Journal of Applied Mathematics & Bioinformatics, vol.6, no.3, 2016, 45-64
ISSN: 1792-6602 (print), 1792-6939 (online)
Scienpress Ltd, 2016

Ladon: A Cyber-Threat Bio-Inspired Intelligence Management System

Konstantinos Demertzis[1] **and Lazaros Iliadis**[2]

Abstract

According to the Greek mythology, Ladon was the huge dragon with the 100 heads, which had the ability to stay continuously up, in order to guard the golden "Esperides" apples in the tree of life. Alike the ancient one, digital Ladon is an advanced information systems' security mechanism, which uses Artificial Intelligence to protect, control and offer early warning in cases of detour or misleading of the digital security measures. It is an effective cross-layer system of network supervision, which enriches the lower layers of the system (Transport, Network and Data). It amplifies in an intelligent manner the upper layers (Session, Presentation and Application) with capabilities of automated control. This is done to enhance the energetic security and the mechanisms of reaction of the general system, without special requirements in computational resources. This paper describes the development of Ladon which is an advanced, incredibly fast and low

[1] Democritus University of Thrace, Department of Forestry & Management of the Environment & Natural Resources, 193 Pandazidou st., 68200 N. Orestiada, Greece. E-mail: kdemertz@fmenr.duth.gr

[2] Democritus University of Thrace, Department of Forestry & Management of the Environment & Natural Resources, 193 Pandazidou st., 68200 N. Orestiada, Greece. E-mail: liliadis@fmenr.duth.gr

Article Info: *Received* : October 12, 2015. *Revised* : December 28, 2015.
Published online : December 20, 2016.

requirements' effective filter, which performs analysis of network flow. Ladon uses Online Sequential Extreme Learning Machine with Gaussian Radial Basis Function kernel in order to perform network traffic classification, malware traffic analysis and fast-flux botnets localization.

Mathematics Subject Classification: 94C99
Keywords: Network Traffic Classification; Malware Traffic Analysis; Fast-Flux Botnets; SSH Traffic; Online Sequential Learning; Extreme Learning Machine; Gaussian Radial Basis Function Kernel

1 Introduction

1.1 Network security vulnerabilities and threats

Vulnerability is a weak spot in a network that might be exploited by a security threat. Risks are the potential consequences and impacts of unaddressed vulnerabilities. The combined network attacks and the highly intelligent detour methods of the digital security mechanisms constitute a new framework of vulnerabilities and threats. This is done by creating extremely complicated localization conditions. For example, the functional complexity combined with the chaotic architecture of the botnets [1], constitute an extremely dangerous case of electronic crime very hard to trace, which is related to a number of illegal actions namely: money mule recruitment sites, phishing websites, illicit online pharmacies, extreme or illegal adult content sites, malicious browser exploit sites and web traps for distributing virus.

1.1.1 Traffic classification

The supervision and categorization of network traffic is a specialized solution and a valuable tool used not only for the effective confrontation of planning, management and supervision of the networks, but also for the trace of attacks and for the study of electronic crimes. Traffic Classification is an automated process which categorizes network traffic according to various parameters into a number of traffic classes. Each resulting traffic class can be treated differently in order to differentiate the service implied for the user. A proper understanding of the applications and protocols in the traffic class is essential for any network manager to implement appropriate security policies [2]. There are two basic approaches to classifying traffic:

a) Classification based on a Payload method in which the packets are classified based on the fields of the payload, such as Layer 2 (Mac address), Layer 3 (IP address) and Layer 4 ports (source or destination or both) and Protocols.
b) Classification based on a Statistical method that uses statistical analysis of the traffic behavior like inter-packet arrival, session time and so on.

A serious drawback of the applications used for the study and characterization of network traffic and especially for the cases of encrypted traffic, which imposes the reconstruction of messages and entities in higher levels, is the complexity of these applications and their requirements in terms of computational resources. These requirements increase exponentially in cases of big volume network traffic analysis which come from broadband high speed networks and especially in cases of forensic analysis. Advanced classification techniques which rely on Deep Packet Inspection (DPI) are much more reliable methods. Also another important weakness of the traditional mechanisms of network flow analysis, related to the determination and phasing of vulnerabilities is the fact that they create high percentages of false alarms, they do not have sophisticated forecasting methods of respective threats and in most of the cases they fail completely to spot zero-day vulnerabilities [2].

1.1.2 Malware traffic analysis

Malware is a kind of software used to disrupt computer operation, gather sensitive information, or gain access to private computer systems. It can appear in the form of code, scripts, active content, or any other. Recent malware developments have the ability to remain hidden during infection and operation. They prevent analysis and removal, using various techniques, namely: obscure filenames, modification of file attributes, or operation under the pretense of legitimate programs and services and obfuscation attacks. Also, the malware might attempt to subvert modern detection software, by hiding running processes, network connections and strings with malicious URLs or registry keys. Malware Traffic Analysis is the primary method of malware and attacks identification and discovery the malware command-and control (C&C) communications related to these attacks. Furthermore it is an important method to estimate the behavior of malware, the purpose of attacks and the damage caused by malware activity [3].

1.1.3 Fast-flux botnets

The most common malware types, aim in the recovery of communication with the Command & Control (C2) remote servers and its retention on a regular basis. This is done in order for the botmasters to gather or distribute information and upgrades towards the undermined devices (bots). More specifically, the communication is achieved by the use of custom distributed dynamic DNS services which are executed in high port numbers. This is done in order to avoid to be traced by security programs located in the gateway of networks. In this way fast-flux botnets are created, whose target is the mapping of a fully qualified domain name to hundreds of IP addresses. These IPs are interchanged too fast, with a combination of random IP addresses and a very small Time-To-Live (TTL) for each partial DNS Resource Record. In this way a domain name can change its corresponding IP address very often (e.g. every 3 minutes). Another usual

approach is the blind proxy redirection (BPR) technique. The BPR continuously redirects the applications received by frontend systems to backend servers, in order to spoil the traces and the data that designate an attack. In this way the complexity of the fast-flux botnets increases [1].

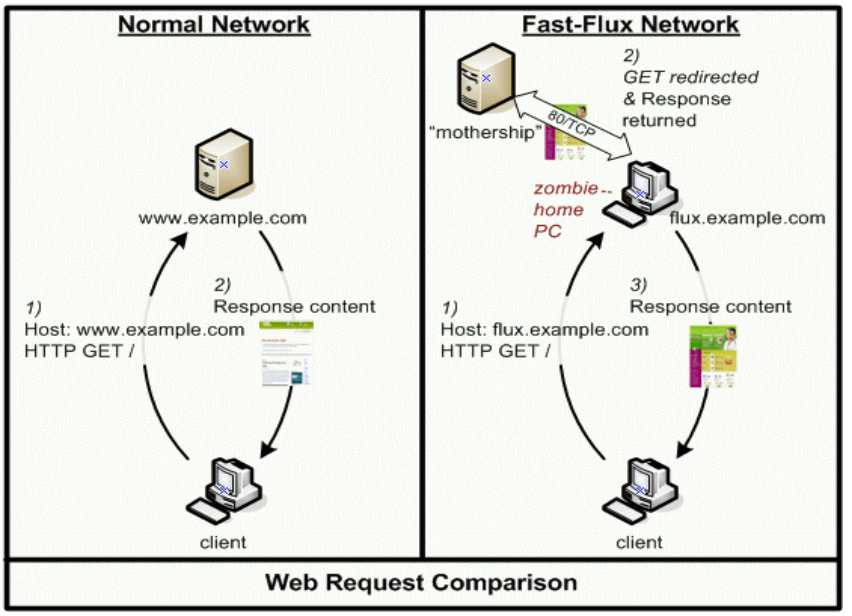

Figure 1: Normal Network vs Fast-Flux Network

The simplest type of fast flux, named single-flux, is characterized by multiple individual nodes within the network registering and de-registering their addresses as part of the DNS A (address) record list for a single DNS name. This combines round robin DNS with very short TTL values to create a constantly changing list of destination addresses for that single DNS name. The list can be hundreds or thousands of entries long. A more sophisticated type of fast flux, referred to itself as double-flux, is characterized by multiple nodes within the network registering and de-registering their addresses as part of the DNS Name Server record list for the DNS zone. This provides an additional layer of redundancy and survivability within the malware network [1].

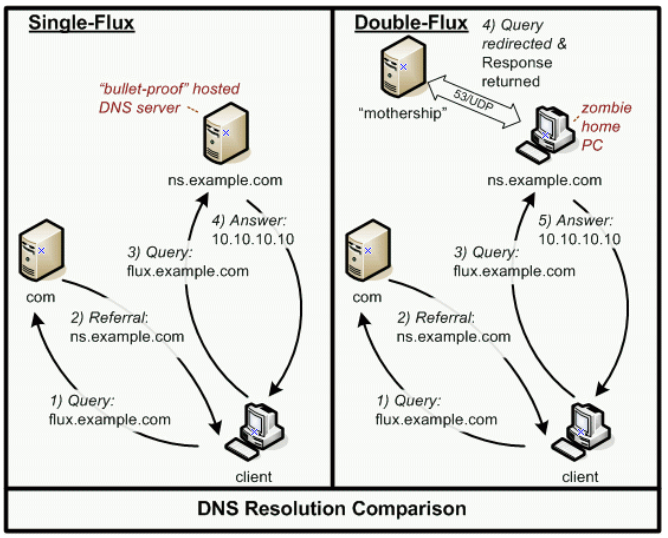

Figure 2: Single-Flux and Double-Flux Network

2 The Ladon system and literature

2.1 Ladon

This paper proposes the development of Ladon, a cyber-threat bio-inspired intelligence management system. Ladon is an innovative Network Traffic Classification, Malware Traffic Analysis and localization Fast-Flux Botnets. Unlike other techniques that have been proposed from time to time and focus in single traffic analysis approaches [4]-[6], Ladon is an efficient network supervision system which provides smart mechanisms for the supervision and categorization of networks. It provides intelligent approaches for the above task and tit is capable of defending over sophisticated attacks and of exploiting effectively the hardware capabilities with minimum computational and resources cost.

It is a biologically inspired artificial intelligence computer security technique [7]-[12] that use Online Sequential Extreme Learning Machine with Gaussian

Radial Basis Function kernel (OSEL_RBF) [13]-[15]. ELM is one of the most advanced artificial intelligence method which simlulates rationaly the function of the human brain by using the synaptic randomness and its dynamics.

A comparative study has been performed between the proposed approach and other evolving and bio-inspired learning methods. Based on the classification accuracy we have verified that the proposed approach has the optimal performance and reliability. More specifically this is achieved by using Radial Basis Function Neural Network (RBFNN), Group Methods of Data Handling (GMDH), Polynomial Artificial Neural Network (PANN) and Feed Forward Neural Networks (FFNN) trained under heuristic techniques such as Genetic Algorithm (FFNNGA), Particle Swarm Optimization (FFNNPSO), Ant Colony Optimization (FFNNACO) and Evolutionary Strategy (FFNNES) [16].

2.2 Literature review

It is a fact that a lot and significant work has been published in the literature, in applying machine learning (ML) techniques, to network traffic classification [17]-[19], malware traffic analysis [20]-[22] and fast-flux botnets [23]-[25] localization. In parallel, several other authors [26]-[28], have also summarized scientific effort of detecting the botnets while proposing novel taxonomies of detection methods, introducing different classes of botnet detection and presenting some of the most prominent methods within the defined classes. The authors have acknowledged the potential of machine learning-based approaches in providing efficient and effective detection, but they have not provided a deeper insight on specific methods, neither the comparison of the approaches by detection performances and evaluation practice.

On the other hand, Cheng et al. [29] proposed the use of ELM methods to classify binary and multi-class network traffic for intrusion detection with high accuracy. Hsu et al. [30] proposed a real-time system for detecting flux domains

based on anomalous delays in HTTP/HTTPS requests from a given client with very promising results. Also, Haffner et. al. [31] employed AdaBoost, Hidden Markov, Nave Bayesian and Maximum Entropy models to classify network traffic into different applications with vary high SSH detection rate and vary low false positive rate respectively, but they employed only few bytes of the payload. Furthermore Alshammari et al. [32] employed Repeated Incremental Pruning to Produce Error Reduction (RIPPER) and AdaBoost algorithms for classifying SSH traffic from offline log files without using any payload, IP addresses or port numbers. Finally, Holz et al. [33] proposed a passive method to locate fast-flux botnets by identifying potential fast-flux domain names in the URLs found in the body of spam emails (typically captured by spam traps and filters).

3 Literature methodologies employed by Ladon

3.1 Extreme Learning Machines

The Extreme Learning Machine (ELM) as an emerging biologically inspired learning technique provides efficient unified solutions to “generalized” Single-hidden Layer feed forward Networks (SLFNs) but the hidden layer (or called feature mapping) in ELM need not be tuned [14]. Such SLFNs include but are not limited to support vector machine, polynomial network, RBF networks, and the conventional feed forward neural networks. All the hidden node parameters are independent from the target functions or the training datasets and the output weights of ELMs may be determined in different ways (with or without iterations, with or without incremental implementations, etc.). ELM has several advantages, ease of use, faster learning speed, higher generalization performance, suitable for many nonlinear activation function and kernel functions.

According to the ELM theory [14], the ELM with Gaussian Radial Basis Function kernel (GRBFk) $K(u,v)=exp(-\gamma||u-v||^2)$ used in this approach. The hidden

neurons are k=20. Subsequently assigned random input weights w_i and biases b_i, i=1,…,N. To calculate the hidden layer output matrix H used the function (1):

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \cdots & h_L(x_1) \\ \vdots & & \vdots \\ h_1(x_N) & \cdots & h_L(x_N) \end{bmatrix} \quad (1)$$

$h(x) = [h_1(x), \ldots, h_L(x)]$ is the output (row) vector of the hidden layer with respect to the input x. $h(x)$ actually maps the data from the d-dimensional input space to the L-dimensional hidden-layer feature space (ELM feature space) H, and thus, $h(x)$ is indeed a feature mapping. ELM is to minimize the training error as well as the norm of the output weights:

$$\text{Minimize} : ||H\beta - T||^2 \text{ and } ||\beta|| \quad (2)$$

where H is the hidden-layer output matrix of the function (1). To minimize the norm of the output weights $||\beta||$ is actually to maximize the distance of the separating margins of the two different classes in the ELM feature space $2/||\beta||$.
To calculate the output weights β used the function (3):

$$\beta=(\frac{I}{C} + H^T H)^{-1} H^T T \quad (3)$$

where C is a positive constant is obtained and T resulting from the *Function Approximation of SLFNs with additive neurons* in which is an arbitrary distinct

samples with $t_i=[t_{i1}, t_{i2},\ldots,t_{im}]^T \in R^m$ and $T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}$. [14].

3.2 Online Sequential Extreme Learning Machines

The Online Sequential ELM (OS-ELM) [13] [15] is an alternative technique for large-scale computing and machine learning approaches that used when data becomes available in a sequential order to determine a mapping from data set

corresponding labels. The main difference between online learning and batch learning techniques is that in online learning the mapping is updated after the arrival of every new data point in a scale fashion, whereas batch techniques are used when one has access to the entire training data set at once. It is a versatile sequential learning algorithm because of the training observations are sequentially (one-by-one or chunk-by-chunk with varying or fixed chunk length) presented to the learning algorithm. At any time, only the newly arrived single or chunk of observations (instead of the entire past data) are seen and learned. A single or a chunk of training observations is discarded as soon as the learning procedure for that particular (single or chunk of) observation(s) is completed. The learning algorithm has no prior knowledge as to how many training observations will be presented. Unlike other sequential learning algorithms which have many control parameters to be tuned, OSEL_RBF only requires the number of hidden nodes to be specified [13], [15].

3.3 OSEL_RBF classification approach

The proposed method uses an OSEL_RBF classification approach in order to perform network traffic classification, malware traffic analysis and fast-flux botnets localization in an energetic security mode, that needs minimum computational resources and time. The OS-ELM consists of two main phases namely: Boosting Phase (BPh) and Sequential Learning Phase (SLPh). The BPh used to train the SLFNs using the primitive ELM method with some batch of training data in the initialization stage and these boosting training data will be discarded as soon as boosting phase is completed. The required batch of training data is very small, which can be equal to the number of hidden neurons [13]-[15]. The general classification process with OSEL_RBF classifier described below:

Phase 1 (BPh) [13], [15]

The process of BPh for a small initial training set $N=\{(x_i, t_i)|x_i \in R^n, t_i \in R^m, i=1, \cdots, \widetilde{N}\}$ described as follow:

a) Assign arbitrary input weight w_i and bias b_i or center μ_i and impact width σ_i, $i=1, \cdots \widetilde{N}$, where $\widetilde{N}$ number for hidden neuron or RBF kernel for a specific application.

b) Calculate the initial hidden layer output matrix $H_0 = [h_1, \cdots, h_{\widetilde{N}}]^{\mathrm{T}}$, where $h_i = [g(w_1 \cdot x_i + b_1), \cdots, g(w_{\widetilde{N}} \cdot x_i + b_{\widetilde{N}})]^{\mathrm{T}}$, $i = 1, \cdots, \widetilde{N}$, where g activation function or RBF kernel.

c) Estimate the initial output weight $\beta^{(0)} = M_0 H_0^T T_0$, where $M_0 = (H_0^T H_0)^{-1}$ and $T_0 = [t_1, \ldots, t_{\widetilde{N}}]^T$.

d) Set $k = 0$.

Phase 2 (SLPh) [13], [15]

In the SLPh the OS-ELM will then learn the train data one-by-one or chunk-by-chunk and all the training data will be discarded once the learning procedure on these data is completed. The essentials step of this phase for each further coming observation (x_i, t_1), where $x_i \in R^n$, $t_i \in R^m$ and $i = \widetilde{N} + 1, \widetilde{N} + 2, \widetilde{N} + 3$, described as follow:

a) Calculate the hidden layer output vector $h_{(k+1)} = [g(w_1 \cdot x_i + b_1), \cdots, g(w_{\widetilde{N}} \cdot x_i + b_{\widetilde{N}})]^{\mathrm{T}}$

b) Calculate latest output weight $\beta^{(k+1)}$ by the algorithm $\hat{\beta} = (H^T H)^{-1} H^T T$ which is called the Recursive Least-Squares (RLS) algorithm.

c) Set $k = k + 1$

4 The Ladon algorithm

The proposed Ladon algorithm includes the following ruleset which is the core of its reasoning.

1: Performs malware traffic analysis by OSEL_RBF with **MTAD** dataset
If the malware analysis gives a positive result (Malware) the network traffic blocked and the process terminated.
If the malware analysis gives a negative result (Benign), no action is required and go to step 2.

2: Performs network traffic analysis by OSEL_RBF with **NTCD** dataset
If the network traffic classification result is not a HTTP, no action is required.
If the network traffic classification result is a HTTP, go to step 3.

3: Performs botnet analysis by OSEL_RBF with **F2BLD** dataset
If the botnet classification result gives a positive result (Botnet) the network traffic blocked and the process terminated.
If the botnet classification result gives a negative result (Benign), no action is required.

The overall algorithmic approach of Ladon that is proposed herein is described clearly and in details in the following Figure 3.

Trying a comprehensive analysis of the way the above architecture works, we clearly realize that in the proposed method, the detection and disposal of the malware or botnet is done in dead time, before disturb the operation of entire system. This innovation creates new perspectives in the design architecture of the network operating systems, which adopt smart defense mechanisms against sophisticated attacks and zero-day exploits. In this way it adds a higher degree of integrity to the rest of the security infrastructures.

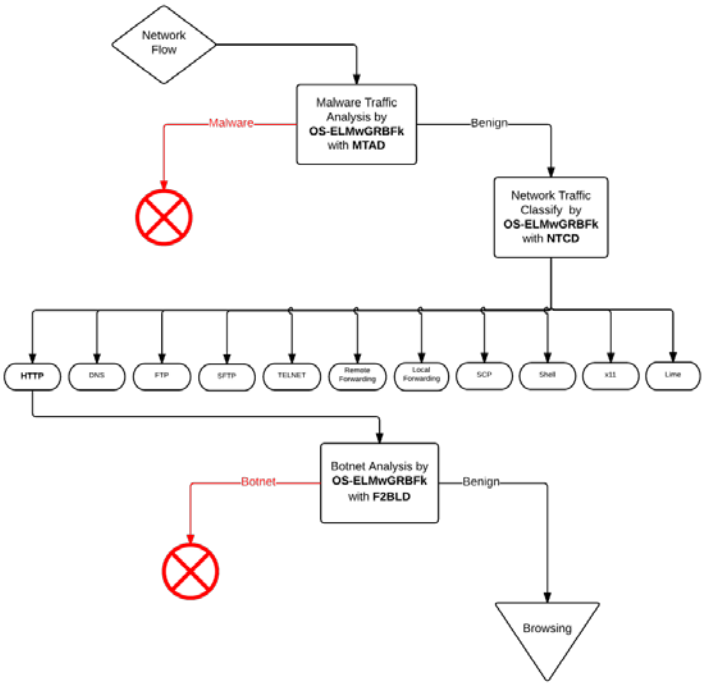

Figure 3: Graphical depiction of Ladon methodology

5 Dataset used for testing the Ladon system

Three datasets with high complexity were constructed and used for testing by Ladon. The first Malware Traffic Analysis Dataset (MTAD) comprised of 32 independent variables and 2 classes (benign or malware). This dataset containing 73,469 patterns (37,127 benign samples they were chosen from the Pcaps from National CyberWatch Mid-Atlantic Collegiate Cyber Defense Competition and 36,342 malicious samples they were chosen from http://malware-traffic-analysis.net/) [34].

The second Network Traffic Classification Dataset (NTCD) comprised of 22 independent variables and 11 network traffic classes (TELNET, FTP, HTTP, DNS, Lime, Local Forwarding, Remote Forwarding, SCP, SFTP, x11 and Shell).

This dataset containing 137,050 patterns they were chosen from the Pcaps from Information Technology Operations Center (ITOC), US Military Academy [35].
Finally, the third Fast-Flux Botnet Localization Dataset (F2BLD) comprised of 15 independent variables and 2 classes (benign or botnet). This dataset containing 131,374 patterns (100,000 URLs they were chosen randomly from the database with the 1 million most popular domain names of Alexa, 16,374 malicious URLs from the Black Hole DNS database and 15,000 malicious URLs they were created based on the timestamp DGA algorithm) [11].

In the preprocessing process the duplicate records and records with incompatible characters were removed. Also the datasets are determined and normalized to the interval [-1,1] in order to phase the problem of prevalence of features with wider range over the ones with a narrower range, without being more important [36].

6 Results and comparative analysis

The performance of the proposed OSEL_RBF is evaluated by comparing it with RBFANN, GMDH, PANN, FNNGA, FNNPSO, FNNACO and FNNES learning algorithms. Regarding the overall efficiency of the methods, the results show that the OSEL_RBF has much better generalization performance and more accurate classification output from the other compared algorithms.

The detailed accuracy by class comparison of the other algorithms is shown in Table 1. In all cases the hold out approach was used (70% training, 15% validation and 15% testing).
According to this comparative analysis, it appears that OSEL_RBF is highly suitable method for applications with huge amounts of data such that traditional learning approaches that use the entire data set in aggregate are computationally infeasible. This algorithm successfully reduces the problem of entrapment in local

minima in training process, with very fast convergence rates. These improvements are accompanied by high classification rates and low test errors as well. The performance of proposed model was evaluated in three network security datasets and the real-world sophisticated scenarios. The experimental results showed that the OSEL_RBF has better generalization performance at a very fast learning speed and more accurate and reliable classification results. The final conclusion is that the proposed method has proven to be reliable and efficient and has outperformed at least for this security problem the other approaches.

Table 1: Comparison between algorithms

	MTADataset		**NTCDataset**		**F2BLDatset**	
Classifier	ACC	RMSE	ACC	RMSE	ACC	RMSE
OSEL_RBF	**98,2%**	**0.3284**	**99,6%**	**0.2951**	**95,8%**	**0.4882**
RBFNN	89,8%	0.5766	91,3%	0.5514	82,9%	0.6244
GMDH	94,4%	0.5017	97,8%	0.3983	88,8%	0.5831
PANN	90,9%	0.5633	96,6%	0.4512	87,3%	0.5937
FFNNGA	96,7%	0.4972	99,1%	0.3048	95,3%	0.4899
FFNNPSO	96,2%	0.4911	99,2%	0.3009	95,1%	0.4907
FFNNACO	89,4%	0.5791	92,7%	0.5336	80,6%	0.6389
FFNNES	90,1%	0.5716	93,5%	0.5125	79,7%	0.6419

7 Discussion – Conclusions

A novel bio-inspired intelligence cyber-threat management system namely Ladon, has been introduced. It performs classification by using OSEL_RBF algorithm. The classification performance and the accuracy of the proposed model were experimentally explored based on several scenarios and reported very

promising results. Moreover, Ladon is an effective cross-layer system of network supervision, which enriches the lower layers of the system (Transport, Network and Data). It amplifies in an intelligent manner the upper layers (Session, Presentation and Application) with capabilities of automated control. This is done to enhance the energetic security and the mechanisms of reaction of the general system, without special requirements. In this way it adds a higher degree of integrity to the rest of the security infrastructure of Network Operating Systems. The most significant innovation of this methodology is that it offers high learning speed, ease of implementation, minimal human intervention and minimum computational power and resources to properly classify network traffic, label malware traffic and identify fast-flux botnets with high accuracy and generalization.

Future research could involve its model under a hybrid scheme, which will combine semi supervised methods and online learning for the trace and exploitation of hidden knowledge between the inhomogeneous data that might emerge. Also, Ladon could be improved towards a better online learning with self-modified the number of hidden nodes. Moreover, additional computational intelligence methods could be explored, tested and compared on the same security task in an ensemble approach. Finally, the ultimate challenge would be the scalability of Ladon with other kernels in parallel and distributed computing in a real-time system.

References

[1] Nazario J., Holz T.: As the net churns: Fast-flux botnet observations, MALWARE ’08, 3rd International Conference on Malicious and Unwanted Software, (2008).

[2] CISCO: WAN and Application Optimization Solution Guide, Cisco Validated Design Document, Version 1.1, (2008), http://www.cisco.com/c/en/us/td/docs/nsite/enterprise/wan/wan_optimization/wan_opt_sg.pdf, CISCO press.

[3] http://www.malware-traffic-analysis.net/

[4] Perdisci R., Corona I., Giacinto G., Early Detection of Malicious Flux Networks via Large-Scale Passive DNS Traffic Analysis, (2012), Published by the IEEE Computer Society.

[5] Xie Y., Yu F., Achan K., Panigrahy R., Hulten G., Osipkov I., Spamming botnets: Signatures and characteristics, (2008), ACM SIGCOMM Computer Communication Review.

[6] Zhao D., Traore I., Sayed B., Lu W., Saad S., Ghorbani A., Botnet detection based on traffic behavior analysis and flow intervals (2013), *Journal Computer Security*, **39**, (2016).

[7] Demertzis K., Iliadis L.: A Hybrid Network Anomaly and Intrusion Detection Approach Based on Evolving Spiking Neural Network Classification, (2014), Communications in Computer and Information Science, **441**, (2014), 11-23, 10.1007/978-3-319-11710-2_2.

[8] Demertzis K., Iliadis L.: Evolving Computational Intelligence System for Malware Detection, *Lecture Notes in Business Information Processing*, **178**, (2014), 322-334.

[9] Demertzis K., Iliadis L.: Bio-Inspired Hybrid Artificial Intelligence Framework for Cyber Security, *Computation, Cryptography, and Network Security*, (2014), 161-193, Computation, Cryptography, and Network Security, DOI 10.1007/978-3-319-18275-9_7, Springer.

[10] Demertzis K., Iliadis L.: Bio-Inspired Hybrid Intelligent Method for Detecting Android Malware (2014), Proceedings of 9th International Conference on Knowledge, Information and Creativity Support Systems (KICSS 2014), ISBN: 978-9963-700-84-4.

[11] Demertzis K., Iliadis L.: Evolving Smart URL Filter in a Zone-based Policy Firewall for Detecting Algorithmically Generated Malicious Domains, Statistical Learning and Data Sciences Volume 9047 of the series Lecture Notes in Computer Science pp 223-233, Third International Symposium, SLDS 2015, Egham, UK, (April 20-23, 2015), Proceedings, DOI 10.1007/978-3-319-17091-6_17, Springer.

[12] Demertzis K., Iliadis L.: SAME: An Intelligent Anti-Malware Extension for Android ART Virtual Machine, Computational Collective Intelligence, *Lecture Notes in Computer Science*, **9330**, (2015), 235-245, 7th International Conference, ICCCI 2015, Madrid, Spain, *Proceedings*, Part II, DOI 10.1007/978-3-319-24306-1_23, Springer.

[13] Liang N.-Y., Huang G.-B., Saratchandran P., Sundararajan N.: A Fast and Accurate On-line Sequential Learning Algorithm for Feedforward Networks, *IEEE Transactions on Neural Networks*, **17**(6), (2006), 1411-1423.

[14] Cambria E., Guang-Bin H.: Extreme Learning Machines, (2013), IEEE InTeLLIGenT SYSTemS, 541-1672/13.

[15] Huang G.-B. , Liang N.-Y., Rong H.-J., Saratchandran P., Sundararajan N.: On-line sequential extreme learning machine, (2005), IASTED.

[16] Mirjalili S., Mirjalili S. M., Lewis A., Let A Biogeography-Based Optimizer Train Your Multi-Layer Perceptron, (2014), *Information Sciences*, In press, DOI: http://dx.doi.org/10.1016/j.ins.2014.01.038

[17] Yingqiu L., Wei L., Yunchun L., Network Traffic Classification Using K-means Clustering, Computer and Computational Sciences, IMSCCS 2007, (2007), DOI:10.1109/IMSCCS.2007.52.

[18] Bereket M., Carvalho M.M., Ham F.M., Network Traffic Classification Using A Parallel Neural Network Classifier Architecture, (2011), CSIIRW '11, October 12-14, ACM, 978-1-4503-0945-5.

[19] Quoc D. L., D'Alessandro V., Park B., Romano L., Fetzer C., Scalable Network Traffic Classification Using Distributed Support Vector Machines,

(2015), *IEEE 8th International Conference on Cloud Computing* (CLOUD), (2015), 1008 - 1012, DOI:10.1109/CLOUD.2015.138.

[20] Comar P.M., Lei L., Saha S., Pang-Ning Tan N., Combining supervised and unsupervised learning for zero-day malware detection, INFOCOM, *2013 Proceedings IEEE*, (2013), 2022 - 2030, DOI:10.1109/INFCOM.2013.6567003.

[21] Bailey M., Oberheide J., Andersen J., Mao Z. M., Jahanian F., Nazario J., Automated classification and analysis of internet malware, in: C. KrÃijgel, R. Lippmann, A. Clark (Eds.), RAID, *Lecture Notes in Computer Science of Springer*, **4637**, (2007), 178–197.

[22] Langin C., Zhou H., Rahimi S., Gupta B., Zargham M., Sayeh M., A self-organizing map and its modeling for discovering malignant network traffic, (2009), in: Computational Intelligence in Cyber Security, IEEE, 122 –129, doi:10.1109/CICYBS.2009.4925099.

[23] Brezo F., Gaviria de la Puerta J., Ugarte-Pedrero X., Santos I., Bringas P.G., Barroso D., Supervised classification of packets coming from a HTTP botnet, (2012), *Informatica*, XXXVIII Conferencia Latinoamericana En, 1 - 8, DOI: 10.1109/CLEI.2012.6427168.

[24] Stevanovic M., Pedersen J. M.: Machine learning for identifying botnet network traffic, *Technical report*, (2013), Aalborg Universitet, http://vbn.aau.dk/files/75720938/paper.pdf.

[25] Perdisci R., Corona I., Dagon D., Lee W., Detecting malicious flux service networks through passive analysis of recursive dns traces, (2009), in: ACSAC ’09, IEEE Computer Society, Washington, DC, USA, (2009), 311–320. doi:10.1109/ACSAC.2009.36.

[26] Bailey M., Cooke E., Jahanian F., Xu Y., Karir M., A survey of botnet technology and defenses, *Cybersecurity Applications Technology*, (2009), 299–304.

[27] Feily M., Shahrestani, A survey of botnet and botnet detection, Emerging Security Information, SECURWARE ’09, (2009), 268-273, doi:10.1109/SECURWARE.2009.48.

[28] Zeidanloo H., Shooshtari M., Amoli P., Safari M., Zamani M., A taxonomy of botnet detection techniques, ICCSIT, **2**, (2010), 158–162.

[29] Cheng C., Peng T. W., Guang-Bin H., Extreme learning machines for intrusion detection, IJCNN, *International Joint Conference*, (2012),DOI: 10.1109/IJCNN.2012.6252449.

[30] Hsu C.-H., C.-Y. Huang, Chen K.-T., Fast-flux bot detection in real time, (2010), in 13th international conference on Recent advances in intrusion detection, ser. RAID’10.

[31] Haffner P., Sen S., Spatscheck O., Wang D., ACAS, Auto-mated Construction of Application Signatures, *Proceedings of the* ACM SIGCOMM, (2005),197-202.

[32] Alshammari R., Zincir-Heywood N. A., A flow based approach for SSH traffic detection, *IEEE International Conference on Cy-bernetics*, ISIC, (2007), 296-301.

[33] Holz T., Gorecki C., Rieck K., Freiling F.: Measuring and detecting fast-flux service networks, (2008), in NDSS ’08: Proceedings of the Network & Distributed System Security Symposium.

[34] http://malware-traffic-analysis.net/

[35] http://www.netresec.com/?page=PcapFiles

[36] Iliadis L., *Intelligent Information Systems and applications in risk estimation*, A. Stamoulis publication, Thessaloniki, Greece, 2008.