An Architecture for Self-healing in Internet of Things

Fernando Mendonça de Almeida, Admilson de Ribamar Lima Ribeiro, Edward David Moreno
Department of Computing
Federal University of Sergipe – UFS
São Cristóvão, Brazil
e-mails: fernando.m.al.91@gmail.com, admilson@ufs.br, edwdavid@gmail.com

Abstract—**Security in Internet of Things has an important role into mass adoption of the technology. There are some research works in this area, but most of them are related to Wireless Sensor Networks or Internet Networks. The association of a security mechanism and the self-* properties is very beneficial for the Internet of Things, considering the growth of connected devices. This paper proposes an architecture that uses the Dendritic Cell Algorithm for a security system with self-healing property for the Internet of Things.**

Keywords-Internet of Things; Dendritic Cells Algorithm; Self-healing.

I. Introduction

The Internet of Things (IoT) is a novel paradigm whose concept is based on ubiquitous presence of objects - sensors, actuators, Radio-Frequency IDentification (RFID) tags, mobile devices etc - that interact with each other using unique addresses to achieve common goals [1]. The IoT is extremely vulnerable to attacks, most of communication is wireless based, most of the components have constrained resources and it is possible to physically attack the IoT components [1][2]. Considering that the IoT will have information about almost everything, security and privacy are key concerns in IoT research [3][4].

Xu, He and Li [3] say that the research about security in IoT is necessary for the massive adoption of this technology in Industry. Gubbi, Buyya, Marusic and Palaniswami [4] highlight the need of self-protection in domestic applications, arguing that actuators will be connected to the system and they will need protection from intruders.

According to Roman, Zhou and Lopez [5], fault tolerance will be essential in the IoT. The number of vulnerable systems and attacks will increase, so there is a need to develop intrusion detection and prevention systems to protect the components of the IoT.

The growth of connected devices in IoT make the human intervention less effective. Autonomic computing aims to reduce the human intervention in complex systems, like the human nervous system controls autonomically some functions of body, like the digestive system [6].

The proposed architecture defines five components, distributed between the monitoring phase, analysis phase and knowledge component, to add the property of self-healing into the Internet of Things. The Dendritic Cell Algorithm (DCA) is combined with an Artificial Intelligence component to allow the DCA learning the role of each information sensed. The architecture described in this paper can implement a self-healing system in IoT nodes distributing roles between them in order to ease design of self-healing systems for IoT.

The remainder of this paper has five more sections: Section II introduces some aspects of this subject and the self-* properties; Section III presents four kinds of attacks in the Internet of Things that will be mitigated with the proposed architecture; Section IV presents three works related to security in IoT, self-protected system and use of dendritic cell algorithm; Section V describes the architecture to mitigate four kinds of attacks in IoT and Section VI presents the conclusion.

II. Autonomic Computing

The first utilization of Autonomic Computing term was made by the International Business Machine (IBM) in 2001 to describe self-managing systems [10]. The Autonomic term comes from biology, where, for example, the autonomic nervous system from the human body takes care of most bodily functions, by removing from the consciousness the need to coordinate all the bodily functions.

In IBM's manifesto, they suggested that complex systems should have autonomic properties and distilled the four properties of self-managing systems: self-configuration, self-optimization, self-healing and self-protecting.

A. Self-configuration

The self-configuration property is found in systems that are capable to self-install and self-set to achieve the user goals.

B. Self-optimization

The self-optimization property is found in systems that can make some changes proactively to improve the performance of the system.

C. Self-healing

The self-healing property is found in systems that detect and diagnose problems. It is important that the self-healing systems have fault tolerance.

D. Self-protecting

The self-protecting property is found in systems that protect themselves from malicious attacks. The autonomic system adjusts itself to offer security, privacy and data protection.

E. MAPE-K Autonomic Loop

In the IBM's manifesto, they presented a reference model, the MAPE-K autonomic control loop, where the responsibilities are shared between the components of the MAPE-K loop: Monitor, Analyze, Plan, Execute and Knowledge, Sensors and Effectors.

The monitor phase is responsible to collect data from sensors and process that data through the analysis phase. The analysis phase is responsible to receive the processed data and detect possible problems in the system. The plan phase organizes the necessary actions to fix the detected problems in analysis phase. The execution phase implements the actions planned.

III. Dendritic Cell Algorithm

The Dendritic Cell Algorithm (DCA) was introduced by Greensmith, Aickelin and Cayzer [11] and is inspired by the Danger Theory of mammalian immune system. The main elements of DCA are: Dendritic Cells (DC), Lymph nodes and antigens. The input signals of DCs are: danger signal, safe signal, PAMP (pathogenic associated molecular patterns) and inflammatory signal. The output signals of DCs are: Costimulatory Molecules (CSM), semi-mature signal and mature signal.

The antigens are the input of DC and they are presented iteratively to dendritic cells. Each antigen increments the CSM. When the CSM pass the migration threshold, the DC migrate to lymph node. The danger signal and PAMP increments the mature signal of DC and the safe signal increments the semi-mature signal. The inflammatory signal raises all other signal increments.

When the DC achieves the migration threshold, it will move to lymph node and the DC will be labeled as mature or semi-mature, comparing the mature and semi-mature signals. After receiving a defined number of DCs, the lymph node will calculate the Mature Context Antigen Value (MCAV), that is the percentage of mature DCs per all DCs received. The Dendritic Cell Algorithm detects an attack if the MCAV surpass a defined threshold.

IV. Security Threats in Internet of Things

Ashraf and Habaebi [2] proposed a taxonomy for security threat mitigation techniques. In this taxonomy, there are fifteen threats classified between actors (Managed Resources and Autonomic Managers), layer (Machine to Machine, Network and Cloud) and approach (Self-Protecting, Self-Healing and Hybrid). This paper discusses four security threats: Jamming, Sinkhole, Hello Flood and Flooding.

A. Jamming

The Jamming threat affects the managed resource and is classified in the Machine to Machine (M2M) layer. It is an attack that occupies the wireless spectrum blocking the communication between the IoT devices. The attacker uses noise signals to interference the wireless communication. To detect a jamming attack, the device monitors the Received Signal Strength Indicator (RSSI) values. It signal is abnormally high when a jamming attack occurs. That technique was used by Salmon et al. [9] to detect jamming attack in a Wireless Sensor Network.

B. Sinkhole

A sinkhole attacker announces a beneficial routing path to receive route traffic through it. It is classified in the Network layer and affects the managed resources and autonomic manager. The Intrusion Detection System proposed by Raza, Wallgren and Voigt [7] uses the representation of the network to find inconsistence and detect a sinkhole attack.

C. Hello Flood

The Hello Flood attack can occur when the routing protocol prompts a node to send hello messages to announce its presence to the neighbors. The Routing Protocol for Low power and lossy networks (RPL) needs to build the routing paths with some kind of hello messages, which makes the RPL vulnerable to Hello Flood attack. The Hello Flood attack is classified in the Network layer and affects both managed resources and autonomic managers.

D. Flooding

In a flooding attack, the attacker tries to run out the victim's resources, e.g. battery, sending many connection establishment requests. The Flooding attack is classified in the Cloud layer and affects the Autonomic Manager. Considering that most part of IoT communication with IP protocol uses UDP, this attack can be mitigated by setting traditional connection barriers [2].

V. Related Work

A. SVELTE

Raza, Wallgren and Voigt [7] designed, implemented and evaluated SVELTE, an Intrusion Detection System (IDS) for the Internet of Things. The SVELTE detects sinkhole and selective-forwarding attacks in IPV6 over Low power Wireless Personal Area Networks (6LoWPAN) wireless network that uses RPL routing protocol.

Their IDS has a hybrid approach, it has distributed and centralized modules. The three main modules are: 6Mapper (6LoWPAN Mapper), Intrusion Detection Component and a mini-firewall.

The 6Mapper builds the network topology of RPL in the border router. Each node needs to have an 6Mapper client. The Intrusion Detection Component uses four algorithms to detect sinkhole and selective-forwarding attacks, the first algorithm detects inconsistence in network topology, the second algorithm detects nodes that may have messages filtered, the third algorithm verifies the network topology validity and the last algorithm verifies the end-to-end losses. The mini-firewall module has a server, in border router, and a client, in each node. Each node, when necessary,aks the border router to block messages from an external attacker.

The authors concluded that SVELTE can be used in the context of RPL, 6LoWPAN and IoT. Detecting sinkhole attacks with nearly 90% true positive rate in a small lossy network and almost 100% true positive rate in a lossless network configuration.

B. *Dai, Hinchey, Qi and Zou*

Dai, Hinchey, Qi and Zou [8] proposed a self-protected system based on feature recognition using virtual neurons. The virtual neurons have three components: information collector, neighbor communicator and feature recognizer. The information collector senses useful data for the self-protecting mechanisms, like processing use, memory, processing status, message size, transmission direction etc.

The virtual neurons communicate with each other in Peer to Peer (P2P) model and hierarchical model. The hierarchical communication allows a fast message propagation between clusters, each cluster having a head-neuron that have a fast communication with each other head-neurons.

The authors propose five self-protecting mechanisms, each one with an associate algorithm to detect and prevent one type of attack. The attack types are: eavesdropping, replay, masquerading, spoofing and denial of service.

Each mechanism senses the environment's data and, if an attack is detected, the connection is finished and all nodes involved are alerted.

In the paper, they present three use cases and the results. The use cases test the self-protection with eavesdropping, replay and denial of service attack. In the eavesdropping use case, with 15% of node coverage and with one node per three seconds of monitor frequency, it is possible to effectually prevent the attack. In the replay use case, it is necessary to use a buffer and the prevention grows when the buffer size and frequency inspection grows too. In denial of service use case, the authors verify that the proposed mechanisms increase the server availability.

C. *Salmon et al.*

Salmon et al. [9] proposed an anomaly based IDS for Wireless Sensor Networks using the Dendritic Cell Algorithm. The proposed IDS architecture has five elements: Monitoring, responsible for sensing the environment's values, Context Manager, responsible for managing the monitoring and parameter base, Intrusion Detection Manager, responsible for organizing the tasks and coordinate the responses and actions to other managers, Decision Manager, responsible for executing the dendritic cell algorithm, detect an attacker and manage the rules base, and Countermeasures, responsible for executing the actions to combat the identified attacks.

In their proposal, the authors divide two roles to be represented by the nodes: Dendritic Cells (sensor-dc) and Lymph node (sensor-lymph). The sensor-lymph have the Decision Manager and Countermeasures components, while sensor-dc have all the other components.

In the experiment, Salmon et al. used MICAz mote [16] with TinyOS [17]. The scenarios were simulated with TOSSIM (TinyOS Simulator) and they try to identify jamming attacks. The environment data used by dendritic cell algorithm included RSSI level, representing the PAMP signal, the received messages rate, as danger signal, and the inverse of received messages rate, as safe signal.

Several experiments were done, including changing configuration, time of attack, number of sensor-dc. Through the tests, the authors concluded that the IDS proposed is efficient for Wireless Sensor Networks saving energy from the nodes while there is a jamming attacker.

D. *Comparison of Related work*

The related work listed in this paper has some common goals, but not all of them. The SVELTE IDS [7] is designed for Internet of Things, but it was not designed considering the autonomic properties. Similar to SVELTE IDS, Salmon et al. [9] work was designed with resource constrains, but did not have the same network topology. Dai, Hinchey, Qi and Zou [8] designed their system with autonomic properties, but it is not possible to know if their approach fits into Internet of Things constrains. There is a summary of features of each work in Table I.

VI. SELF-HEALING ARCHITECTURE

The proposed architecture is based on MAPE-K loop. The phases of MAPE-K loop are divided into components and distributed on the nodes. The architecture is based on RPL Destination-Oriented Directed Acyclic Graph (DODAG), that is the default topology of a 6LoWPAN network that uses the RPL protocol. A typical RPL DODAG is depicted in Figure 1, it has a root and the other nodes. All nodes have a node ID, i.e., an IPv6 address, and all nodes, except the root, have one or more parents and a rank, that is relative to the distance between the node and the root.

TABLE I. RELATED WORK FEATURES COMPARISON

Feature	**Related Work**		
	SVELTE [7]	***Dai, Hinchey, Qi and Zou [8]***	***Salmon et al. [9]***
Has Autonomic Property		x	
Designed for IoT	x		
Detect Jamming			x
Detect sinkhole	x		
Detect DoS		x	
Detect Selective-Forwarding	x		
Detect Eavesdropping, Replay, Masquerading and Spoofing		x	

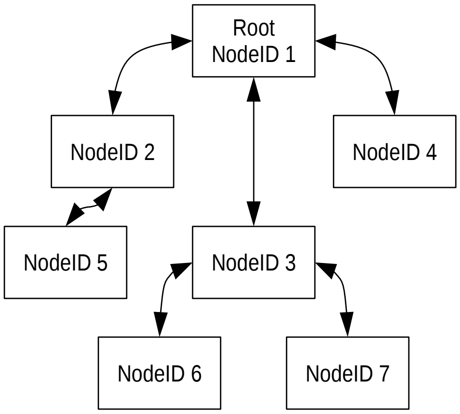

Figure 1. A typical RPL DODAG with one root and six other nodes.

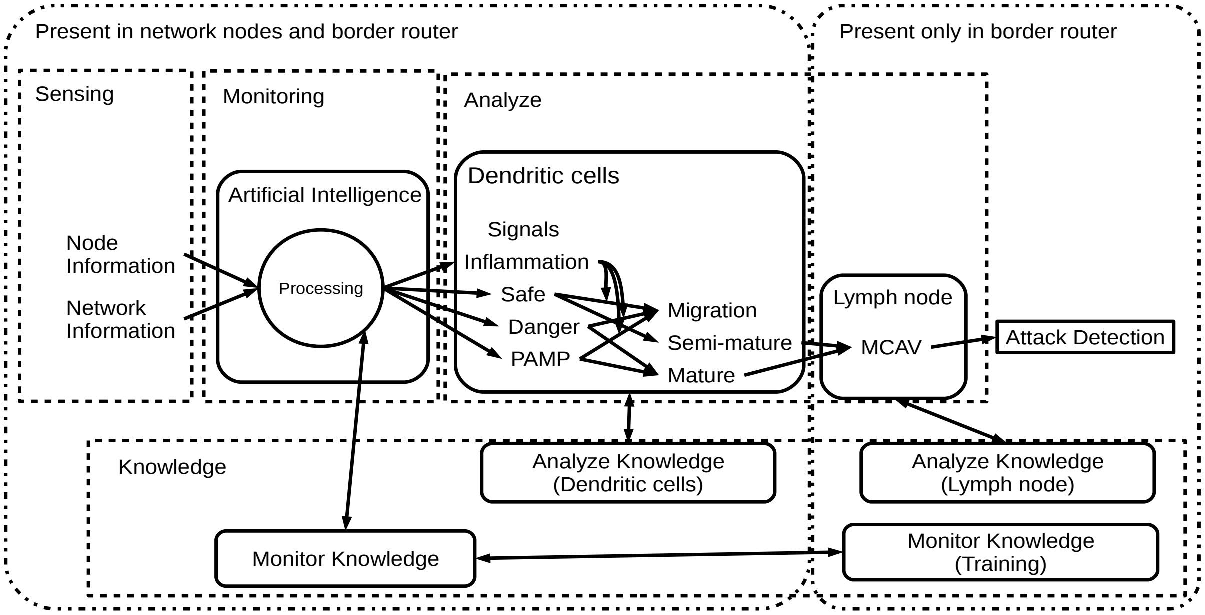

Figure 2. The proposed Architecture

There are five components described in the proposed architecture: Artificial Intelligence, Dendritic Cell, Lymph node, Monitor Knowledge and Analyze Knowledge. The other components of MAPE-K loop, Planning and Execution Phases, are not defined yet in the proposed architecture, they will be analyzed in future works.

The current components of the proposed architecture are distributed between Sensing, Monitoring, Analyzing and Knowledge elements of MAPE-K loop. The distribution of the proposed architecture components in the MAPE-K loop elements is depicted in Figure 2. Some components are not present on all network nodes, the Lymph node, Analyze Knowledge and part of Monitor Knowledge are present only in border router. Sensing, Monitoring, Analyzing, Monitor Knowledge and Analyze Knowledge components can be present in any network node and are present in border router.

A. Sensing Phase

The sensing phase is present in all nodes. In this phase the network and node information will be sent to monitoring phase. Node information will be the rate of successfully sent packets, total sent packets, RSSI level and more. Network information will be the network packets information, the DODAG routing tree information and more. As the root node is connected to the Internet, it will get more information about the network.

B. Monitoring Phase

The monitoring phase also is present in all nodes of the local network. In this phase there is the Artificial Intelligence component responsible to get the information from the sensing phase, the node and network information, process it and generate useful information for the analyzing phase.

The Artificial Intelligence proposed in this paper is an Artificial Neural Network Multi-Layer Perceptron (MLP). The MLP will be used to get useful information from the sensing phase, e.g. the network packets processed have some information such as total time to process and respond, and the MLP can try to predict this information without processing the packet, in order to provide this information quickly to the Analyzing phase.

The MLP will give a mix of real and inferred information, to the Analyzing phase in order to detect a possible attack to the node and network.

C. Analyzing Phase

In analyzing phase, the information generated by the monitoring phase will be used as input to the dendritic cells DC component. The DC component is present in all nodes of the local network. The signals of dendritic cells are classified as safe, danger or inflammatory signal. The monitoring phase receives the information from the sensing phase and infers the signal levels to this phase.

When the DC have enough information, they will migrate their result to the lymph node, present only in border router. The lymph node processes the DC result and detects if there is an attack on the network. This attack detection information in passed to the Planning phase.

D. Knowledge

The Knowledge components described in this paper are the Analyzing and Monitor Knowledge. The Plan and Execute Knowledge components will be present in the architecture too.

The monitor knowledge is split in two components, the training component and the component itself. The monitor knowledge itself is present in all nodes of the local network while the training monitor knowledge is present only in border router. This split occurs because the training of the artificial intelligence may need more resources than the node can offer.

The analyze knowledge is split in two parts, the dendritic cell part and the lymph node part. Each one is present where it counterpart component in analyze phase is present.

E. Planning, Execution and Effection Phase

Planning, execution and effection phase are not described in this architecture yet. Most efforts to define the network are concentrated on define how the nodes of an IoT network will detect an attack.

The planning phase will receive the analyze phase warning about an attack in the network and plan how to mitigate the side effects of the attack. This phase should consult previous network packages to improve the plan. The planning phase will be split in two components, one present in border router, the node with more processing resources in the local network and the other present in all nodes of the local network. The planning phase in the border router will list actions to mitigate the side effects of the attack and will distribute these actions to the planning component inside all the nodes. When the planning component receives the actions, it will pass the actions to the Execution phase.

The Execution phase will receive the actions planned in the planning phase and deliver each order from each action to the effection phase. The effectors of the node will receive clear orders from the execution phase and they will perform it to mitigate the side effects of the attack detected in the planning phase.

F. Attack Detection

The early implementation of the proposed architecture will try to detect Jamming, Sinkhole, Hello Flood and Flooding attacks. To detect Jamming attacks, as in SALMON et al. [9], the RSSI level and rate of messages will be sensed in the sensing phase. To detect sinkhole attack, like SVELTE [7], the DODAG tree information will be processed and inconsistency, validity etc will be sensed in the sensing phase. To detect hello flood and flooding, the number of connection attempts and RPL's hello messages from the same address will be sensed. The proposed architecture will capture the mentioned information and use it to detect an attack.

VII. RESULTS

The initial results of this research are about the technique used in the monitoring phase. The chosen technique for the first efforts is an Artificial Neural Network, a Multi-Layer Perceptron with Limited Weights (MLPLW) based on the neural network with limited precision weights [12].

The MLPLW implemented has 10 neurons in the hidden layer and each weight is represented by a byte. The training technique of the MLPLW is the Quantized Back-Propagation Step-by-Step (QBPSS) [12], a modified version of Back-Propagation for neural network with limited weights.

To check the implementation, the KDD99 dataset [13], widely used in Intrusion Detection Systems, was used. Yan, Wang and Liu [14] used the KDD99 dataset to evaluate their hybrid technique that uses a rule-based decision and neural network. Their accuracy rate was 99.75% and the false positive rate was 0.57%.

The KDD99 dataset was used with our ANNLW implementation with a stream based training [15]. Each input is used once and the accuracy and false positive rates were measured every thousand inputs. After the first thousand inputs, the MLPLW achieved 97,65% accuracy, but oscillated until the thirty-fourth thousand input. The accuracy rate after the stabilization is close to the accuracy in Yan, Wang and Liu[13], considering the use of fewer neurons in the hidden layer. The oscillation of the accuracy rate of the MLPLW is depicted in Figure 3.

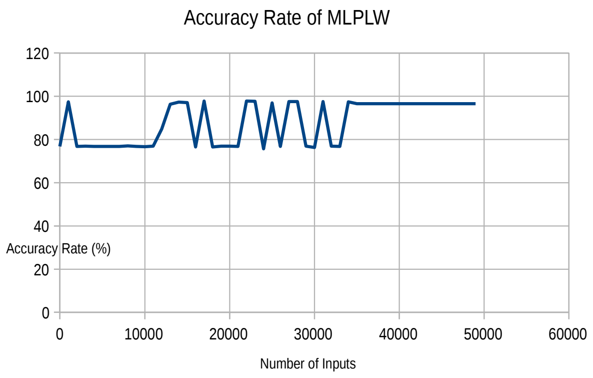

Figure 3. Accuracy rate of MLPLW over the number of inputs.

As our proposed architecture shall be used in the Internet of Things context, the used techniques should be in accordance to the use of resources. The memory used by MLPLW can be seen in Table II. It is possible to see in Table II that the MLPLW have a small impact in memory utilization in an ARM Cortex-M3 with 512 KB of Persistent memory and 64 KB of Volatile memory, 0.33% and 0.64% respectively.

VIII. CONCLUSION AND FUTURE WORK

The security in Internet of Things is a key property for the mass adoption of the technology. The research of security in Wireless Sensor Network and Internet itself can be used to show paths into security in IoT.

This work presents an architecture with self-healing property for the Internet of Things using ideas from Wireless Sensor Networks applied to a 6LoWPAN network.

The system will reuse components to detect multiple types of attacks, only by using additional information from the nodes and network, but without a dramatic algorithm change.

The proposed architecture will mitigate four different kinds of attacks of three different layers: Machine to Machine, Network and Cloud.

The MLPLW implemented shows that the monitor phase of the proposed architecture can use less than 1% of the memory of the embedded system and still have a high accuracy rate.

TABLE II. MLPLW MEMORY CONSUMPTION TABLE

Resource	***MLPLW consumption***
ROM memory	1716 bytes
RAM memory	420 bytes
Related ROM (related to 512 KB)	0.33%
Related RAM (related to 64 KB)	0.64%

For future work, there will be the implementation of the proposed architecture to validate it. The Artificial Intelligence component algorithm should be defined. The performance of the system should be evaluated to verify if the proposed architecture implementation has better results than related work.

ACKNOWLEDGMENT

This work was supported by CAPES and FAPITEC/SE

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey”. Computer Networks, vol. 54, no. 15, 2010, pp. 2787-2805.

[2] Q. M. Ashraf and M. H. Habaebi, “Autonomic schemes for threat mitigation in Internet of Things.” Journal of Network and Computer Applications, vol. 49, 2015, pp. 112-127.

[3] L. D. Xu, W. He, and S. Li, “Internet of things in industries: A survey.” Industrial Informatics, IEEE Transactions on, vol. 10, no. 4, 2014, pp. 2233-2243.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions.” Future Generation Computer Systems, vol. 29, no. 7, 2013, pp. 1645-1660.

[5] R. Roman, J. Zhou, and J. Lopez, “On the features and challenges of security and privacy in distributed internet of things.” Computer Networks, vol. 57, no. 10, 2013, pp. 2266-2279.

[6] M. C. Huebscher and J. A. McCann, “A survey of autonomic computing—degrees, models, and applications.” ACM Computing Surveys (CSUR), vol. 40, no. 3, 2008, pp. 7.

[7] S. Raza, L. Wallgren, and T. Voigt, “SVELTE: Real-time intrusion detection in the Internet of Things.” Ad hoc networks, vol. 11, no. 8, 2013, pp. 2661-2674.

[8] Y. S. Dai, M. Hinchey, M. Qi, and X. Zou, “Autonomic security and self-protection based on feature-recognition with virtual neurons.” Dependable, Autonomic and Secure Computing, 2nd IEEE International Symposium on. IEEE, 2006.

[9] H. M. Salmon, et al.. “Intrusion detection system for wireless sensor networks using danger theory immune-inspired techniques.” International journal of wireless information networks, vol. 20, no. 1, 2013, pp. 39-66.

[10] J. O. Kephart and D. M. Chess, “The vision of autonomic computing.” Computer, vol. 36, no. 1, 2003, pp. 41-50.

[11] J. Greensmith, U. Aickelin, and S. Cayzer, “Introducint dendritic cells as a novel immune-inspired algorithm for anomaly detection.” Artificial Immune Systems. Springer Berlin Heidelberg, 2005, pp. 153-167

[12] J. Bao, Y. Chen and J. Yu, “An optimized discrete neural network in embedded systems for road recognition.” Engineering Applications of Artificial Intelligence, vol. 25, no. 4, 2012, pp. 775-782.

[13] KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, June 2015.

[14] K. Q. Yan, S. C. Wang and C. W. Liu, “A hybrid intrusion detection system of cluster-based wireless sensor networks.” Proceedings of the International MultiConference of Engineers and Computer Scientists, 2009, pp. 18-20.

[15] K. Faceli, A. C. Lorena, J. Gama and A. C. P. L. F. de Carvalho, “Aprendizado em Fluxos Contínuos de Dados.” Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina, Grupo Gen-LTC, 2011, pp. 260-269.

[16] MEMSIC, “MICAz Datasheet”. Available on: http://www.memsic.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf, June 2015.

[17] P. Levis et al., “Tinyos: An operating system for sensor networks.” Ambient intelligence. Springer Berlin Heidelberg, 2005.,pp. 115-148.

ISBN: 978-1-61208-418-3