. If an "a" is received while in q3, the PDFA/DFA then travels to the accepting state and the input string is accepted. An assessment of Figure 10 explains that any input disrupting the pattern "aba" sends the PDFA/DFA back to its initial state. While the DFA in Figure 10 has only one final state, this is not a specification for PDFA/DFAs in general. A PDFA/DFA can have more than one final state.

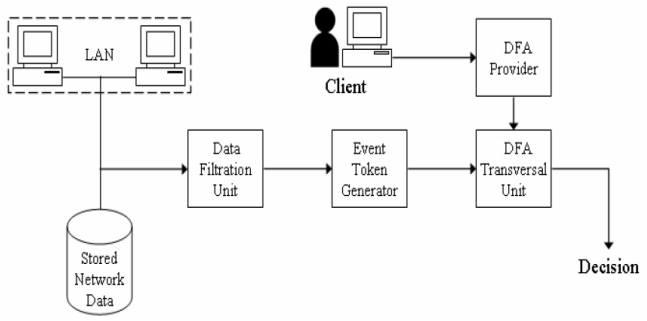


Figure 9: Conceptual Model

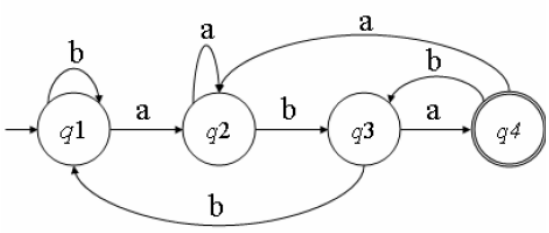
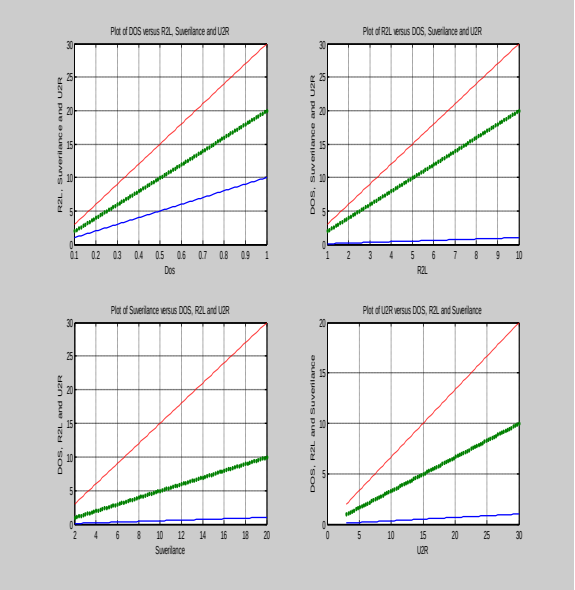


Figure 10: Simple Illustration of PDFA/DFA

4. RESULT & DISCUSSION

The data transfer from Intrusion Detection Model in a certain time on random basis, may the system will hang due to huge amount of data but if we will use Time Management Automata we can easily remove such problem. The implementation of such data on simulation software verifies this statement. Following are the result from the Simulator.

Table 4: Simulation Results



Experimental results obtained after applying simulation on the basis of tested dataset, containing 311,029 numbers of records. Here, we are only interested in knowing to which category (*Normal, DoS, R2L, U2R, and Probing)* a given connection belongs. The accuracy of each experiment is based on percentage of successful classification (PSC) on the test dataset, where

DOS	Probe	R2L	U2R
Known			
Back, land, Neptune, Pod, smurf, teardrop	ipsweep, satan, nmap, portsweep	ftp_write, guess_passwd, warezmaster, warezclient, imap, phf, spy, multihop	rootkit, loadmodule, buffer_overflow, perl

From Table 5, we can see that the last two classes R2L and U2R are not well detected. The low presence of these two classes of attacks in the training dataset accounts for the poor detection. The percentage of R2L and U2R in the dataset are 0.23% (1,126 records) and 0.01 (52 records) respectively.

Data Samples

D = Columns 1 through 10

0.6020 0.2630 0.6541 0.6892 0.7482 0.4505 0.0838 0.2290 0.9133 0.1524

R = Columns 1 through 10

0.8258 0.5383 0.9961 0.0782 0.4427 0.1067 0.9619 0.0046 0.7749 0.8173

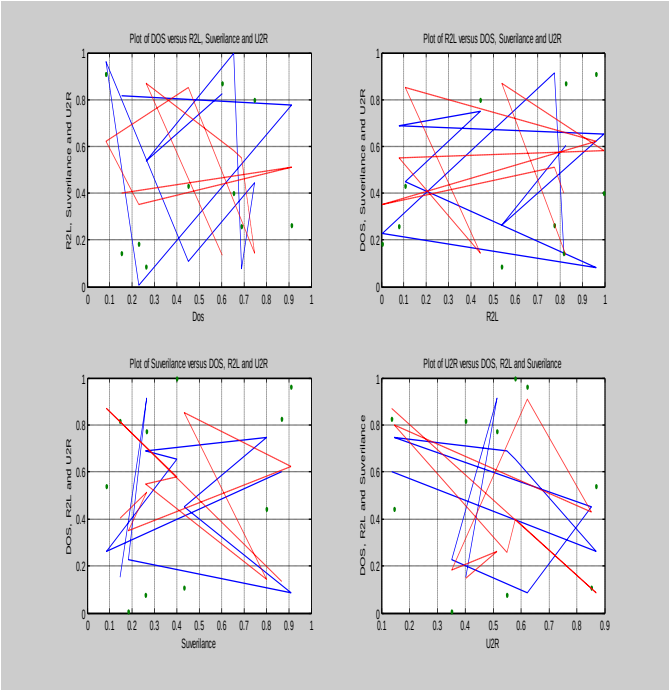
S = Columns 1 through 10

0.8687 0.0844 0.3998 0.2599 0.8001 0.4314 0.9106 0.1818 0.2638 0.1455

U = Columns 1 through 10

0.1361 0.8693 0.5797 0.5499 0.1450 0.8530 0.6221 0.3510 0.5132 0.4018

Table 5: Detection Graph



5. CONCLUSION

FSM is an efficient technique to check the incoming errors of system and produce the efficient flow of data rate without any interrupt. The use of PDA further refine the quality of incoming data by this mean one can store huge data in existing system which helps us to manage the error attacks in order, so that IDS easily be reduce the error rate. There is another issue when the rate of known and unknown attacks is high we cannot maintain the handling because of stack or memory, therefore we used Push Down Automata model to maintain the frequent flow of attacks. In PDA models we use double stack, first stack is use to store the recursive flow of attacks and second stack is use for traffic of non-recursive attacks. Finally the filter process of model will reduce the flow of dynamic attacks and smooth the flow of data in network model which increase the performance of entire IDS.

6. REFERENCES

[1] S.M.Aqil Burney and M.Sadiq Ali Khan (2010); "Network Usage Security Policies for Academic Institutions", International Journal of Computer Applications, October Issue, Published By Foundation of Computer Science.

[2] S.M.Aqil Burney; M.Sadiq Ali Khan and Mr.Jawed Naseem (2010); "Efficient Probabilistic Classification Methods for NIDS"; (IJCSIS) International Journal of Computer Science and Information Security,Vol. 8, No. 8, November 2010

[3] Roschke, S.; Feng Cheng; Meinel, C.(2009), "An Extensible and Virtualization-Compatible IDS Management Architecture", Fifth International Conference on Information Assurance and Security IAS '09.Volume: 2, DOI 10.1109/IAS.2009.151 Page(s): 130 – 134.

[4] Zaman, S.; Karray, F.(2009), "Lightweight IDS Based on Features Selection and IDS Classification Scheme", International Conference on Computational Science and Engineering, CSE '09, Volume: 3, DOI: 10.1109/CSE.2009.180, Page(s): 365 - 370.

[5] Orfila, A.; Carbo, J.; Ribagorda, A.(2003), "Fuzzy logic on decision model for IDS", The 12th IEEE International Conference on Fuzzy Systems, Volume: 2, DOI 10.1109/FUZZ.2003.1206608, Page(s): 1237 - 1242

[6] Murakami, M.; Honda, N. (2009), "Development of an IDS hardware unit for real-time learning applications", International Conference on Fuzzy Systems, DOI 10.1109/FUZZY.2009.5277072, Page(s): 227 – 233.

[7] Mathew, B.K.; John, S.K.; Pradeep, C.(2008), "New Technique for Fault Detection in Quantum Cellular Automata", First International Conference on Emerging Trends in Engineering and Technology ICETET '08, DOI 10.1109/ICETET.2008.186, Page(s): 834 – 837.

[8] Preston, K., Jr.(1990), " Detection of weak, sub pixel targets using mesh-connected cellular automata", IEEE Transactions on Aerospace and Electronic Systems, Vol 26 Issue 3, DOI 10.1109/7.106134, Page(s): 548 - 559

[9] Yasami, Y.; Mozaffari, S.P.; Khorsandi, S.(2008), "Stochastic learning automata-based time series analysis for network anomaly detection", International Conference on Telecommunications ICT 08, DOI 10.1109/ICTEL.2008.4652664.Page(s): 1 – 6.

[10] Zong-Fen Han; Jian-Ping Zou; Hai Jin; Yan-Ping Yang; Jian-Hua Sun(2004), " Intrusion detection using adaptive time-dependent finite automata", International Conference on Machine Learning and Cybernetics Volume: 5, DOI 10.1109/ICMLC.2004.1378554 , Page(s): 3040 - 3045 vol.5

[11] Sinaie, S.; Ghanizadeh, A.; Majd, E.M.; Shamsuddin, S.M.(2009), "A Hybrid Edge Detection Method Based on Fuzzy Set Theory and Cellular Learning Automata", International Conference on Computational Science and Its Applications ICCSA '09, DOI 10.1109/ICCSA.2009.19, Page(s): 208 - 214

[12] Toony, Z.; Jamzad, M.(2010), "A modified saliency detection for content-aware image resizing using cellular automata ",*International Conference on* Signal and Image Processing (ICSIP), *DOI* 10.1109/ICSIP.2010.5697464, Page(s): 175 – 179.