A Spiking One-Class Anomaly Detection Framework for Cyber-Security on Industrial Control Systems

Konstantinos Demertzis[1], Lazaros Iliadis[2], Stefanos Spartalis[3]

[1,2] School of Engineering, Department of Civil Engineering,
Faculty of Mathematics Programming and General courses,
Democritus University of Thrace, Kimmeria, Xanthi, Greece,
kdemertz@fmenr.duth.gr[1], liliadis@civil.duth.gr[2]
[3]School of Engineering, Department of Production
and Management Engineering, Democritus University of Thrace,
Kimmeria 67100, Xanthi, Greece
sspart@pme.duth.gr[3]

Abstract. Developments and upgrades in the field of industrial information technology, particularly those relating to information systems' technologies for the collection and processing of real-time data, have introduced a large number of new threats. These threats are primarily related to the specific tasks these applications perform, such as their distinct design specifications, the specialized communication protocols they use and the heterogeneous devices they are required to interconnect. In particular, specialized attacks can undertake mechanical control, dynamic rearrangement of centrifugation or reprogramming of devices in order to accelerate or slow down their operations. This may result in total industrial equipment being destroyed or permanently damaged. Cyber-attacks against *Industrial Control Systems* which mainly use *Supervisory Control and Data Acquisition* (SCADA) combined with *Distributed Control Systems* are implemented with *Programmable Logic Controllers*. They are characterized as *Advanced Persistent Threats*. This paper presents an advanced *Spiking One-Class Anomaly Detection Framework* (SOCCADF) based on the *evolving Spiking Neural Network* algorithm. This algorithm implements an innovative application of the *One-class classification methodology* since it is trained exclusively with data that characterize the normal operation of ICS and it is able to detect divergent behaviors and abnormalities associated with APT attacks.

Keywords: Industrial Control Systems, SCADA, PLC, APT, evolving Spiking Neural Network, One-Class Classification, Anomaly Detection

1. Introduction

1.1 Industrial Control System (ICS)

Automation and remote control are the most important methods used by critical infrastructure in order to improve productivity and quality of service. In this respect, the efficient management of industrial IT and the introduction of sophisticated automation systems, have contributed to the emergence of advanced Cyber-attacks

against Industrial Control Systems (ICS) [1]. These systems are active devices of the infrastructure network whereas the successful completion of specialized activities requires all the devices used to be accurately controlled and totally reliable. Typical ICS automation devices are SCADA, Distributed Control Systems (DCS), Programmable Logic Controllers (PLC) together with the sensors used in control loops to collect the measurements [2]. The above systems are properly interconnected to allow remote monitoring and control of processes with high response rates, even in cases where the devices are distributed between different distant points. The most important categories of ICS applications concern water and sewage network infrastructure, natural gas, fuel and chemicals, building and building management systems in general, power generation and distribution, automation of road arteries - railways - airports - metro and telecom infrastructure management and Networks [3].

1.2 APT against ICS

Integration into critical ICS infrastructures, especially where ICS includes features related to communications and internet technologies, introduces risks and new threats to the security and to the uninterrupted smooth operation of the critical infrastructure they include [4]. Exploiting the vulnerabilities of the wired and wireless communication networks used to interface these devices, as well as the vulnerabilities associated with their operating control, may cause total taking of critical devices and applications, or unavailability of necessary services even partial or total destruction of them [5]. The consequences may be severe. Critical infrastructures, however, are exposed not only to new risks due to the vulnerabilities of the communications and computer network (*malware, spyware, ransomware*) but also to the dangers inherent in the heterogeneity currently characterized by these systems [6]. Physical attacks interrupt service provision, while cyberattacks attempt to gain remote access for their benefit [7]. In any case, attacks against ICS are characterized as Advanced Persistent Threats (APT)s, as cybercriminals are fully familiar with specialized methods and tools for exploiting unknown vulnerabilities to the public (zero days). Most of the time, they are highly competent and organized, they are funded and they have significant incentives. The APT attack usually follows four steps [8].

Access: The attacker gathers as much information as possible and targets to specific ICS elements (SCADA systems) by using zero days' malware. In this way, he will exploit weaknesses that will provide access [6][7][8].

Discovery: After gaining access to the critical infrastructures network, discovery tactics are applied to the processes performed on it. For example, long term analysis and monitoring of the information flow in the location of the attacker. This is done to disclose information such as server mode, engineering workstation positions, architecture of local devices controlling individual components-units, connected Master Stations and so on [6][7][8].

Control: Once the network architecture and ICS mode have been understood, there are several ways to control the system. Typical potential targets that can control the network are the engineering workstation used to upgrade the software, database systems and the application server that hosts various applications used in the general system.

Hiding: In this step, attackers hide all the elements of the attacks by deleting specific folders that can betray their presence in automation systems [6][7][8]. Attacks designed to attack SCADA systems (eg the Stuxnet virus) have created many doubts about the level of critical infrastructure security and serious concerns about the consequences, as society is heavily dependent on the routine operations of these infrastructures. Intelligent anomaly detection is a process of high importance [6][7][8].

1.3 Anomaly Detection

The concept Anomaly Detection (AD) [9] refers to the recognition of standards from a set of data that exhibit a different behavior than expected. The goal is high level detection of possible anomalies combined with low false alert rates. The AD can be supervised, (performed on a training set containing normal versus anomalous classes) and semi-supervised where the training set is usually characterized as normal. The usual semi-supervised approach constructs a model to respond to normal behavior which is applied to determine the anomalies in the test data. In the unsupervised approach, no training is performed. It is based on the assumption that the normal incidents are more than the extreme ones in the testing data. If this reasoning does not apply, then the techniques have a large error rate [10]. Several abnormality detection techniques have been proposed in the literature, with more popular the *One-class classification* (*OCC)* methods, the *Distance Based* [11] ones, the *Replicator Neural Networks* [12] and the *Conditional Anomaly Detection* [13].

1.4 One-class classification

In machine learning (ML), the OCC method [14], tries to find objects of a particular class among all objects, by learning from a training set containing only objects of this class. Typically, these algorithms aim to implement classification models in which the negative class is absent, either because the missing class is not sampled, or due to the fact that it is difficult to do so. This mode of operation in which classifiers are required to determine effectively and reliably the boundaries of the class separation only based on the knowledge of the positive class, is a particularly complex problem of ML. When only data from the target class is available, the classifier is trained to receive target objects and to reject the ones that deviate significantly. Finally, it should be noted that the basic concept in OCC problem solving is the reverse of the generalization that is being pursued in other ML problems [15]. Particularly, it is intended that the parameter setting is fully defined, even if this exponentially increases the complexity of the classifier, provided it is able to correctly classify the target data.

1.5 The proposed SOCCADF

Identifying anomalies that lead to scrapping or deprecation of ICS devices is an extremely complex matter, due to the fact that ART attacks are the most advanced and highly intelligent cyber-engineering techniques, operating under a chaotic architecture

of industrial networks. Also, given the passive operation of traditional security systems which in most cases are unable to detect serious threats, alternative more active and more meaningful methods of locating ART attacks are necessary. Our research team has developed several innovative approaches of computational intelligence towards security threats identification. Herein we are proposing an intelligent system that significantly enhances the security level of critical infrastructures, by consuming the minimum level of resources. It is the *Spiking One-Class Anomaly Detection Framework (SOCCADF)*, which exploits for a first time a special operation form of the *evolving Spiking Neural Network* (*eSNN*) algorithm, in order to effectively classify the ICS anomalies resulting from APT attacks [16][17][18][19][20][21][22][23][24][25][26] [27][28][29][30].

1.6 Innovation

An important innovation of SOCCADF is the use for first time of the *eSNN* algorithm (incorporated in Spiking Neural Networks) for the implementation of an OCC anomaly detection system. SNNs simulate the functioning of biological brain cells in a most realistic way and they rationally model data in a spatiotemporal mode. The produced signals are transmitted by discharges of temporal pulses, where duration and frequency of time pulses between neurons are the crucial factors. Also, innovation is attributed to the addition of artificial intelligence at the level of *real-time* analysis of industrial equipment, which greatly enhances the defensive mechanisms of critical infrastructures. It is much easier to locate ARTs Attacks by controlling the interdependencies of ICS at all times. Finally, there is innovation in the data selection process. This data has emerged after extensive research in the way ICS work and after comparisons and tests regarding the boundaries of their inherent behavior that determine their classification in normal or outliers.

2 Literature review

Moya et al. [31] originated the term *One-Class Classification* in their research. Different researchers have used other terms such as *Single Class Classification* [32][33] [34]. These terms originated as a result of different applications to which OCC has been applied. Juszczak [35] has defined *One-Class Classifiers* as class descriptors able to learn restricted domains in a multi-dimensional pattern space, using primarily just a positive set of examples. Luo et al., [36] have proposed a cost-sensitive *OCC-SVM* algorithm for intrusion detection problem. Their experiments have suggested that giving different cost or importance to system users than to processes results in higher performance in intrusion detection than other system calls. Shieh and Kamm [37] have introduced a *kernel* density estimation method to give weights to the training data objects, such that the outliers get the least weights and the positive class members get higher weights for creating bootstrap samples.
Soupionis et al., [38] proposed a combinatorial method for automatic detection and classification of faults and cyber-attacks occurring on the power grid system when there is limited data from the power grid nodes due to cyber implications. In addition, Tao et

al. have described the network attack knowledge, based on the theory of the factor expression of knowledge, and studied the formal knowledge theory of SCADA network from the factor state space and equivalence partitioning. This approach utilizes the *factor neural network* (FNN) theory which contains high-level knowledge and quantitative reasoning described to establish a predictive model including analytic FNN and analogous FNN. This model abstracts and builds an equivalent and corresponding network attack and defense knowledge factors system.
Also, Qin et al., [39] have introduced an analytic factor neuron model which combines machine reasoning based on the cloud generator with the FNN theory. The FNN model is realized based on mobile intelligent agent and malicious behavior perception technology. 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 practices. Qian and Sherif [40] have applied autonomic computing technology to monitor SCADA system performance, and proactively estimate upcoming attacks for a given system model of a physical infrastructure. Finally, Yasakethu and J. Jiang in [41] have introduced a new *European Framework-7* project "*Cockpit CI* (*Critical Infrastructure*)" and roles of intelligent machine learning methods to prevent SCADA systems from cyber-attacks.

3. evolving Spiking Neural Network (eSNN)

The eSNNs based on the "Thorpe" neural model [42] are modular connectionist-based systems that evolve their structure and functionality in a continuous, self-organized, on-line, adaptive, interactive way from incoming information [43]. In order to classify real-valued data sets, each data sample is mapped into a sequence of spikes using the *Rank Order Population Encoding* (ROPE) technique [44] [45]. In this encoding method, neurons are organized into neuronal maps which share the same synaptic weights. Whenever a synaptic weight is modified, the same modification is applied to the entire population of neurons within the map. Inhibition is also present between each neuronal map. If a neuron spikes, it inhibits all the neurons in the other maps with neighboring positions. This prevents all the neurons from learning the same pattern. When propagating new information, neuronal activity is initially reset to zero. Then, as the propagation goes on, each time one of their inputs fire, neurons are progressively desensitized. This is making neuronal responses dependent upon the relative order of firing of the neuron's afferents [46] [47]. Also in this model, the neural plasticity is used to monitor the learning algorithm by using one-pass learning method. The aim of this learning scheme is to create a repository of trained output neurons during the presentation of training samples [48].

4. ICS Anomaly Datasets

In order to carry out the research and evaluate the proposed model, 3 suitable datasets were chosen to best match the ICS communication and transaction data [49]. These sets

include data logs from a gas pipeline, a lab scale water tower, and a lab scale electric transmission system. The logs include flagged network transactions during the normal operation of specific ICSs, as well as transactions during 35 different cyber-attacks. In addition to the logs, measurements include normal behavior as well as abnormalities detected during attacks that were simulated in a virtual ICS environment, including Human Machine Interface (HMI), Virtual Physical Process, and Virtual Programmable Logic Controller (VPLC) and a Virtual Network (VN). Although the configured virtual systems do not have physical limits to the size of the modeling they simulate, the virtual platform on which the data was collected was implemented by escalating the ICS to represent their operating states in the most realistic way [49].

All three data sets contain network transaction data, preprocessed in a way to strip *lower layer transmission* data (*TCP, MAC*) [49]. The "*water_tower_dataset"* includes 23 independent parameters and 236,179 instances, from which 172,415 are normal and 63,764 outliers. In the case of the "*water_train_dataset"* the algorithm was trained by using 86,315 normal instances, whereas the rest 86,100 normal instances and the 63,764 outliers comprised the testing set "*water_test_dataset*". The "*gas_dataset* " contains 26 independent parameters and 97,019 instances, from which 61,156 are normal and 35,863 outliers. For the "*gas_train_dataset*" case 30,499 normal instances are used for the training process, whereas 30,657 normal instances and 35,863 outliers comprise the "*gas_test_dataset*". Finally, the "*electric_dataset"* includes 128 independent features and 146,519 instances, from which 90,856 are normal and 55,663 outliers. The "*electric_train_dataset*" has 45,402 normal instances. The rest 45,454 normal ones and the 55,663 outliers comprise the "*electric_test_dataset*". More details related to the dataset and to the data selection process can be found in [49].

5. Methodology

5.1 Description of the eSNN one-class classification method

The proposed methodology uses an eSNN classification approach in order to detect and verify the anomalies on ICS. The topology of the developed eSNN is strictly feed-forward, organized in several layers and weight modification occurs on the connections between the neurons of the existing layers. The encoding is performed by ROPE technique with 20 Gaussian Receptive Fields (GRF) per variable [46]. The data are normalized to the interval [-1, 1] and so the coverage of the Gaussians is determined by using *i_min* and *i_max*. Each input variable is encoded independently by a group of one-dimensional GRF. The GRF of neuron i is given by its center μ_i by equation (1) and width σ by equation (2) [46]

$$\mu_i = I^n_{min} + \frac{2i-3}{2}\frac{I^n_{max} - I^n_{min}}{M-2} \quad (1) \qquad \sigma = \frac{1}{\beta}\frac{I^n_{max} - I^n_{min}}{M-2} \quad (2)$$

where $1 \leq \beta \leq 2$ and the parameter β directly controls the width of each Gaussian receptive field. When a neuron reaches its threshold, it spikes and inhibits neurons at equivalent positions in the other maps so that only one neuron will respond at any location [50]. Every spike triggers a time based Hebbian-like learning rule that adjusts

the synaptic weights. For each training sample i with class label l, a new output neuron is created and fully connected to the previous layer of neurons, resulting in a real-valued weight vector $w^{(i)}$ with $w_j^{(i)} \in R$ denoting the connection between the pre-synaptic neuron j and the created neuron i. In the next step, the input spikes are propagated through the network and the value of weight $w_j^{(i)}$ is computed according to the order of spike transmission through a synapse [46]

$$j: w_j^{(i)} = (m_l)^{order(j)} \qquad (3)$$

where j is the *p*re-synaptic neuron of i. Function *order(j)* represents the rank of the spike emitted by neuron j. The firing threshold $\theta^{(i)}$ of the created neuron i is defined as the fraction $c_l \in$R, $0 < c_l < 1$, of the maximal possible potential [46]

$$u_{max}^{(i)}: \theta^{(i)} \leftarrow c_l u_{max}^{(i)} \qquad (4) \qquad u_{max}^{(i)} \leftarrow \sum_j w_j^{(i)} (m_l)^{order(j)} \qquad (5)$$

The weight vector of the trained neuron is compared to the weights corresponding to neurons already stored in the repository. Two neurons are considered too "similar" if the minimal *Euclidean* distance between their weight vectors is smaller than a specified similarity threshold s_l (the eSNN object uses optimal similarity threshold s=0.6) [46]. Both the firing thresholds and the weight vectors were merged according to equations (6) and (7):

$$w_j^{(k)} \leftarrow \frac{w_j^{(i)} + N w_j^{(k)}}{1+N} \qquad (6) \qquad \theta^{(k)} \leftarrow \frac{\theta^{(i)} + N\theta^{(k)}}{1+N} \qquad (7)$$

Integer N denotes the number of samples previously used to update neuron k. The merging is implemented as the average of the connection weights, and of the two firing thresholds. After merging, the trained neuron i is discarded and the next sample processed. If no other neuron in the repository is similar to the trained neuron i, the neuron i is added to the repository as a new output [46].
All parameters of eSNN included in this search space, are optimized according to the *Versatile Quantum-inspired Evolutionary* Algorithm (vQEA) [46].

5.2 Threshold Deciding Criteria for the Proposed Method

The choice of the threshold value used for class separation is the most important and critical factor for the success of the OCC approach. In order to determine the threshold, and given that the training set contains only positive samples, this paper proposes a reliable heuristic selection method based solely on criteria of merit. In particular, the proposed algorithm assumes that a distance function d between the objects and the target class is employed in the training phase. The determination of the *threshold* θ for the class separation (normal or outlier) is performed in a way that it discards a set of training samples, most of which diverge from the target class, in order to strengthen the classifier. Even when all samples are correctly labeled, the rejection of a small but representative rate of training samples helps the classifier to learn the most representative set of training samples. This approach significantly enhances active security of critical infrastructure. The following pseudocode presents the algorithmic approach for the determination of the class separation threshold θ.
Algorithm 1: Optimal Threshold

***Optimal Threshold**:*
1: Calculate the error using Euclidean distance between actual and predicted on each training data;
2: Arrange the error in decreasing order;
3: Set the threshold at rejection of 10% most erroneous data (false negative rate at the rate of 10%);

6. Results and comparative analysis

The efficiency of the proposed OCC classifier is estimated by employing the following statistical indices. The numbers of misclassifications are related to the False Positive (FP) and False Negative (FN) indices A FP is the number of cases where you wrongfully receive a positive result and the FN is exactly the opposite. On the other hand, the True Positive (TP) is the number of records where you correctly receive a Positive result. The True Negative (TN) is defined respectively. The *True Positive rate* (TPR) also known as *Sensitivity*, the *True Negative rate* also known as *Specificity* (TNR) and the *Total Accuracy* (TA) are defined by using equations 8, 9, 10 respectively [50]:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \ (8) \quad \mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN+FP}} \ (9) \quad \mathrm{TA} = \frac{\mathrm{TP+TN}}{\mathrm{N}} \ (10)$$

The Precision (PRE) the Recall (REC) and the F-Score indices are defined as in equations 11, 12 and 13 respectively:

$$\mathrm{PRE} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} (11) \quad \mathrm{REC} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \ (12) \quad \mathrm{F-Score} = 2\mathrm{X}\frac{\mathrm{PRE\ X\ REC}}{\mathrm{PRE\ +REC}} \ (13)$$

The *ROC* (*Receiver Operating Characteristic*) is a standard technique for summarizing classifier performance over a range of trade-offs between TP and FP error rates. ROC curve is a plot of *Sensitivity (the ability of the model to predict an event correctly)* versus *1-Specificity* for the possible cut-off classification probability values π_0 [50]. The *Precision* measure shows what percentage of positive predictions where correct, whereas *Recall* measures the percentage of positive events that were correctly predicted. The *F-Score* can be interpreted as a weighted average of the precision and recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F-Score is usually more useful than accuracy and it works best if false positives and false negatives have similar cost, in this case. Finally, the ROC curve is related in a direct and natural way to cost/benefit analysis of diagnostic decision making [50].

In the training process of the OCC, only the probability density of the positive class is known, which means that during training only the number of the positive class items that are not classified correctly (FN) can be minimized. Basically, this means that due to the fact that there are no examples of samples' distribution belonging to other classes (outliers) in the training phase, it is not possible to estimate the number of objects of other classes that were misclassified as Positive (FP) by the OCC classifier. So given the fact that *TP + FN = 1* the algorithm during training can provide estimations only for TP and FN. However during testing all four indices (TP,FN, TN, FP) can be obtained.

The proposed system manages to operate effectively in a particularly complex cyber security problem with high levels of accuracy. The performance of the SOCCADF is evaluated by comparing it with OCC Support Vector Machines (OCC-SVM) and OCC *Combining Density and Class Probability Estimation* (OCC-CD/CPE) learning.

Regarding the overall efficiency of the method, the results show that the proposed system significantly outperforms the other algorithms. The following table 1, presents the analytical values of the predictive power of the SOCCADF and the corresponding results when competitive algorithms were used.

Table 1. Comparison between algorithms

water_tower_dataset						
Classifier	Classification Accuracy & Performance Metrics					
	Total Accuracy	RMSE	Precision	Recall	F-Score	ROC Area
OCC-eSNN	**98.08%**	**0.1305**	**0.981**	**0.981**	**0.981**	**0.994**
OCC-SVM	98.01%	0.1312	0.980	0.980	0.980	0.995
OCC-CD/CPE	96.75%	0.1389	0.975	0.975	0.975	0.980
gas_dataset						
Classifier	Classification Accuracy & Performance Metrics					
	Total Accuracy	RMSE	Precision	Recall	F-Score	ROC Area
OCC-eSNN	**98.82%**	**0.0967**	**0.988**	**0.988**	**0.988**	**0.995**
OCC-SVM	97.98%	0.0981	0.980	0.980	0.980	0.990
OCC-CD/CPE	95.67%	0.1284	0.960	0.960	0.960	0.975
electric_dataset						
Classifier	Classification Accuracy & Performance Metrics					
	Total Accuracy	RMSE	Precision	Recall	F-Score	ROC Area
OCC-eSNN	**98.30%**	**0.1703**	**0.983**	**0.983**	**0.983**	**0.999**
OCC-SVM	97.63%	0.1840	0.978	0.978	0.978	0.990
OCC-CD/CPE	97.02%	0.1897	0.970	0.970	0.970	0.985

7. Discussion and Conclusions

This comparison generates expectations for the identification of the SOCCADF as a robust anomaly detection model suitable for difficult problems. According to this comparative analysis, it appears that SOCCADF 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. The eSNN 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 a high complex dataset and the real-world sophisticated scenarios. The experimental results showed that the SOCCADF has better 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 the other approaches for the specific security problem.

Future research should include further optimization of the eSNN parameters, aiming to achieve an even more efficient and faster categorization process. Also, it would be important for the proposed framework to expand, based on "*metalearning"* methods to self-improve and redefine its parameters so that it can fully automate the process of

locating APT attacks. Finally, an additional element that could be studied in the direction of future expansion is the creation of an additional cross-sectional anomaly analysis system. This could act counter-diametrically on the philosophy of the eSNN classifier with potential enhancement of the system's efficiency.

References

[1] Falco, Joe, et al., IT Security for Industrial Control Systems, NIST Internal Report (NISTIR) 6859, (2002), http://www.nist.gov/customcf/get_pdf.cfm?pub_id=821684.

[2] Bailey, David, and Edwin Wright, (2003), Practical SCADA for Industry, Vancouver: IDC Technologies.

[3] Boyer, Stuart, (2010), SCADA: Supervisory Control and Data Acquisition. 4th ed. Research Triangle Park, North Carolina: International Society of Automation.

[4] Weiss, Joseph, (2003), "Current Status of Cybersecurity of Control Systems," Presentation to Georgia Tech Protective Relay Conference.

[5] Alvaro A. Cárdenas, Saurabh Amin, Shankar Sastry, (2008), "Research Challenges for the Security of Control Systems", 3rd USENIX workshop on Hot Topics in Security (HotSec '08). Associated with the 17th USENIX Security Symposium. San Jose, CA, USA.

[6] Vaishali S Raj, Dr. R. Manicka Chezhian, M.Mrithulashri, (2014), Advanced Persistent Threats & Recent High Profile Cyber Threat Encounters, International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 2, Issue 1.

[7] E. Hutchins, M. Cloppert, and R. Amin, (2010), Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains, In The 6th International Conference on Information-Warfare & Security, pp. 113–125.

[8] Aditya K Sood, Richard J. Enbody, (2013), Targeted Cyberattacks: A Superset of Advanced Persistent Threats". IEEE Security & Privacy, vol. 11, no. 1, pp. 54-61, doi:10.1109/MSP.2012.90.

[9] Chandola, V.; Banerjee, A.; Kumar, V. (2009). "Anomaly detection: A survey", ACM Computing Surveys. 41 (3): 1–58. doi:10.1145/1541880.1541882.

[10] Zimek, A.; Schubert, E.; Kriegel, H.-P. (2012). "A survey on unsupervised outlier detection in high-dimensional numerical data". Statistical Analysis and Data Mining. 5 (5): 363–387. doi:10.1002/sam.11161

[11] Knorr, E. M.; Ng, R. T.; Tucakov, V., (2000), "Distance-based outliers: Algorithms and applications". The VLDB Journal the International Journal on Very Large Data Bases. 8 (3–4): 237–253. doi:10.1007/s007780050006

[12] Hawkins, S., He, H., Williams, G.J., Baxter, R.A., (2002), Outlier detection using replicator neural networks. In: Kambayashi, Y., Winiwarter, W., Arikawa, M. (eds.) DaWaK 2002. LNCS, vol. 2454, pp. 170–180. Springer, Heidelberg

[13] M. Valko, G. Cooper, A. Seybert, S. Visweswaran, M. Saul, and M. Hauskrecht, (2008), "Conditional anomaly detection methods for patient-management alert systems," in Workshop on Machine Learning in Health Care Applications in The 25th International Conference on Machine Learning.

[14] Skabar, (2003), A.: Single-class classifier learning using neural networks: An application to the prediction of mineral deposits. In: Proceedings of the Second International Conference on Machine Learning and Cybernetics, vol. 4, pp. 2127–2132.

[15] Manevitz, L.M., Yousef, M., (2001), One-class svms for document classification. Journal of Machine Learning Research 2, 139–154.

[16] Demertzis K., Iliadis L., (2015), Intelligent Bio-Inspired Detection of Food Borne Pathogen by DNA Barcodes: The case of Invasive Fish Species Lagocephalus Sceleratus, Engineering Applications of Neural Networks, Vol 517 pp 89-99, DOI 10.1007/978-3-319-23983-5_9.

[17] Demertzis K., Iliadis L., (2014), A Hybrid Network Anomaly and Intrusion Detection Approach Based on Evolving Spiking Neural Network Classification. In: E-Democracy, Security, Privacy and Trust in a Digital World. Communications in Computer and Information Science, 441, 11-23. doi:10.1007/978-3-319-11710-2_2
[18] Demertzis K., Iliadis L., (2014), Evolving Computational Intelligence System for Malware Detection, In: Advanced Information Systems Engineering Workshops, Lecture Notes in Business Information Processing, 178, 322-334. doi: 10.1007/978-3-319-07869-4_30
[19] Demertzis K., Iliadis L., (2014, April), Bio-Inspired Hybrid Artificial Intelligence Framework for Cyber Security. Springer Proceedings 2nd Conference on CryptAAF: Cryptography Network Security and Applications in the Armed Forces, Springer, Athens, 161-193. doi: 10.1007/978-3-319-18275-9_7
[20] Demertzis K., Iliadis L., (2014, November). Bio-Inspired Hybrid Intelligent Method for Detecting Android Malware, Proceedings of the 9th KICSS 2014, Knowledge Information and Creative Support Systems, Cyprus, 231-243. ISBN: 978-9963-700-84-4.
[21] Demertzis K., Iliadis L., (2015, April), Evolving Smart URL Filter in a Zone-based Policy Firewall for Detecting Algorithmically Generated Malicious Domains. Proceedings SLDS (Statistical Learning and Data Sciences) Conference LNAI (Lecture Notes in Artificial Intelligence) 9047 Springer, Royal Holloway University London, UK, 223-233. doi: 10.1007/978-3-319-17091-6_17.
[22] Demertzis K., Iliadis L., (2015, September), SAME: An Intelligent Anti-Malware Extension for Android ART Virtual Machine. Proceedings of the 7th International Conference ICCCI, Lecture Notes in Artificial Intelligence 9330, 235-245. doi: 10.1007/978-3-319-24306-1_23.
[23] Demertzis K., Iliadis L., (2016), Computational Intelligence Anti-Malware Framework for Android OS, Special Issue on "Vietnam Journal of Computer Science (VJCS)", Springer, DOI 10.1007/s40595-017-0095-3.
[24] Demertzis K., Iliadis L., (2016), Detecting Invasive Species with a Bio-Inspired Semi Supervised Neurocomputing Approach: The Case of Lagocephalus Sceleratus, Special issues Neural Computing and Applications journal by Springer, DOI :10.1007/s00521-016-2591-2.
[25] Demertzis K., Iliadis L., (2016), SICASEG: A Cyber Threat Bio-Inspired Intelligence Management System, Journal of Applied Mathematics & Bioinformatics, vol.6, no.3, 2016, 45-64, ISSN: 1792-6602 (print), 1792-6939 (online), Scienpress Ltd.
[26] Bougoudis I., Demertzis K., Iliadis L., (2016), Fast and Low Cost Prediction of Extreme Air Pollution Values with Hybrid Unsupervised Learning, Integrated Computer-Aided Engineering, vol. 23, no. 2, pp. 115-127, DOI: 10.3233/ICA-150505, IOS Press.
[27] Bougoudis I., Demertzis K., Iliadis L., (2016), HISYCOL a Hybrid Computational Intelligence System for Combined Machine Learning: The case of Air Pollution Modeling in Athens, EANN Neural Computing and Applications, pp 1-16 DOI 10.1007/s00521-015-1927-7.
[28] Anezakis VD., Demertzis K., Iliadis L., Spartalis S., (2016a,), A hybrid soft computing approach producing robust forest fire risk indices. IFIP Advances in Information and Communication Technology, AIAI, Thessaloniki Greece, 475:191-203.
[29] Anezakis VD., Dermetzis K., Iliadis L., Spartalis S., (2016b), Fuzzy cognitive maps for long-term prognosis of the evolution of atmospheric pollution, based on climate change scenarios: The case of Athens. Lecture Notes in Computer Science including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, 9875:175-186. doi: 10.1007/978-3-319-45243-2_16.
[30] Bougoudis I., Demertzis K., Iliadis L., Anezakis VD., Papaleonidas A., (2016c), Semi-supervised hybrid modeling of atmospheric pollution in urban centers. Communications in Computer and Information Science, 629:51-63.
[31] Moya, M., Koch, M., Hostetler, L., (1993), One-class classifier networks for target recognition applications. In: Proceedings World Congress on Neural Networks. 797–801.

[32] D.T. Munroe and M.G. Madden, (2005), Multi-class and single-class classification approaches to vehicle model recognition from images. In Proc. of Irish Conference on Artificial Intelligence and Cognitive Science, Portstewart.
[33] H. Yu. (2003), SVMC: single-class classification with support vector machines. In Proc. of International Joint Conference on Artificial Intelligence, pages 567–572.
[34] R. El-Yaniv and M. Nisenson, (2007), Optimal single-class classification strategies - google scholar. In Proc. of the 2006 NIPS Conference, volume 19, pages 377–384. MIT Press.
[35] P. Juszczak, (2006), Learning to Recognise. A study on one-class classification and active learning. PhD thesis, Delft University of Technology.
[36] J. Luo, L. Ding, Z. Pan, G. Ni, and G. Hu, (2007), Research on cost-sensitive learning in one-class anomaly detection algorithms. In Autonomic and Trusted Computing, volume 4610 of Lecture Notes in Computer Science, pages 259–268. Springer Berlin Heidelberg.
[37] A.D. Shieh and D.F. Kamm, (2009), Ensembles of one class support vector machines. In Lecture Notes in Computer Science, volume 5519, pages 181–190. Springer-Verlag.
[38] Yannis Soupionis, Stavros Ntalampiras and Georgios Giannopoulos, (2016), DOI: 10.1007/978-3-319-31664-2_29 Vol 8985 of the book series Lecture Notes in Computer Science
[39] Yong Qin ; Xiedong Cao ; Peng Liang ; Qichao Hu ; Weiwei Zhang, (2014), Research on the analytic factor neuron model based on cloud generator and its application in oil&gas SCADA security defense, Cloud Computing and Intelligence Systems (CCIS), IEEE 3rd International Conference on, DOI: 10.1109/CCIS.2014.7175721
[40] Qian Chen and Sherif Abdelwahed (2013), A model-based approach to self-protection in computing system, Proceeding CAC '13 Proceedings of the ACM Cloud and Autonomic Computing Conference, Article No. 16
[41] S.L.P. Yasakethu and J. Jiang, (2013), Intrusion Detection via Machine Learning for SCADA System Protection, Learning and Development Ltd, Proceedings of the 1st International Symposium for ICS & SCADA Cyber Security Research.
[42] Thorpe S. J., Delorme A., Rullen R., (2001), Spike-based strategies for rapid processing, Neural Networks, Volume 14, Issues 6–7, 9 July 2001, Pages 715–725, Elsevier.
[43] Schliebs S., Kasabov N., (2013), Evolving spiking neural network—a survey, Evolving Systems 4: 87. doi:10.1007/s12530-013-9074-9, Springer.
[44] Delorme A., Perrinet L. & Thorpe S. J., (2000), Networks of Integrate-and-Fire Neurons using Rank Order Coding, Neurocomputing, 38-40(1-4), 539-545.
[45] Thorpe S. J. and Gautrais J., (1998), Rank order coding, In CNS '97: 6th conf on Computational neuroscience: trends in research, pages 113–118, Plenum Pr.
[46] Kasabov, N., (2002), Evolving connectionist systems: Methods and Applications in Bioinformatics, Brain study and intelligent machines, Springer.
[47] Wysoski S. G., Benuskova L., Kasabov N. K., (2006), Adaptive learning procedure for a network of spiking neurons and visual pattern recognition, Springer.
[48] Schliebs S., Defoin-Platel M., Kasabov N., (2009), Integrated feature and parameter optimization for an evolving spiking neural network, Neural Networks, Volume 22, Issues 5–6, Pages 623–632, 2009 International Joint Conference on Neural Networks.
[49] Thomas H. Morris, Zach Thornton, Ian Turnipseed, (2015), Industrial Control System Simulation and Data Logging for Intrusion Detection System Research, International Journal of Network Security (IJNS), Vol.17, No.2, PP.174-188.
[50] Fawcett, T., (2006), An introduction to ROC analysis. Pattern Recognition Letters. Elsevier Science Inc. **27**(8), 861-874. doi: http://doi.org/10.1016/j.patrec.2005.10.010

View publication stats