Journal of Theoretical and Applied Information Technology
15th May 2022. Vol.100. No 9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A REVIEW OF THE PARTICLE SWARM CLUSTERING METHOD FOR INTRUSION DETECTION IN IOT

MOHAMED EL BEKRI[1], OUAFAA DIOURI[2], DALILA CHIADMI[3]

[1]Mohammed V University, Avenue des Nations Unies, Agdal, Rabat, Morocco

[2]Mohammed V University , Avenue des Nations Unies, Agdal, Rabat, Morocco

[3]Mohammed V University V, Avenue des Nations Unies, Agdal, Rabat, Morocco

E-mail : [1]elbekri.mohamed@gmail.com

ABSTRACT

Intrusion Detection Systems (IDS) are security components that could serve IoT security. They do, however, face challenges in terms of autonomy, scalability, and efficiency. The pressing question is how to make the IDS extract the correct network's behavior while being intelligent enough to detect new intrusions. It is, therefore, essential to explore new possibilities that could lead to further improvement in the efficiency of these systems. Introducing particle swarm optimization in intrusion detection systems is a way to approach the problem differently. In this paper, we explain the combination of two techniques, machine learning, a field that provides robust methods for learning and knowledge extraction, and particle systems that include collaborative heuristics for search and detection. There is a shortage of researches which have addressed the IoT intrusion detection problem based on this combination. We will try to fill the gap and discuss the aspects to consider for implementing particle swarm clustering method for intrusion detection in IoT.

Keywords: *Particle systems; IoT; Intrusion Detection; Machine learning; Clustering; Particle Swarm Optimization*

1. INTRODUCTION

Internet of things is growing fast. Billions of objects known as "things" [1] are connected to the internet to bring entirely new services to end-users. However, security issues are rising due to the limited computational capacity and the lack of the defense mechanisms of some objects that can be targets of attacks.

Researchers have suggested innovative solutions to protect IoT networks from attacks in the last decades. Such solutions include deploying encryption mechanisms, advanced device monitoring, authentication, access control, etc.; however, as the security mechanisms tend to advance over time, so do the attacks. For example, the Mirai botnet [2] exploited IoT devices and caused massive distributed Denial of Service (DDoS) attacks.

Intrusion Detection Systems for IoT networks are still in their infancy; however, researchers have presented comprehensive studies on approaching the intrusion detection problem in an IoT Context [3]. The challenge is that the IDS must be less computational resource-demanding and sufficiently intelligent to learn better and identify unknown attacks. Conventional signature-based detection methods have been used for computer networks, but they cannot detect new types of intrusions because they don't have a corresponding signature yet. The signature database needs to be systematically reviewed to include newly discovered attacks. researchers have used data mining and machine learning algorithms to train on labeled network data to identify attack patterns after [4].

The limitation of these approaches is essentially linked to labels of data. Obtaining labeled data needs simulating IoT intrusions, but we would always be limited to only known attacks that we can emulate, in contrast, unknown attacks and new types which will emerge in the future will not be covered. Also, labeling data is a costly exercise for intrusion detection experts, especially in an IoT context.

Recently researchers moved to entirely new areas of inspiration such as biology, game theory, and fuzzy logic [5, 6, 7], etc. efforts are put to extract relevant ideas from these fields and make them accessible to the intrusion detection problem.

This paper follows this dynamic and explains the method behind the association of particle systems and clustering as an unsupervised machine learning

technique for optimizing the intrusion detection problem in the IoT context.

First, we will review the IoT-based IDSs, explain their detection techniques, the threats they have addressed, and the advantages and disadvantages of each method. Secondly, we will present a general architecture of an IoT-based IDS. Third, we will dive into the particle systems' origins and dynamics, focusing on the particle swarm optimization algorithm at the end of this section. Fourthly, we will explain the steps to build the intrusion detection-based particle swarm clustering approach and the underlying parameters to consider for implementation.

2. REVIEW OF IDS SYSTEMS FOR IOT

As with computer networks, an IDS-Intrusion Detection System- for IoT is a system that examines the network data to identify suspicious behavior then issues alerts when such behavior is determined. Although the main functions are anomaly detection and reporting, some intrusion detection systems go beyond these functions and take action when a malicious activity or anomalous traffic is detected, including blocking traffic from suspicious IP or mac addresses.

2.1 Types of IDSs

Intrusion detection systems can broadly be classified based on two parameters:

- The scope of the protection: Based on this parameter, IDS can be host-based or Network-based. In host-based systems, the system collects data from sources internal to the device, usually at the operating system level (various logs, etc.), monitors the execution of the device program, whereas the Network-based systems collect network packets, this is usually done by using network devices that are set to the promiscuous mode; hence, the network device operating captures all network traffic accessible to it, not only that addressed to it. Network-based systems also have nodes deployed at strategic locations, inspect network traffic, and monitor devices activities on the network.
- The detection pattern: According to this parameter, IDS may belong to two main categories: misuse detection (or signature detection) and anomaly detection. The former examines the activity of the entire infrastructure for patterns of misuses already available in the signature database, usually referred to as "attack identities," while the latter analyzes the behavior of the protected system over time to estimate what is considered normal (or legitimate) behavior. Any action that significantly deviates from that behavior is viewed as an attack or an intrusion.

An efficient IDS must detect intrusion with high accuracy but not confuse legitimate actions with intrusive ones. For anomaly-based intrusion detection systems, two performance metrics were usually examined, the Detection Rate (DR), which is defined as the ratio of the number of correctly detected attacks to the total number of attacks, and the False Alarm Rate (FAR), or false positive rate, which is the ratio of the number of normal connections that are wrongly classified as attacks to the total number of normal connections. An Efficient IDS maximizes the detection rate while maintaining the false alarm rate lower.

2.2 Benschmark of IoT-based IDS

Several proposals for IoT-based IDSs have been investigated;

For a better understanding of these systems, we examined different intrusion detection systems used in IoT environments, we investigated the associated detection methodologies, the treated threats, and advantages and disadvantages of each method. Table 1 summarizes this benchmark.

3. AN ARCHITECTURAL MODEL FOR IOT INTRUSION DETECTION SYSTEMS

Several IoT Based architectures have been introduced [17]. The literature is rich in general architectures of the intrusion detection systems and their constituents for the one to present the typical components of the intrusion detection system architecture.

Table 1: Benchmark of IDSs in IoT

Ref.	Detection Method	Treated threat	Advantage	Disadvantage
[8]	Hybrid -Signature and anomaly based	Spoofing, sinkhole, selective forwarding and information alteration	• Resource-constrained things, connected via IPv6 in 6LoWPAN networks • Proposition and implementation of a distributed mini-firewall to protect IP connected devices in 6LoWPAN networks; • Flexible and can be extended to detect more attacks	• Focusing only on routing attacks.
[9]	Signature based	DoS attacks for 6LoWPAN	• IDS runs on a host computer, it overcomes the resource constraint problems and provides more power to detect complicated attacks • False alarms reduction • IDS real-word applicable	• Limited to 6LoWPAN • Detected attacks depend on declared rules
[10]	Signature based	-	• Real-time detection • Less time consumed compared to traditional IDS • Low memory consumption • Takes into consideration massive data generated in IoT environments	• More CPU resources are consumed • The detection depends on the signature data base
[11]	Hybrid (trust and reputation strategy)	Sinkhole attack	• INTI Takes into consideration the impact of device mobility • Performance in detection rate, false positive rates and false negative rates.	• IDS depends on the trust and reputation estimation of the attacker model • Packets drop • Focuses mainly on sinkhole attacks on the rooting services
[12]	Specification based	Sinkhole attack	• Resource constraints challenge is taken into consideration in InDReS • Low average energy consumption • Low packet drop ratio • Instant network response against detected attacks • improvement on many QoS metrics over the existing INTI scheme	• Cannot detect unknown attacks
[13]	Hybrid (signature and anomaly based)	Jam attack, false attack and reply attack	• Heterogeneity of IoT networks is taken into consideration • Resource constraints challenge is taken into consideration • Low false positive rate	• Based on the assumption that the complexity of the iot system is “normal” • Depends on the accuracy of the normal and the abnormal the action libraries • DoS attack can affect the solution

Ref.	Detection Method	Treated threat	Advantage	Disadvantage
[14]	Hybrid (signature and anomaly based)	DoS, routing and network attacks	• Real-time detection • Lightweight in terms of CPU and RAM requirements • first self-adapting, knowledge-driven IDS for IoT • Different IoT communication protocols and applications are taken into consideration • Deployable on border router or as a standalone tool • Provides a knowledge sharing mechanism that enables collaborative incident detection, • Can act as data source for multisource security information management (SIEM) systems	• Scalability is sensitive to number of IDS nodes.
[15]	Signature based	Attacks against 6LoWPAN and CoAP, as well as DoS	• Cross-layered attack detection • Detection of external (internet) and internal attacks (sensors end devices) • Quick reaction and attackers blocking • The system can be extended and configured • Interoperability between communication technologies	• Sensitive to declared rules
[16]	Anomaly Based	DDoS	• Magnitude of DDoS attack is taken into consideration • Prevention and Action module could isolate malicious nodes	• Scalability sensitive to the number of nodes • Cannot detect unknown attacks

Every IDS system would include at least the typical components:

- Audit collection:

Audit data helps the system make decisions. The monitored system feeds this component by device logs data, application logs data and network traffic data, etc.

- Audit storage:

Audit storage is the space for storing audit data, either indefinitely or temporarily awaiting processing.

- Processing unit:

The processing unit is the kernel of the intrusion detection architecture; at this level, the algorithms are run to identify suspicious behaviors or patterns in the audit data.

- Configuration data:

Configuration data are the parameters of control and tuning of the intrusion system. At this level, we can define what audit data should be collected and the way to respond to intrusions.

- Reference data:

The reference data stores information about known intrusion signatures—for misuse systems — or normal behavior profiles —for anomaly systems—. In the latter case, the processing block updates the profiles as new patterns about the observed behavior become available. This update is often carried out regularly. Stored intrusion signatures are updated by the Security Expert as and when new intrusion signatures become well known.

- Active/ processing data:

It is a space in which the processing unit must regularly store intermediate treatments results.

- Alarm unit:

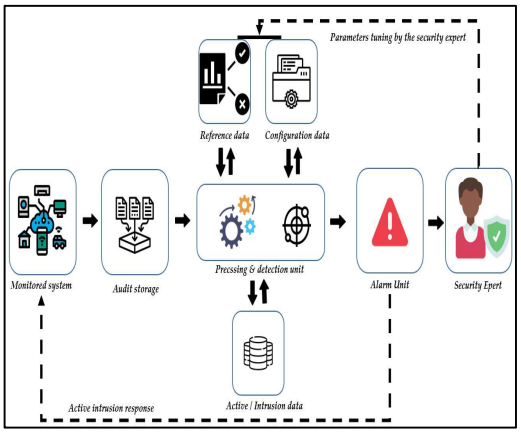

Figure 1: Organization o a typical IoT-based Intrusion detection system

This component allows managing the system's output, whether it is an automatic response to suspicious activity or the notification of the Security Expert. Fig.1; presents the organizational architecture and the relationship between the typical components.

4. THE LIMITATION OF THE INTRUSION DETECTION SYSTEMS AND THE SOLUTIONS

As discussed, IDSs belong to two main categories: signature-based detection (or "misuse detection") and anomaly detection. The method used for signature-based detection is effective, widely used by tools like Snort [18] or Suricata [19], but it can only detect known attacks available in the dataset. On the other hand, the method used by anomaly detection-based IDS can detect unknown attacks but often generates a high number of false alarms [20].

Recently, research has focused on improving anomaly-based IDSs thanks to their ability to detect unknown attacks. Machine learning techniques have been proposed for both misuse and anomaly detection. These techniques provide the ability to learn from the data without being explicitly programmed. However, anomaly detection algorithms are less deployed in practice despite these advantages, and misuse detection still dominates.

Supervised Machine learning-based solutions for improving detection quality in anomaly-based IDS have been investigated [21, 22]. However, lowering the false alarm rate and improving the detection accuracy in general on known datasets isn't enough to achieve this goal, because the results aren't transferred to real-world networks and the system must be re- trained on the monitored network, which is difficult to do because it necessitates labeled datasets containing real-world attacks.

It is more suitable to focus on unsupervised machine learning techniques that can learn new attacks from different traffic datasets without labels.

5. PARTICLE SYSTEMS AND PARTICLE SWARM OPTIMIZATION

5.1 History of particle systems

The term particle systems was coined by William T. Reeves in 1983; he published his paper "Particle Systems - A Technique for Modeling a Class of Fuzzy Objects"[23]; Reeves explained how he invented the particle system paradigm for computer graphics, especially for use in his film Star Trek II: The Wrath of Khan.

Based on applying basic fundamental newton's laws to a virtual collection of particles on a computer, he created interesting graphics presenting fuzzy objects [23]. Computer graphics consisted of shapes created mainly by polygons and edges until this invention. Reeves' vision of particle systems enabled the creation of objects that did not have sharp edges. This new paradigm allowed the representation of much more complex effects such as the snow, the rain, the fire, the clouds and the swarm of bees, etc. (Figure 2 illustrates an image of a forest).

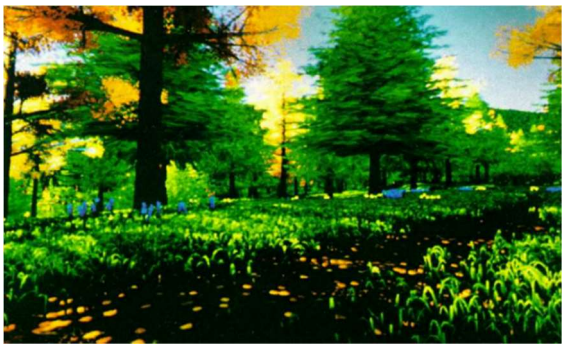

Figure 2 : Image Of A Forest Through The Particle System By T.Reeves

Recent years have seen a growing interest in complex systems, i.e., systems of elements that interact nonlinearly and are difficult to model or understand at a global scale based solely on understanding individual elements. Many systems in nature are of this type; also, an increasing number of

applications in chemistry [24], engineering and social networks resemble such systems given the large number of elements and the complex interactions between the components.

Complex systems can differ in their nature. However, they share one common aspect: the behavior of the system is difficult to understand, no matter how simple the behavior of its constituents, although mathematical models exist for the analysis of such systems, the design of intelligent complex systems remains an elusive problem, this is linked essentially to the difficulty of deducing the impacts of the elements minor variations and how they would influence the structure of the system as a whole, and also to the lack of a single and a central point of control.

5.2 Particle Swarm Optimisation

Beyond computer graphics, the concept of particle systems has found many applications. Among these applications, the particle swarm optimization algorithm (PSO). This algorithm [25] extends particle systems to problem solving, and more specifically to high-dimensional spaces based on a social model of interactions between agents.

Particle swarm optimization is an optimization algorithm based on swarm intelligence. Particle Swarm Optimization (PSO) seeks inspiration in the coordinated dynamics of groups of animals [25].

Collective behaviors such as flocks stem from applying specific local rules that are generic and independent of a particular time or location. Reynolds; indicates that the shape of the entire flock is a result of the individual behavior of birds, which follow three primary rules:

- Separation: it allows the birds to avoid their neighbors by adjusting their physical position;
- Alignment: it allows the birds lining up with agents close by;
- Cohesion: it allows individuals to move toward the average position of local flock mates.

To form a swarm, each bird has a position. A velocity vector, and has some awareness of everything happening around it in some vicinity but has little visibility outside that vicinity. In terms of the particle system model, the PSO algorithm maintains a population of particles (the swarm). Each one is defined by its location in a multidimensional search space. Every particle represents a potential solution to the optimization problem at hand. The particles can start at random locations and explore the search space, looking for an objective function's minimum (or maximum). Over time, and after a series of explorations and calculations of good positions in the search space, the particles converge together over a specific optimum or several optima. Fig.3 illustrates the convergence of the swarm. There are many parameters in PSO which affect the quality of the solution, such as the swarm size, the swarm communication topology, the swarm initialization pattern and the fitness function used, etc [38].

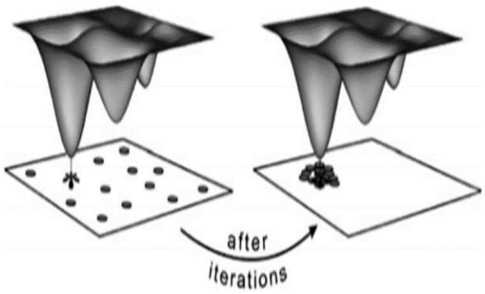

Figure 3: The convergence of the swarm over optima

The particle movement is computed as follows [25]:

$$xi(t+1) = xi(t)+vi(t) \quad (1)$$

$$vi(t+1)=wvi(t)+c1r1(pbesti(t)-xi(t))+c2r2(gbest(t)- xi(t)) \quad (2)$$

xi(t) is the position of particle i at time t, vi(t) is the velocity of particle i at time t, pbesti(t) is the best position found by particle i so far, gbest(t) is the best position found by the swarm so far, w is an inertia factor with values between 0 and 1, c1 is the cognitive parameter and c2 is the social parameter. r1 and r2 are random variables between 0 and 1. Fig.4, presents the flow chart of the general particle swarm optimization algorithm.

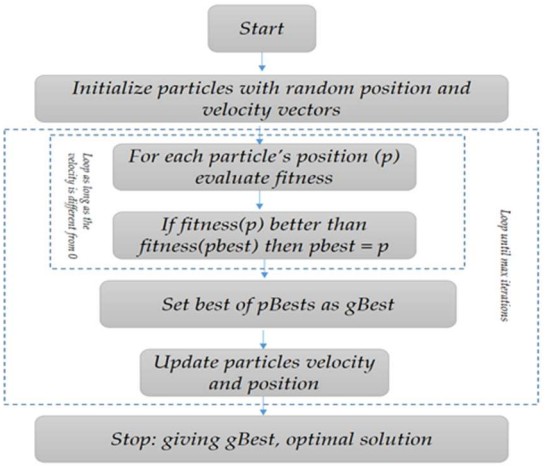

Figure 4: Flow Chart Of The General Particle Swarm Optimization Algorithm

6. BUIDLING AN INTRUSION DETECTION CLASSIFIER USING THE PARTICLE SWARM CLUSTERING APPROACH

The intrusion detection problem can be reduced from the machined learning perspective to a classification or a clustering problem [26, 27]. An intrusion detection system based on clustering must follow these steps:

1. Modeling the normal behaviors of the network by creating clusters from unlabeled training datasets;
2. Labeling clusters as 'normal' or 'anomalous';
3. Using the labeled clusters to classify network data.

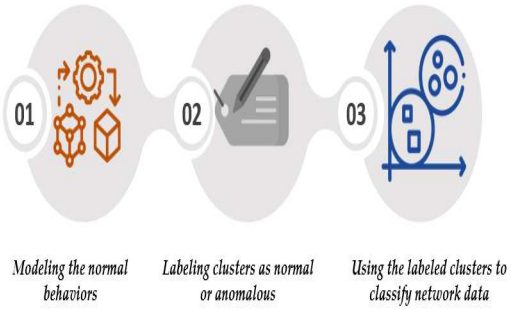

Fig.5; The Steps To Build An Intrusion Detection Classifier

Before explaining the intrusion detection process, it is important discussing the datasets used in this problem. The dataset is a critical component for building machine learning-based IDSs since they require representative knowledge of the IoT environment. The dataset quality affects the performance of the IDS system. However, IoT datasets are scarce, especially labeled ones; this is due mainly to the labelling process itself. Most of the time, it is hard to label, especially when experts cannot determine whether the traffic is an attack or not. A number of IoT and non-IoT datasets have been used to address the intrusion detection problem for IoT systems. Table 2, lists some of the most used datasets for training and testing machine learning-based IDSs for IoT.

Table 2. Common Datasets Used For Iot-Based Idss

Common IoT Dataset	KDD99
	NSL-KDD
	CICandMal2017
	BOT-IoT
	IoTID20

Published studies show that KDD99 is the dataset used by large in IDS and machine learning areas; however, it is very large and contains many redundant records. NSL-KDD was introduced as the duplicates removed and size-reduced version of the KDD99 dataset.

Android malware dataset (CICAndMal2017) proposed by Shiravi, A.; [29], includes more than 80 attributes and regroups malware and benign applications. The malware samples used to build the dataset are Adware, Ransomware, Scareware, and Short Message Service (SMS) malware.

The Bot-IoT dataset [30] contains over 72,000,000 records related to IoT traffic, including DDoS, DoS, OS and Service Scan, Key-logging, and data exfiltration attacks. Unlike other datasets, the Bot-IoT is dedicated to validating IDS in an IoT environment. The Botnet dataset is an internet-connected devices-based dataset that contains training and test data for 7 and 16 different types of botnet attacks. The botnet dataset comprises four types of data: byte-, packet-, time-, and behavior-based.

IoTID20 was created to detect unusual activity in the IoT ecosystem [31]. Laptops, smartphones, Wi-Fi cameras, and other IoT devices were used to create it. Intel Lab Dataset goes back to 2004, it was created, by collecting data from 54 sensors deployed in the Intel Berkeley Research lab. It was collected using the TinyDB in-network query processing system, built on the TinyOS platform.

Intel lab, is not considered an IoT dataset, to make it so, researchers had to manually add some abnormal records to it to simulate attacks.

Prior to any model design, data preprocessing is required. This includes feature selection and data cleaning. The preprocessing operation involves reducing the dataset's dimensionality by removing irrelevant data, which can benefit the model by reducing overfitting, improving accuracy, and shortening the training time.

6.1 Modeling The Normal Behavior

This phase consists of two stages, setting the original clusters and optimizing them using PSC.

6.1.1 Setting the initial clusters:

To achieve this end, there are two encoding methods [32]:

- Particle based encoding: This encoding makes each particle a vector of I integers, where the jth element represents the cluster label assigned to element j, j ∈ {1, ..., I} and I is the number of data elements to regroup;
- Centroid based encoding: On the other hand, in the centroid-based encoding [35, 36], the position of each particle p defines a set of potential cluster centroids (Fig. 7) in a d-dimensional data space: xp = {mp,1, . . ., mp,j , . . . , mp,K}, where mp,j represents the jth cluster centroid and K is the number of clusters. In the same manner as K-means, PSC with the centroid-based encoding divides the dataspace into Voronoi cells (represented in Fig.6). This encoding is preferred since it allows a simple management of centroids even in large dimensions of the search space.

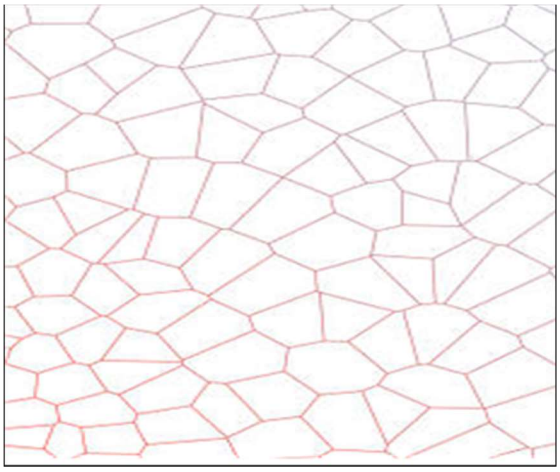

Fig. 6; Voronoi Cells

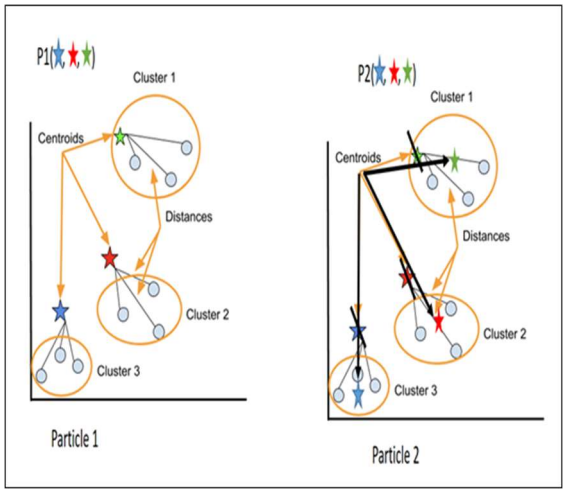

Fig. 7; Particle Based Encoding scheme

6.1.2 Optimizing clusters

Using the PSO algorithm for clustering is commonly referred to as Particle Swarm Clustering (PSC). As in other PSO applications, each particle represents a solution to the problem.

In fact, the clusters obtained can be evaluated with three types of Cluster Validity Indices: external, internal, and relative.

External CVIs compare clustering results with information known a priori (labels). As PSC is used where labels are not available, external CVIs cannot be used in that case. On the other hand, internal CVIs evaluate clusters solely based on the data. Commonly used measures are clusters compactness and separation. Relative CVIs can be used to compare two clustering results and to indicate which one is better. While most external CVIs are relative, the term usually refers to internal relative CVIs. Most novel PSC techniques or applications (e.g. [33, 34])) simply adopt a certain relative CVI as their fitness function. Still, in general, any PSC technique can be easily modified to use any CVI as its fitness function.

The standardized version of the centroids based encoding PSC was presented for the first time in [37]. Ballardini, has implicitly used this encoding method in his tutorial on Particle Swarm Optimization clustering.

For the sake of presenting it here in the paper, we consider the following symbols:

- N_d *denotes the input dimension, i.e. the number of parameters of each data vector*
- N_0denotes the number of data vectors to be clustered
- N_c, denotes the number of cluster centroids (as provided by the user), i.e. the number of clusters to be formed
- Z_p denotes the p-th data vector

- m_j denotes the centroid vector of cluster j
- nj, is the number of data vectors in cluster j
- Cj, is the subset of data vectors that form cluster;

The particle is constructed as follow:

$$x_i = (m_{i_1}, m_{i2}, m_{ij}, \ldots . m_{i,Nc})$$

where m_{ij} represents the jth vector of the centroids of the ith particle in the cluster Cij. The swarm represents a configuration of candidate centroids for the current data vectors. The PSC algorithm is presented as follows:

1. Initialize each particle to contain N_c, randomly selected cluster centroids
2. Fort = 1 to t_{max} do
   a. For each particle i do
   b. For each data vector z_p
      i. Calculate the Euclidean distance d(Z_p, m_{ij}) to all cluster centroids C_{ij}
      ii. Assign Z_p to cluster C_{ij} such that d(Z_p, m_{ij})=min$_{\forall c=1,..N}$ $\{d(z_p, m_{ic})\}$
      iii. Calculate the fitness using equation (3)
   c. Update the global best and local best positions
   d. Update the cluster centroids using equation (3) and (4)
3. Where t_{max} is the maximum number of iterations.

the fitness function of the particles is evaluated by the quantification error:

$$J_e = \frac{\sum_{j=1}^{N_c} \left[\sum_{\forall Z_p \in C_{ij}}^{c} d(z_p, m_j)/|C_{ij}|\right]}{N_c} \quad (3)$$

Where d is defined in equation (4), and $|C_{ij}|$ is the number of data vectors belonging to cluster C_{ij}.

$$d(z_p, m_j) = \sqrt{\sum_{k=1}^{N_d} (z_{pk} - m_{jk})^2} \quad (4)$$

6.1.3 Labeling clusters

The labeling of the clusters is based on two assumptions [26]:

- The first assumption is that the number of normal traffic records is more significant than abnormal records;
- The second assumption is that the abnormal traffic features are different from the features of the normal traffic;

In other terms, instances that appear in small clusters are labeled as anomalies since that the number of normal instances largely exceeds the number of intrusions according to assumption 1, meaning that normal instances should form large clusters compared to intrusions; and according to assumption 2 intrusions and normal instances should not fall in the same cluster;

Also, an additional observation is that many attacks have closely similar patterns with minor differences in the values of the features. For example, smurf is one of the popular denial of service attacks, and the records belonging to this class have 1032 for src_bytes, 0 for dst_bytes, and many other features are also exactly the same. Since the denial of service attack generates a great number of packets to overwhelm the target resources there would be a lot of repetitive packets that are sent to the target, which causes a lot of data points with the identical feature values. If this is the case, a cluster which contain such data points may be very dense. With this observation, it is possible to extend the assumptions above that characterizes anomaly clusters meaning that cluster with a very low density will be regarded as anomalous. The new process of obtaining the estimated label information is thus as follows:

- If a cluster is extremely dense, label it as anomalous;
- If a cluster is small or sparse, label it as anomalous;
- Otherwise, label it as normal.

6.1.4 Using the labeled clusters to classify network data

Once the clusters are created from a training data set, the system is ready to detect intrusions. we can affect each data item to the closest cluster (we can use the centroid only as it represents the cluster) under the distance metric used to calculate the distance.

7. CONCLUSION

There has been a shortage of researches that approached Intrusion Detection Systems in IoT using machine learning and particle systems combination. This paper, which extends previous work [38], tries to fill the gap by adding another approach for addressing the IDS systems in the IoT environment to the already existing methods being developed in this field. The key contribution of our work is to review this combination and present the ways to build an intrusion detection classifier using the particle swarm clustering approach. Several aspects have been examined; starting from reviewing the IoT-based IDSs; we presented a general architectural model for IoT-based IDSs. We discussed the particle swarm clustering approach in intrusion detection, the input datasets, the encoding techniques to set the initial clusters, the underlying assumptions for labeling them, and how to use them to classify the network data. Our future work will focus on applying the method to a real IoT-based network to evaluate the approach's efficacy. We also plan to focus on some specific networks of smart IoT devices such as wearable techs to better understand the patterns of attacks and the defense mechanisms to apply.

REFERENCES:

[1] M. El Bekri, O. Diouri and A. Tioutiou, "Towards a classification of things," 2016 SAI Computing Conference (SAI), 2016, pp. 1243-1246, doi: 10.1109/SAI.2016.7556138

[2] J. Margolis, T. T. Oh, S. Jadhav, Y. H. Kim and J. N. Kim, "An In-Depth Analysis of the Mirai Botnet," 2017 International Conference on Software Security and Assurance (ICSSA), 2017, pp. 6-12, doi: 10.1109/ICSSA.2017.12.

[3] A. Aris and S. F. Oktug. Poster: State of the Art IDS Design for IoT. In Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks, EWSN '17, pages 196–197, USA, February 2017. Junction Publishing.

[4] Al-Imran, M., Ripon, S.H. Network Intrusion Detection: An Analytical Assessment Using Deep Learning and State-of-the-Art Machine Learning Models. Int J Comput Intell Syst 14, 200 (2021). https://doi.org/10.1007/s44196-021-00047-4

[5] Hua Yang, Tao Li, Xinlei Hu, Feng Wang, Yang Zou, "A Survey of Artificial Immune System Based Intrusion Detection", The Scientific World Journal, vol. 2014, Article ID 156790, 11 pages, 2014. https://doi.org/10.1155/2014/156790

[6] Zhiyong Wang, Shengwei Xu, Guoai Xu, Yongfeng Yin, Miao Zhang, Dawei Sun, "Game Theoretical Method for Anomaly-Based Intrusion Detection", Security and Communication Networks, vol. 2020, Article ID 8824163, 10 pages, 2020. https://doi.org/10.1155/2020/8824163

[7] E. Vishnu Balan, M.K. Priyan, C. Gokulnath, G. Usha Devi, Fuzzy Based Intrusion Detection Systems in MANET, Procedia Computer Science, Volume 50,2015,Pages 109-114, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.04.071.

[8] Shahid Raza, Linus Wallgren, Thiemo Voigt,SVELTE: Real-time intrusion detection in the Internet of Things,Ad Hoc Networks,Volume 11, Issue 8,2013,Pages 2661-2674,ISSN 1570-8705, https://doi.org/10.1016/j.adhoc.2013.04.014..

[9] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits. Denial-ofService detection in 6lowpan based internet of things. In IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications, pages pp. 600–607, 2013

[10] C. Jun and C. Chi. Design of Complex Event-Processing IDS in Internet of Things. In 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation, pages 226–229, January 2014

[11] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos. Detection of sinkhole attacks for supporting secure routing on 6lowpan for Internet of Things. In 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pages 606–611, May 2015.

[12] M. Surendar and A. Umamakeswari. InDReS: An Intrusion Detection and response system for Internet of Things with 6lowpan. In 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pages 1903–1908, March 2016.

[13] Y. Fu, Z. Yan, J. Cao, O. Koné, and X. Cao. An Automata Based Intrusion Detection Method for Internet of Things, May 2017.

[14] [MRMB17] D. Midi, A. Rullo, A. Mudgerikar, and E. Bertino. Kalis A System for Knowledge Driven Adaptable Intrusion Detection for the

Internet of Things. In 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pages 656–666, June 2017.

[15] Jorge Granjal and Artur Pedroso. An Intrusion Detection and Prevention Framework for Internet-Integrated CoAP WSN, 2018. ISSN: 1939-0114 Library Catalog: www.hindawi.com Pages: e1753897 Publisher: Hindawi Volume: 2018.

[16] A. Aldaej. Enhancing Cyber Security in Modern Internet of things (IoT) Using Intrusion Prevention Algorithm for IoT (IPAI). IEEE Access, pages 1–1, 2019. Conference Name: IEEE Access.

[17] Mohamed, Tamara & Aydin, Sezgin. (2021). IoT-Based Intrusion Detection Systems: A Review. Smart Science. 1-18. 10.1080/23080477.2021.1972914.

[18] M. Roesch, "Snort - lightweight intrusion detection for networks," in Proceedings of the 13th USENIX Conference on System Administration, LISA '99, (USA), p. 229–238, USENIX Association, 1999.

[19] https://suricata.readthedocs.io/en/latest/what-is-suricata.html

[20] Martin Grill, Tomáš Pevný, Martin Rehak, Reducing false positives of network anomaly detection by local adaptive multivariate smoothing, Journal of Computer and System Sciences, Volume 83, Issue 1, 2017, Pages 43-57, ISSN 0022-0000, https://doi.org/10.1016/j.jcss.2016.03.007.

[21] Aboueata, Nada & Alrasbi, Sara & Erbad, Aiman & Kassler, Andreas & Bh, Deval. (2019). Supervised Machine Learning Techniques for Efficient Network Intrusion Detection. 18.10.1109/ICCCN.2019.8847179.

[22] Liu, H.; Lang, B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. Appl. Sci. 2019, 9, 4396. https://doi.org/10.3390/app9204396

[23] WILLIAM T. REEVES, Particle Systems a Technique for Modeling a Class of Fuzzy Objects, Volume 17, Number 3, July 1983;

[24] M. Afkhami, A. Hassanpour, M. Fairweather, Effect of Reynolds number on particle interaction and agglomeration in turbulent channel flow, Powder Technology,Volume 343,2019,Pages 908-920,ISSN 00325910, https://doi.org/10.1016/j.powtec.2018.11.041.

[25] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39-43, doi: 10.1109/MHS.1995.494215.

[26] Portnoy, Leonid & Eskin, Eleazar & Stolfo, Salvatore. (2001). Intrusion Detection with Unlabeled Data Using Clustering.

[27] Bohara, Binita et al. "A SURVEY ON THE USE OF DATA CLUSTERING FOR INTRUSION DETECTION SYSTEM IN CYBERSECURITY." International journal of network security & its applications vol. 12,1 (2020): 1-18. doi:10.5121/ijnsa.2020.12101

[28] Serkan Kiranyaz, Jenni Pulkkinen, Moncef Gabbouj, Multi-dimensional particle swarm optimization in dynamic environments, Expert Systems with Applications, Volume 38, Issue 3, 2011,

[29] Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput. Secur. 2012, 31, 357–374. [CrossRef]

[30] Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, Benjamin Turnbull (2018). Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT. arXiv:1811.00701.

[31] Ullah, Imtiaz & Mahmoud, Qusay. (2020). A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks. 508-520. 10.1007/978-3-030-47358-7_52.

[32] Jenni Raitoharju, Kaveh Samiee, Serkan Kiranyaz, Moncef Gabbouj, Particle swarm clustering fitness evaluation with computational centroids, Swarm and Evolutionary Computation, Volume 34, 2017, Pages 103-118, ISSN 2210-6502, https://doi.org/10.1016/j.swevo.2017.01.003.

[33] K. Govindarajan, D. Boulanger, V. S. Kumar, and Kinshuk, "Parallel particle swarm optimization (ppso) clustering for learning analytics," in Proc. of IEEE Int. Conf. on Big Data, pp. 1461–1465, Oct 2015.

[34] C. Vimalarani, R. Subramanian, and S. N. Sivanandam, "An enhanced PSO-based clustering energy optimization algorithm for wireless sensor network," The Scientific Worl

[35] Hongying Zheng, Meiju Hou, Yu Wang, An Efficient Hybrid Clustering-PSO Algorithm for Anomaly Intrusion Detection, JOURNAL OF

SOFTWARE, VOL. 6, NO. 12, DECEMBER 2011

[36] D. W. van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," The 2003 Congress on Evolutionary Computation, 2003. CEC '03., 2003, pp. 215-220 Vol.1, doi: 10.1109/CEC.2003.1299577

[37] Augusto Luis Ballardini, A tutorial on Particle Swarm Optimization Clustering, CoRR, abs/1809.01942, 2018, http://arxiv.org/abs/1809.01942

[38] Mohamed El Bekri, Ouafaa Diouri, Pso Based Intrusion Detection: A Pre-Implementation Discussion, Procedia Computer Science, Volume 160,2019,Pages 837-842, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2019.11.002.