International Journal of Information & Network Security (IJINS)
Vol.2, No.6, December 2013, pp. 492~498
ISSN: 2089-3299

The Prospects of Using Spiking Neural P System for Intrusion Detection

Rufai Kazeem Idowu*, Ravie Chandren Muniyandi*, Zulaiha Ali Othman*
* School of Computer Science, Faculty of Information Science & Technology,
Universiti Kebangsaan Malaysia

Article Info

Article history:

Received Jun 12th, 2013
Revised Aug 20th, 2013
Accepted Nov 16th, 2013

Keyword:

Spiking Neural P System
Intrusion Detection
Membrane Computing
Parallel Computation
Cyber-security

ABSTRACT

Spiking Neural P (SN P) System is one of the variants of Membrane computing. SN P system is a parallel computing model which derives its motivation from the biological living cells. On the other hand, 'Intrusion' issue has become a major concern not only to the cyber security experts but also to all the users of the internet. Therefore, to totally eradicate this menace or putting it in a state of abeyance, several approaches like the use of Expert system, Intelligent algorithms, Artificial Neural Networks, Statistical methods and a host of others had been deployed. However, there is still room for improvement. SN P system being a maximally parallel biological model, has proved to be a versatile tool. This paper therefore attempts to evoke a new direction in the application of SN P to intrusion detection. Specifically, it answers the following questions among others: What are the principles of intrusion detection? What are the approaches being used and the challenges impeding the realization of an efficient Intrusion Detection System (IDS)? What is an SN P system? Does SN P syetem have the potentials to enhance the performance of IDS? In all, the paper points to a new direction for using SN P systems in detecting known and unknown attacks in Intrusion detection systems thereby providing the baseline for future works.

Corresponding Author:

Rufai Kazeem Idowu,
School of Computer Science,
Faculty of Technology and Information Science,
Universiti Kebangsaan, Malaysia.
Email: ruffyk200l@yahoo.com

1. INTRODUCTION

There is no gain saying the fact that the rate at which networks and stand alone computers are connected to the internet increases astronomically on daily basis. Consequent upon these extended networks, attempts to breach information security of the systems are also on the rise. Primarily, the usual objective of the aforesaid attacks is to undermine the conventional security processes on the systems and perform actions in excess of the attacker's permissions [1]. Therefore, when a system is intruded, it suffers from;

(i) compromise of its integrity
(ii) denial of its availability
(iii) inefficiency in its performance.

Hence, in an attempt by concerned individuals and corporate bodies to find lasting solution to this intrusion imbroglio, different approaches have been introduced. Intrusion detection techniques are continuously evolving, with the goal of improving the security and protection of networks and computer infrastructures [2].

For example, Lee and Stolfo [3] in 1998 built a data-mining framework for intrusion detection system. Based on this extension, Hai et al. [4] in 2004 applied fuzzy data mining algorithm to intrusion detection. Also, Arafat [5] proposed new model for monitoring Intrusion based on Petri Nets. From another perspective however, in 2009, Shanmugam and Norbik [6] proposed the application of hybrid model of an advanced

fuzzy and data mining methods to find out both misuse and anomaly attacks. Other efforts came from Yu et al. [7], where they presented a three-layered collaborative architecture for multiple IDSs to detect real-time network intrusions. From another perspective, Ghosh and Schwartzbard [8] in 1999, conducted a study in which they used Neural Networks to build intrusion detection models. Amini and Jalili [9] proposed the use of unsupervised adaptive resonance theory towards detecting network-based intrusion. More recently, in 2012 precisely, Taghanaki et al. [10] attempted to primarily improve IDSs' accuracy by presenting a synthetic feature transformation using Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) with Radial Basis Function (RBF)-Neural Network as a classifier.

Painfully however, all these attempts and many more others appear to be efforts in futility because none of the intrusion detection approaches so far discovered is sufficient in its entirety to address all the likely threat a computer system may encounter.

It is in the light of the above that attention is now being drawn to an aspect of membrane computing, which is called Spiking Neural P (SN P) system to determine how suitable it could be in resolving the ever rearing ugly face of intrusion. SN P system which is a biologically inspired class of distributed parallel computing devices functions by the way neurons communicate. Apart from other intrinsic advantages of SN P systems, they have been proved to be computationally complete [11].

1.1. Intrusion Detection System Defined

An ***intrusion*** is any set of action which attempts to compromise the integrity, confidence or availability of resource. Simply put, an intrusion is a security threat deliberately done to access and/or manipulate information and to render a system unreliable or unusable.

Researches have shown that computer systems suffer from security vulnerabilities (especially intrusion) regardless of their purpose, manufacturer, or origin, and that it is both technically difficult and economically costly to build and maintain computer systems and networks which are not susceptible to attacks.

An Intrusion-Detection System (IDS) therefore, is a software product of hardware technology that automate a monitoring process of events which occur in a computer system or network with a view to analysing them for signs of intrusion. [12] defines IDS as a system which watches over networked devices and searches for anomalous or malicious behaviors in the patterns of activity in the audit stream.

In similar perspective, [13] submitted that an IDS is a system which dynamically monitors the action taken in a given environment, and decides whether or not these actions are symptomatic of an attack or constitute a legitimate use of the environment. The following figure 1 depicts the organization of a generalized IDS [14]. Solid lines indicate data/control flow, while dashed lines indicate responses to intrusive activities.

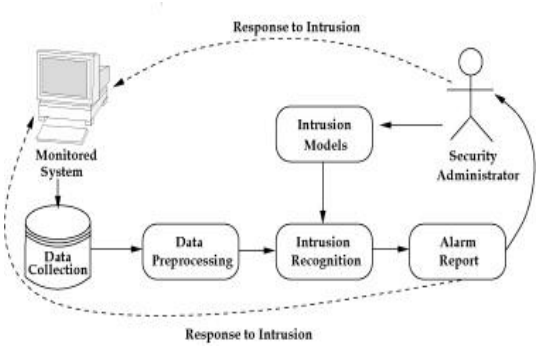

Figure 1: Organization of an IDS [14]

In all, a reliable and dependable IDS should only not work in real time, it must however be able to make distinctions between unauthorized use, misuse, abuse and legitimate use of the computer systems be it from within or outside the information system

1.2. Categorization of Intrusion Detection System

Generally, any Intrusion Detection System can be classified into the following three main categories:

Host-Based IDS: Host-based intrusion detection was the first area explored in intrusion detection [13]. It operates based on information found on a single or multiple host systems, including contents of operating systems, system and application files. So, it parses audit logs for evidence of suspicious or malicious

activity, monitors key system files for tampering. Although, it has the ability of detecting local attacks in near-real-time, but it is most often difficult to deploy and manage when many hosts are involved.

Network-Based IDS: Primarily, it monitors and evaluates information captured from a network traffic. This is done by analyzing the stream of packets traveling across the network. Packets are captured through a set of sensors. Majority of commercial intrusion detection systems are Network-based. It works by watching live packets, looks for attacks, misuse and anomalies. The main limitations of this IDS are that it cannot analyze encrypted information. Also, processing all packets may be overwhelming especially in a large/busy network, hence may suffer from unnecessary high alarms.

Vulnerability-Assessment IDS: It detects vulnerabilities on internal networks and firewalls. It usually analyzes the events transpiring within a software application. Generally, it uses application's transaction log files. However, it is more vulnerable to attacks as the applications logs are not as well-protected. Also, most often it monitors events at the user level of abstraction**.**

2. IDS PRINCIPLES, APPROACHES AND THEIR INHERENT FLAWS

2.1 IDS Principles

In the past till now, different methods were adopted and are still being used to build IDSs. However, the following two detection principles are peculiar to every IDS:

(a) Misuse-based (or signature-based): In this detection scheme, intrusions are flagged by matching observed data with already known descriptions of intrusive behaviour of attacks which exist in the signature database. It simply uses pattern matching technique to discover malicious packets. It has high detection accuracy for known attacks with minimal false positive rate (that is, an indication of error rates of mistakenly detected non-intrusive attacks). However, it is incapacitated to detect new attacks.

(b) Anomaly-based: This is premised on the principle that there is a perfect dichotomy between abnormal and normal behaviours. So, it raises alarm whenever the observed activities deviate significantly from the normal ones. Although, it has high tendency of detecting new attacks but most often, it raises false alarm. In all, the main distinction between signature- and anomaly-based detections is that the former models intrusions, while the later creates a model of normal 'use' and finds activity which runs contrary to it.

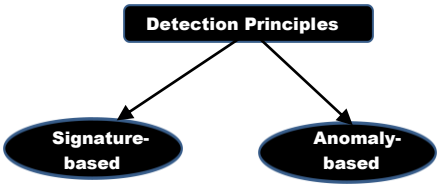

Figure 2. Groupings of Detection Principles

2.2 IDS Approaches and their flaws

Furthermore, some of the approaches used are here-under considered and their perceived challenges are also discussed.

A. *Intrusion Detection Expert System (IDES) Approach*

This is a comprehensive system which uses innovative statistical algorithms for anomaly detection as well as an expert system which encodes known intrusion scenarios [15]. IDES maintains profiles, which is a description of a subject's normal behavior with respect to a set of intrusion detection measures. Profiles are updated periodically, thus allowing the system to learn new behavior as users alter their behaviour. The profiles are kept so as to compare the user's behaviour and subsequently flagging deviations as intrusion. The advantage of this approach is that it adaptively learns the behavior of users, which is thus potentially more sensitive than human experts. However, its disadvantages include: (i) the system can be trained for certain behaviour gradually making the abnormal behaviour as normal, which may make the intruders undetected. (ii) Determining the threshold above which an intrusion should be detected is a bit herculean.

B. *Artifitiicial Neural Networks Approach*

Matching a user's behaviour to a model of the user's past behaviour could be a bit hectic, thereby creating room for false alarm. Artificial Neural Network (ANN) is promising field of research

as relates to IDS. ANNs have the ability of learning-by-example and generalization from limited, noisy, and incomplete data [13]. So, they learn the characteristics of misuse attacks and identify instances that are unlike any which have been observed before by the network. It has a high degree of accuracy to recognize known suspicious events. The main problem is in the training of neural networks, which is so germane if an efficient neural network is desirable. The training phase also requires a very large amount of data.

C. The Rule-Based Approach

In this approach, the main emphasis is identifying known vulnerabilities and attacks, and the greatest threat may be the vulnerabilities which have not yet been known or tried. An intrusion scenario that does not trigger a rule will not be detected by the rule based approach. Besides this, maintaining a complex rule-based system can be difficult as maintaing any other piece of software of comparable magnitude, especially if the system depends heavily on procedural extensions such as rule ranking and deleting facts. A typical rule has antecedent (condition) part and consequent (action) part. The format is:

IF
condition1, condition2, ...
THEN
action1, action2, . . .

D. *The Data Mining Approach*

Raw data is first converted into ASCII network packet information, which in turn is converted into connection level information. Data mining (DM) algorithms are applied to this data to create models to detect intrusions. Basically, DM employs the techniques of frequent pattern mining, classification, clustering and mining data stream. The main advantage of this system is the automation of data analysis through data mining, which enables it to learn rules inductively replacing manual encoding of intrusion patterns. The problem is that it deals mainly with misuse detection. Hence, some novel attacks may not be detected.

E. *Model-Based Reasoning Approach*

With this approach, one can develop specific models of proscribed activities. It is advantageous because it processes more data, it can predict what the intruder's next action will be. This approach is however limited because it looks for known intrusion vunerabilities and the attack that have not yet been tried.Some of the drawbacks are that the intrusion patterns must always occur in the behavior it is looking for and patterns for intrusion must always be distinguishable from normal behaviour and also easily recognizable.

3. THE FUNDAMENTALS OF SN P SYSTEM

SN P system is class of distributed and parallel computing model which is inspired by the neurophysiological behaviour of neurons sending electrical impulses (spikes) to other neurons. The set of neurons are placed in the nodes of a graph which facilitate the movement of the spikes along the synapses (edges of the graph), under the control of firing rules. For the main purpose of communication, these *neurons* are connected to each other in an intricate pattern. They have three functionally distinct parts called ***dendrites, soma*** and ***axon***. Hence, when they interact, there is an exchange of spikes. In doing this though, pre-synaptic neuron is configured to have a kind of 'handshake' with the post-synaptic neuron at a junction known as ***synapse*** by means of specific rules.

In general therefore, an SN P system of degree $m \geq 1$, is a construct of the form:

$\prod = (O, \sigma_1 \ldots . \sigma_m \text{ syn, out})$,

Where:

1. $O = \{a\}$ is the singleton alphabet called spike);
2. $\sigma_1, \ldots, \sigma_m$ are neurons, of the form $\sigma_1 = (n_i, R_i)$, $1 \leq i \leq m$, where:
   a) $n_i \geq 0$ is the initial number of spikes contained by the neuron;
   b) R_i is a finite set of rules of the following two forms:
      i) E/a^c a;d, where E is a regular expression over O, $c \geq 1$, and $d \geq 0$;
      ii) $a^s \rightarrow \lambda$ for some $s \geq 1$, with the restriction that as $L\epsilon$ for no rule E/a^c a; d of type (1) from R_i;
3. syn $\subseteq \{ 1, 2, \ldots ., m \} \times \{ 1, 2, \ldots ., m \}$ with $(i, i) \epsilon$ syn, for $1 \leq i \leq m$ (synapses);

4. out $\epsilon\{1, 2, \ldots, m\}$ indicates the output neuron.

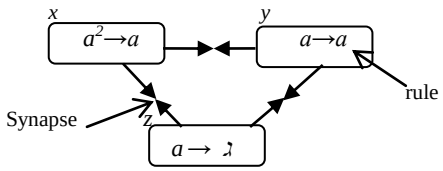

Figure 3. Schematic representation of how Neurons communicate

Figure 3 above depicts a simple schematic representation of an SN P system with three neurons *x* , *y* and *z*. The spike, denoted as "*a"* which is the basic unit of information is stored in the neuron. While neuron *x* has rule $a^2 \rightarrow a$, *y* has rule $a \rightarrow a$ and *z* has rule $a \rightarrow \lambda$ The synapse is also captured.

Furthermore, when the rules (which may be used concurrently) are applied, the system is transformed. By assuming the presence of a global clock, the system is synchronized. Atimes, the cell sending out spikes is "closed" during a refractory period of a neuron. At this point, the neuron does not only closes to the acceptance of input, it also cannot fire spike again. Depending on the exact formalisation of the model, the notion of a successful computation is defined together with its output [16].

4. SN P RULES' APPLICATION

Ordinarily, a spike is either moved, created or consumed, within or outside a neuron, but could never be modified (because there is only one type of objects in the system). Hence, SN P system's working is sequential based on the fact that out of the available rules that exist; only one at most is used at a time. So, it evolves according to a set of *spiking rules* and *forgetting rules* each of which is associated with a neuron that uses the rules for sending or internally consuming the spikes.

4.1 The Spiking Rule

The rules for spiking should take into account all spikes present in a neuron not only part of them, although not all spikes are consumed in this way.

So, going by the architecture and construct of SN P system given in section 3 above, objects are evolved by means of *spiking rules*, which are of the form: $E/a^c \rightarrow a; d$, where *E* is a regular expression over {a} and *c*, *d* are natural numbers, $c \geq 1$, $d \geq 0$. The meaning is that a neuron containing *k* spikes such that $a^k \epsilon L(E)$, $k \geq c$, can consume *c* spikes and produce one spike, after a delay of *d* steps. The produced spike is sent (maybe with a delay of some steps) to all neurons to which a synapse exists outgoing from the neuron where the rule was applied. This implies that there is replication of the spike which are distributed to all the neurons in the interconnection. However, a neuron called *output neuron* sends its own spike to the environment. Hence, this leads to the generation of *spike train* which is formed from the binary numbers 1s and 0s of the released spikes or otherwise.

4.2 The Forgetting Rule

This removes all the spikes from the neurons and is of the form $a^s \rightarrow \lambda$ with the meaning that $s \geq 1$ spikes are removed, provided that the neuron contains exactly *s* spikes. We say that the rules "cover" the neuron, all spikes are taken into consideration when using a rule.

The application of the rules depends on the contents of the neuron. This implies that the applicability of a rule is established based on the total number of spikes contained in the neuron. If no firing rule can be applied in a neuron, there may be the possibility to apply a forgetting rule, which removes from the neuron a predefined number of spikes.

More importantly however, it should be observed that the applicability of a rule is established based on the total number of spikes contained in the neuron.

5. IMPLEMENTING SN P SYSTEM

The Figure 4 below, [17] introduces an example of the standard way of representing an SN P system. The output neuron, denoted by σ_7 has an arrow pointing to the environment. At the first instance, only neurons σ_1, σ_2, σ_3 and σ_7 contain spikes, hence they fire immediately thereby releasing spikes. In particular, the output neuron spikes, so, a spike is sent to the environment. It is pertinent to note that in the first step we cannot use the forgetting rule $a \rightarrow \lambda$ in $\sigma_1, \sigma_2, \sigma_3$ because we have more than one spike present in each neuron.

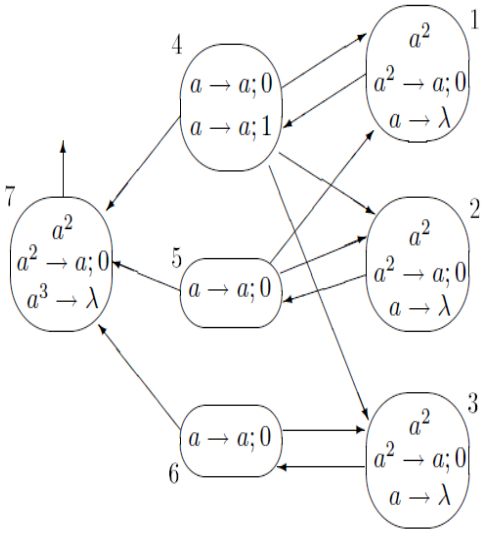

Figure 4: Representing SN P system [17]

The spikes of neurons $\sigma_1, \sigma_2, \sigma_3$ will be passed to neurons $\sigma_4, \sigma_5, \sigma_6$. In step 2, $\sigma_1, \sigma_2, \sigma_3$ contain no spike inside, hence will not fire, but $\sigma_4, \sigma_5, \sigma_6$ fire. Neurons σ_5, σ_6 have only one rule, but neuron σ_4 behaves non-deterministically, choosing between the rules $a \to a; 0$ and $a \to a; 1$. Assume that for $m \geq 0$ steps we use here the first rule. This means that three spikes are sent to neuron σ_7, while each of neurons $\sigma_1, \sigma_2, \sigma_3$ receives two spikes. In step 3, neurons $\sigma_4, \sigma_5, \sigma_6$ cannot fire, but all $\sigma_1, \sigma_2, \sigma_3$ fire again. After receiving the three spikes, neuron σ_7 uses its forgetting rule and gets empty again. These steps can be repeated arbitrarily many times.

In order to have neuron σ_7 firing again, we have to use sometimes the rule $a \to a; 1$ of neuron σ_4. Assume that this happens in step t (it is easy to see that $t = 2m + 2$). This means that at step t only neurons σ_5, σ_6 emit their spikes. Each of neurons $\sigma_1, \sigma_2, \sigma_3$ receives only one spike - and forgets it in the next step, $t + 1$. Neuron σ_7 receives two spikes, and fires again, thus sending the second spike to the environment. This happens in moment $t+1 = 2m+2+1$, hence between the first and the second spike sent outside have elapsed $2m + 2$ steps, for some $m \geq 0$. The spike of neuron (the one "prepared-but-not-yet-emitted" by using the rule $a \to a; 1$ in step t) will reach neurons $\sigma_1, \sigma_2, \sigma_3$ and σ_7 in step $t+1$, hence it can be used only in step $t+2$; in step $t+2$ neurons $\sigma_1, \sigma_2, \sigma_3$ forget their spikes and the computation halts. The spike from neuron σ_7 remains unused, there is no rule for it. Note the effect of the forgetting rules $a \to \lambda$, from neurons $\sigma_1, \sigma_2, \sigma_3$: without such rules, the spikes of neurons σ_5, σ_6 from step t will wait unused in neurons $\sigma_1, \sigma_2, \sigma_3$ and, when the spike of neuron σ_4 will arrive, we will have two spikes, hence the rules $a^2 \to a; 0$ from neurons $\sigma_1, \sigma_2, \sigma_3$ would be enabled again and the system will continue to work.

6 THE FUTURE OF SN P SYSTEM IN IDS

Spiking neural P systems are a versatile formal model of computation that can be used for designing efficient parallel algorithms for solving known computer science problems. Going through the literature, it would be observed that SN P system has enjoyed little or no application in the area of cyber-security at large and intrusion detection in particular. However, it has extensively been applied in solving Boolean satisfiability (SAT) problem, Traveling Salesman Problem (TSP) and some other n-p hard problems. Infact, recently it was applied to solve the problem of faulty section estimation in Power system [18]. From the viewpoint of real-world applications, [19] opined that SN P system is highly attractive based on the following reasons:

- parallel computing advantage,
- high understandability (due to their directed graph structure),
- dynamic feature (neurons firing and spiking mechanisms make them suitable to model dynamic behaviors of a system),
- synchronization (that makes them suitable to describe concurrent events or activities),
- non-linearly (which makes SN P systems suitable to process non-linear situation).

This is.because if a bound is imposed on the number of spikes present in any neuron during a computation, then a characterization of semilinear set of numbers is obtained.

More importantly however, within any parallel environment, neuron supports three kinds of parallel processing. These include:

a.) Multiple simulations distributed over multiple processors. This implies that each processor executes its own simulation.

b.) Distributed network models with gap junctions.

c.) Distributed models of individual cells (each processor handles part of the cell). Setting up distributed models of individual cells may somehow require considerable effort.

Specifically therefore, it may be said that the parallel computing advantage imbedded in SN P is a goldmine which would have significant application in the modeling of IDSs..

7. CONCLUSION

So far, this presentation had dwelt on the preview of Intrusion Detection System in general and Spiking Neural P system in particular. Their converging point was established. Also, SN P rules and formalism were explained. More importantly however, The computational capabilities of neurons were also not left out. Finally, areas through which SN P system could be used in IDS was pin-pointed.

In order to effectively address the various intrusion threats, it is not out of place to call for a shift of focus to IDS. Going by the literature reveiew, SN P system has not been applied to IDS. So, we hereby seize this opportunity to advocate for more interest and concerted efforts in this regard because of its potetialities.

REFERENCES

[1] R. Shanmugavadivu & N. Nagarajan, "An Anomaly-Based Network Intrusion Detection System Using Fuzzy Logic", *(IJCSIS) International Journal of Computer Science and Information Security*, Vol. 8, No. 8, 2010.

[2] P. Garcı´a-Teodoroa, et al, "Anomaly-based Network Intrusion Detection", *Techniques, systems and challenges; computers & security*, pp. 18 – 28, 2009.

[3] Lee, W. & S.J. Stolfo, "Data Mining Approaches for Intrusion Detection". *Proceedings of the Seventh USENIX Security Symposium*, pp. 79-93, 1998

[4] J. Hai et al, "A Fuzzy Data Mining Based Intrusion Detection Model", *Distributed Computing Systems*, 10th IEEE International Workshop on Future Trends pp. 191 – 197, 2004.

[5] R.A. Arafat, "A new model for monitoring Intrusion based on Petri Nets" in *Information Management & Computer Security*, pp. 175-182, 2004.

[6] B. Shanmugam &, N. B. Idris, "Improved Intrusion Detection System Using Fuzzy Logic for Detecting Anamoly and Misuse Type of Attacks", in *Proceedings of the International Conference of Soft Computing and Pattern Recognition*, pp. 212-217, 2009.

[7] J. Yu, et al, "TRINETR: An Architecture for Collaborative intrusion detection and knowledgebased alert evaluation", *Advanced Engineering Informatics* pp. 93–101, 2005.

[8] A. K. Ghosh, & A. Schwartzbard, "A Study in Using Neural Networks for Anomaly and Misuse Detection", *Proceedings of the Eighth USENIX Security Symposium*, pp. 141-151, 1999.

[9] M. Amini & R. Jalili, "Network-Based Intrusion Detection Using Unsupervised Adaptive Resonance Theory (ART)", *Advances in Neural Information Processing Systems* 10, Cambridge, MA: MIT Press, 1998.

[10] S. A. Taghanaki et al. "Synthetic Feature Transformation with RBF Neural Network to Improve the Intrusion Detection System Accuracy and Decrease Computational Costs" *International Journal of Information & Network Security (IJINS)* Vol.1, No.1, pp. 28-36, April 2012

[11] A. Păun & Gh. Păun, "Small Universal Spiking Neural P Systems", *Journal of Biosystems*, Elsevier, Vol.90, pp.48-60, 2007.

[12] M. C.Marcos & B. L. Milenova, "Creation and Deployment of Data Mining-Based Intrusion Detection Systems in Oracle Database 10g", in *Proceedings of the Fourth International Conference on Machine Learning and Applications*, 2005

[13] H. Debar, et al, "Towards a Taxonomy of Intrusion-Detection Systems*" Computer Networks*, 31 (8), pp. 805–822, 1999.

[14] S.X. Wu & W. Banzhaf, "The Use of Computational Intelligence in Intrusion Detection Systems: a Review", *Applied Soft Computing Journal* 10, pp. 1–35, 2010.

[15] Y. Yuan & D. Guanzhong, "An Intrusion Detection Expert System with Fact-Base" *Asian Journal of Information Technology*, 6: pp. 614-617, 2007.

[16] P. M. Venkata, et al., "Protocol Modeling in Spiking Neural P systems and Petri nets" *International Journal of Computer Applications* (0975 - 8887) Volume 1 – No. 24 , 2010

[17] M. Ionescu et al, "Spiking Neural P systems". *Fundamenta Informaticae* 71(2–3): pp. 279–308, 2006.

[18] H. Peng et al, "Fuzzy Reasoning Spiking Neural P System for Fault Diagnosis" in *Information Sciences* 235, pp. 106-116, 2013.

[19] J. Wang & H. Peng, "Adaptive Fuzzy Spiking Neural P systems For Fuzzy Inference and Learning". In *Proc. Eight Brainstorming Week on Membrane Computing*, Sevilla, Spain, pp. 235-248, 2010.