Dynamic Inertia Weight Particle Swarm Optimization for Anomaly Detection: A Case of Precision Irrigation

Mohamed EL Bekri[1*], Ouafaa Diouri[2] and Dalila Chiadmi[3]

[1*] Phd Student, Department of Computer Engineering - Mohammed V University of Rabat, Morocco. mohamed.elbekri@gmail.com, Orcid : https://orcid.org/0000-0003-2133-4707

[2]Professor, Department of Computer Engineering - Mohammed V University of Rabat, Morocco. diouri@emi.ac.ma, Orcid: https://orcid.org/0000-0003-2407-1615

[3]Professor, Department of Computer Engineering - Mohammed V University of Rabat, Morocco. chiadmi@emi.ac.ma, Orcid: https://orcid.org/0000-0002-5709-5675

Received: March 02, 2023; Accepted: April 08, 2023; Published: May 30, 2023

Abstract

Anomaly-based Intrusion Detection System (IDS) is a type of IDS that detects abnormal behaviors by analyzing system activity and network traffic. Anomaly-based IDS works by establishing a baseline of normal behavior for a system or a network. However, these types of systems are less used compared to signature-based IDS for one primary challenge: How to define this normal behavior baseline? The answer to this question is complicated, since it involves not only analyzing or learning from historical data, but requires an understanding of the business domain the system is implemented in. The present study proposes a novel approach to constructing an unsupervised data classifier that combines both Particle Swarm Optimization (PSO) and clustering techniques for anomaly detection. The primary objective of this methodology is to surmount the limitations that conventional clustering algorithms suffer from, such as their inability to identify non-linear patterns within the data, susceptibility to initial conditions, and difficulty in overcoming the problem of local optima. The concept of particle systems is discussed by examining their origins, search strategies, and convergence mechanisms. We use a variant of the Particle Swarm Optimization called Dynamic Inertia Weight-Particle Swarm optimization (DIW-PSO) for our clustering process, and we elaborate on the reasoning behind this decision. Subsequently, we describe the labeling algorithm used for the resulting clusters and we explain the process for identifying anomalous clusters. We have demonstrated the effectiveness of our method by applying it to an intelligent irrigation control system for cotton plants. The results show that our classifier was able to accurately detect abnormal patterns that deviated from the optimal water requirements and growth conditions of the plants.

Keywords: Particle Swarm Optimization, Intrusion Detection, Machine Learning, Clustering, Precise Irrigation.

1 Introduction

Throughout history, individuals from various communities around the world have utilized birds to help them explore their surrounding space and forecast weather. Nomadic tribes, for example, who used to

Journal of Internet Services and Information Security (JISIS), volume: 13, number: 2 (May), pp. 157-176
DOI: *10.58346/JISIS.2023.I2.010*

*Corresponding author: Phd Student, Department of Computer Engineering - Mohammed V University of Rabat, Morocco.

prospect new places to settle in, used the presence of birds as an indication of a nearby spring, because most of the times, birds fly around areas with abundant supply of water. Also, people in the past, used to observe the behaviors of birds in order to forecast weather, for instance, if birds are observed flying high in the sky, this typically indicates fair weather, on the other hand, if they are flying low this serves as a warning of inclement weather to come. If a storm is approaching, birds will stop flying and seek refuge at coast. It is obvious that our ancestors were interpreting birds' behaviors to predict specific patterns in space and time (Wyndham, F. S., Park, K. E. (2018)).

We believe that two crucial strategies underlie theses advanced sensory and adaptive capabilities of birds:

The First strategy is flying in a flock: this allows the flock to collectively have a much wider field of vision than any individual bird could achieve on its own. Also, birds flying in a flock can take advantage of the air currents created by the birds in front of them. By flying slightly behind and to the side of the bird in front, each bird can benefit from reduced air resistance and therefore use less energy to maintain its speed and altitude. This allows the birds to fly for longer distances without becoming fatigued, which in turn allows them to cover more ground and spot potential food sources, atmospheric depressions and predators from a greater distance.

The second strategy is pragmatic foraging, birds select food on a benefit to coast basis (Moermond, T. C. (1990)), which means, if the energy required to obtain the food exceeds the energy gained from consuming it, and then it is not a worthwhile option for the birds. By selecting the most energy-efficient food sites, birds can optimize their energy expenditure while foraging.

These birds' mechanisms surpass avian biology, and can be extended to address a broad spectrum of problems including data patterns exploration and analysis.

In this work we use a variant of Particles Swarm optimization (PSO) called Dynamic Inertia weight PSO (DIW-PSO) algorithm which enforces the energy optimization strategy while exploring optimal clusters in data. In fact, the algorithm begins by emphasizing global searching, taking advantage of the birds' initial burst of energy. As the number of iterations decreases, the algorithm gradually shifts towards prioritizing local searching. The aim is to build an anomaly classifier that groups instances of data into clusters based on their similarities. Each cluster represents a group of similar behaviors or patterns, and by analyzing each cluster, the classifier distinguishes normal instances from anomalous ones.

Although various approaches for anomaly detection rely on observing the data traffic in the system to identify targeted attacks or some specific suspicious behaviors, this article presents an alternative method for protecting the system. Our approach involves monitoring the behavior and performance of the entire system, including its components, such as devices, routers, and sensors. This approach offers a comprehensive view of the network performance and helps to identify a wide range of potential vulnerabilities.

In terms of building the classifier, many clustering methods, including FCM and K-MEANS and even the standard version of PSO, are commonly used in anomaly detection. However, only a few clustering algorithms can guarantee a global optimal solution. Therefore, in order to find global optimal clusters, we investigated the efficacy of inertia weight in enhancing the quality of clusters. We use a variant of the Particle Swarm Optimization (PSO) algorithm, which we have coined DIW-PSO.

At the beginning, the paper provides an overview of various Intrusion Detection Systems (IDS) along with their respective detection techniques, benefits, and limitations, in addition to highlighting their essential components. Subsequently, the paper elucidates on the origins and underlying dynamics of particle systems.

A dedicated section is then allocated to discuss the Inertia Weight parameter and its potential impact on enhancing clustering outcomes. The paper then proceeds to explicate the proposed methodology for anomaly detection, encompassing the initiation and labeling of clusters, as well as the mechanism for anomaly detection. Finally, the proposed approach is applied to a practical scenario of anomaly detection in an intelligent irrigation control system for cotton plants.

The results revealed the classifier's capability to identify the abnormal patterns, which are not consistent with the plant's growth conditions and water requirements.

2 Review of IDS Systems

An IDS (Intrusion Detection System) is a system that monitors network data to identify potential malicious activity and generate alerts in response. Some IDS systems also have the ability to take action, such as blocking traffic from suspicious IP or MAC addresses when malicious activity or anomalous traffic is detected. The primary functions of an IDS are anomaly detection and reporting, but it can also perform additional tasks to protect against intrusions.

Types of IDSs

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

1. The scope of protection: This parameter refers to the nature of data the IDS collects and monitors in order to detect anomalies. There are two main types of IDS based on this parameter: host-based and network-based. Host-based IDS systems collect data from sources within the device, such as the device logs, and monitor the execution of programs on the device. These systems are useful for detecting intrusions at the device level, but may not be as effective at the network-level. Network-based IDS systems, on the other hand, collect network packets by using network devices that are set to promiscuous mode, which allows them to capture all traffic on the network. These systems may also have nodes deployed at strategic locations on the network to inspect traffic and monitor activity. Network-based IDS systems are useful for detecting network-level threats, but may not provide as much protection at the device level as host-based systems.
2. The detection pattern: There are two main types of detection patterns: misuse detection (also known as signature detection) and anomaly detection. Misuse detection systems examine the activity of the entire infrastructure for patterns of misuse that are already stored in a signature database, often referred to as attack identities. These systems are able to detect known attacks and intrusions based on their specific patterns of behavior. Anomaly detection systems, on the other hand, analyze the behavior of the protected system over time to determine what is considered normal or legitimate behavior. Any action that significantly deviates from this baseline is viewed as an attack or intrusion. These systems are able to detect unknown attacks and intrusions by identifying unusual behavior.

An effective IDS should be able to accurately detect intrusions without mistaking legitimate actions for malicious ones. In the case of anomaly-based intrusion detection systems, two performance metrics are commonly used to evaluate the system's efficiency: the detection rate (DR) and the false alarm rate (FAR). The detection rate is the proportion of correctly detected attacks to the total number of attacks, while the false alarm rate is the proportion of normal connections that are incorrectly classified as attacks to the total number of normal connections. To be efficient, an IDS should maximize the detection rate while keeping the false alarm rate low.

Challenges and Proposed Solutions for IDSs

As discussed, intrusion detection systems (IDS) can be divided into two main categories: signature-based detection (also known as "misuse detection") and anomaly detection. Signature-based detection is effective at detecting known attacks and is used by tools such as Snort (Roesch, M. (1999)) and Suricata. However, it can only detect attacks that are already included in its database. On the other hand, anomaly detection-based IDS can detect unknown attacks, but it often generates a high number of false alarms (Grill, M., Pevný, T., Rehak, M. (2017)).

Recently, there has been a focus on improving anomaly-based intrusion detection systems (IDSs) due to their ability to detect unknown attacks. Machine learning techniques, have been proposed as a means to improve these systems. These techniques allow the system to learn from data without requiring explicit programming. However, despite the benefits of using anomaly detection algorithms, misuse detection is still more commonly used in practice (Giorgi, G., 2022).

There have been efforts to use supervised machine learning techniques to improve the performance of anomaly-based IDSs (Aboueata, N.(2019; Liu, H., Lang, B. (2019)), but these approaches have shown limitations as well, because IDS improve false alarm rate and detection accuracy only on training datasets, which is not sufficient to effectively detect attacks in real-world scenarios, the system must be re-trained on the specific network to be monitored, which requires labeled datasets containing real-world attacks, which can be difficult to obtain. Instead, it may be more effective to focus on unsupervised machine learning techniques that can learn and adapt without the need for labels. This paper is a contribution in that respect.

3 Related Works

A number of different approaches have been proposed for building unsupervised anomaly detection classifiers (Al-Imran, M., Ripon, S. H. (2021)). One popular approach is to use statistical methods, such as the Mahalanobis distance or kernel density estimation, to calculate a score for each data point that indicates how anomalous it is relative to the rest of the dataset. Alternatively, some researchers have explored the use of deep learning methods, such as auto-encoders or variational auto-encoders, to learn a compressed representation of the data and then identify points that are poorly reconstructed as anomalous.

A study by Chalapathy, R. (2019) compared the performance of several deep learning-based methods on a range of datasets and found that autoencoders and variational autoencoders tended to perform well, but again noted that the optimal approach varied depending on the specifics of the dataset.

When it comes to clustering algotthms, Overall, the choice of an approach for unsupervised anomaly detection will depend on the specific requirements of the task at hand, such as the size and complexity of the dataset, the desired level of sensitivity and specificity, a careful evaluation of the performance of different approaches on relevant benchmark datasets is recommended to determine the most effective approach for a given task.

4 A Retrospective Examination of Particle Systems and Particle Swarm Optimization History of Particle Systems

The term "particle systems" was coined for the first time by William T. Reeves in his 1983 paper "Particle Systems - A Technique for Modeling a Class of Fuzzy Objects" Reeves, W. T. (1983). In his publication, Reeves explained how he developed the concept of particle systems for use in computer graphics, particularly in the production of the film Star Trek II: The Wrath of Khan. William T. Reeves created visually appealing graphics depicting fuzzy objects by utilizing autonomous entities known as "particles" that move based on the local effects they have on each other. Before the invention of particle systems, computer graphics were mainly created using polygons and edges. The introduction of particle systems allowed the creation of objects with softer, more natural edges and enabled the representation of a wide range of complex effects, such as snow, rain, fire, clouds, and swarms of bees. This new paradigm expanded the capabilities of computer graphics and allowed the creation of more realistic and dynamic visual effects.

The success of the particle systems has gone beyond computer graphics and they have been used to model group movements of ant and bee colonies, biological cells, robot team movement etc. what's interesting is that, even though the movement of each particle is predetermined, the interactions between these particles in a population can result in a very sophisticated collective behavior. This phenomenon is referred to as "self-organization" and enables the entire particle system to move as if it were a single entity, composed of separate individuals. Following all these successes, researchers extended the use of particle systems to other optimization problems such as data mining and machine learning, and Particle Swarm Optimization algorithm we will present below, is a concrete example of this extension.

Particle Swarm Optimization

PSO, has been introduced for the first time by Eberhart and Kennedy in 1995, this algorithm utilizes the principles of particle systems to solve problems, particularly in high-dimensional spaces. It is based on a model of social interactions among agents and draws inspiration from the coordinated dynamics of animal groups (Eberhat, R., Kennedy, J. (1995)) to search for solutions to problems.

Collective behaviors, such as flocking, emerge from the application of simple, generic rules that are independent of time and location. Craig Reynolds, a computer scientist who is known for his research and development work in the field of artificial life and simulation. He is particularly known for his boids algorithm, which simulates the flocking behavior of birds, indicated 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.

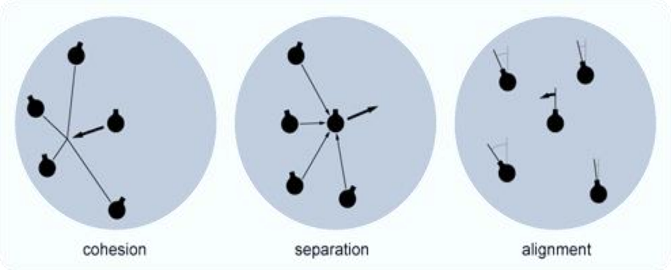


Figure 1: The Three Flocking Rules of the Swarm

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.

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.

Particles begin at random locations in a search space and move towards potential minima (or maxima) of an objective function. These particles communicate and update their positions based on the positions of other particles in the swarm and the quality of their own positions. Through repeated iterations and calculations, the particles eventually converge on one or more optimal solutions. There are various factors that can influence the effectiveness of PSO, including the size of the swarm, the communication patterns among the particles, the initial positions of the particles, the search mechanism and the function being optimized. Figure 2 shows an example of the swarm convergence.

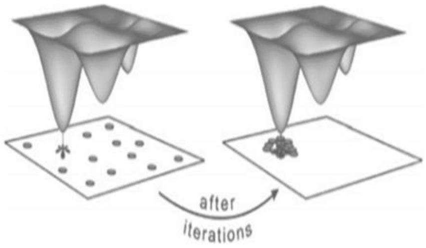


Figure 2: Illustration of the Convergence of the Swarm Over Optima

The particle movement is computed as follows:

- $x_i(t+1) = x_i(t)+v_i(t)$ (1)
- $v_i(t+1)=wv_i(t)+c_1r_1(pbest_i(t)-x_i(t))+c_2r_2(gbest(t)- x_i(t))$ (2)

$x_i(t)$ is the position of particle i at time t, $v_i(t)$ is the velocity of particle i at time t, w is an inertia factor with values between 0 and 1, $pbest_i(t)$ is the best position found by particle i so far, gbest(t) is the best position found by the swarm so far, c1 is the cognitive parameter and c2 is the social parameter. r1 and r2 are random variables between 0 and 1.

Figure 3, presents the flow chart of the general particle swarm optimization algorithm.

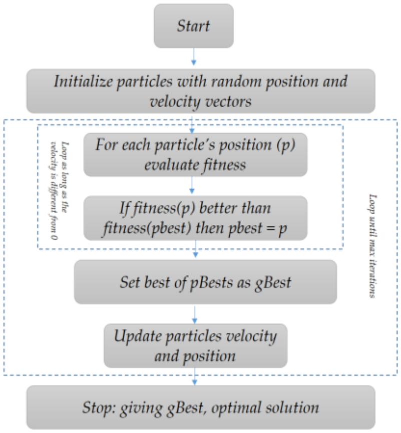

Start
Initialize particles with random position and velocity vectors
For each particle's position (p) evaluate fitness
If fitness(p) better than fitness(pbest) then pbest = p
Loop as long as the velocity is different from 0
Set best of pBests as gBest
Update particles velocity and position
Loop until max iterations
Stop: giving gBest, optimal solution

Figure 3: Flowchart of the Particle Swarm Optimization Algorithm

5 Particle Swarm Optimization and Improved Clustering

Conventional clustering algorithms, such as K-means, DBSCAN, PAM clustering, and hierarchical clustering, typically presume convex-shaped or spherical shaped-clusters (Sun, D., Toh, K. C., & Yuan, Y. (2021)) while real life data may contain irregularities, and clusters can be of different shapes. K-means, the well-known clustering algorithm, is known for its sensitivity to initialization, incorrect initialization can result in incorrect cluster assignments, leading to misclassification of anomalies.

That's where, PSO algorithm came into play, since it makes no assumptions concerning the geometry of clusters. PSC is not limited to any particular cluster shape. Instead, it relies on the collective behavior of the particles to adapt and converge to cluster solutions that minimize an objective function, which is typically associated with intra-cluster similarity and inter-cluster dissimilarity. In addition, PSO mitigates the influence of initial conditions by incorporating a population of particles that explore the search space. Particles search collectively an optimal cluster configuration, increasing the probability of detecting a globally optimal or near-optimal solution.

Setting the Inertia Weight for PSO

The inertia weight is the most critical parameter that regulates both exploration and exploitation, thus, judiciously selecting the inertia value will significantly improve the PSO algorithm associated solution, and will improve cluster quality when applied to clustering.

The inertia weight parameter is used to control the speed of the flying swarm particles. It determines how much a particle's previous velocity affects its velocity at the current time step.

The original PSO, introduced by Eberhart and Kennedy, did not include inertia weight.

However, the algorithm has suffered from the following issues:

- The particles may get stuck in local optima;
- The particles oscillate in the search space, they move back and forth between two positions without making any progress towards the optimal solution.
- The algorithm may converge too quickly or too slowly.

In 1998, Shi and Eberhart introduced the concept of Inertia Weight with a constant value. They concluded that a large Inertia Weight encourages global searching and a small one favors local searching.

To gain further insights on how the inertia weight impacts the PSO algorithm, we performed an experiment using the Griewank function.

In fact, the Griewank function has a complex topology and has many local minima and a single global minimum. Clustering algorithms can be evaluated by how well they can find the global minimum of the Griewank function. If a clustering algorithm can successfully find the global minimum of the Griewank function, then it is likely that it will be able to successfully group data points into meaningful clusters.

The experimental parameters included a generation limit of 200, 40 particles, a precision of 10E-10, an individual factor of 1, and a collective factor of 1. Three different inertia values were chosen for analysis, namely 0.1, 0.15, and 0.2.

We selected the best result after 20 repetitions for each w value.

From the experiments conducted, it is clear that when the swarm particle speed is very big (large inertia weight) the optimal fitness solution can be easily skipped although the swarm convergence is very quick. On the other hand, when the particle speed (small inertia weight) is very slow, the swarm convergence is also slow and can be trapped easily in a local minimum.

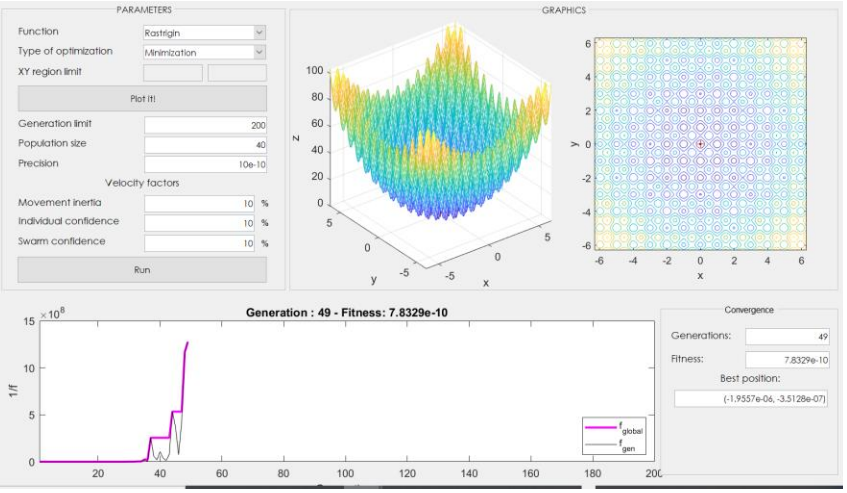
The idea, is to use a linear decreasing weight factor that allows both local and global searching, where the algorithm initially prioritizes global searching at first (large inertia weight) then gradually shifts towards favoring local searching as the iterations progress (small inertia weights). In that respect, we used the following equation:

Equation 1: Formula of the inertia weight

$$\boldsymbol{W = Wmax - K \times \frac{Wmax - Wmin}{iters}}$$

- W_{max} is the Maximal inertia weight;
- W_{min} is the minimal inertia weight;
- Iters is the number of iterations;
- The variable "k" will be assigned to each number in the range [0, iters], one at a time, as the loop iterates.

DIW-PSO, is the name we have given to this particular variant of PSO which utilizes this inertia equation.

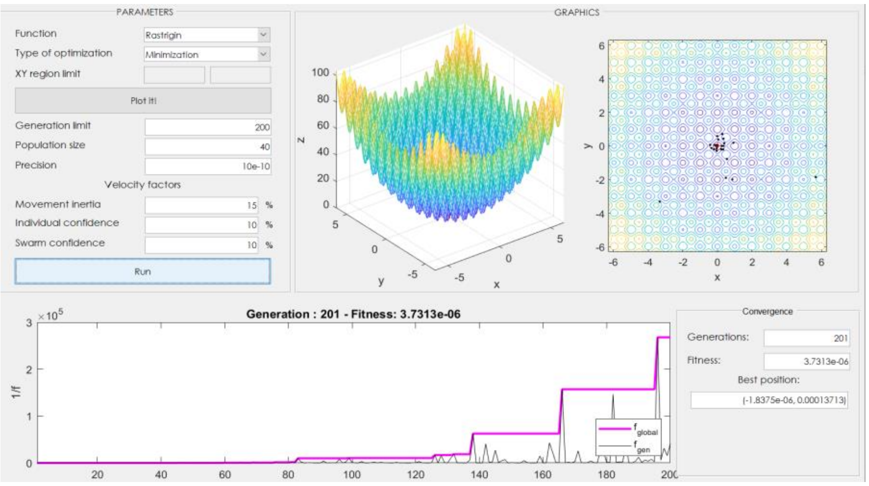


Function: Rastrigin
Type of optimization: Minimization
XY region limit:
Generation limit: 200
Population size: 40
Precision: 10e-10
Velocity factors
Movement inertia: 10 %
Individual confidence: 10 %
Swarm confidence: 10 %

Generation : 49 - Fitness: 7.8329e-10

Convergence
Generations: 49
Fitness: 7.8329e-10
Best position: (-1.9557e-06, -3.5128e-07)

Figure 4: PSO Convergence with Inertia Weight Equals to 0.1

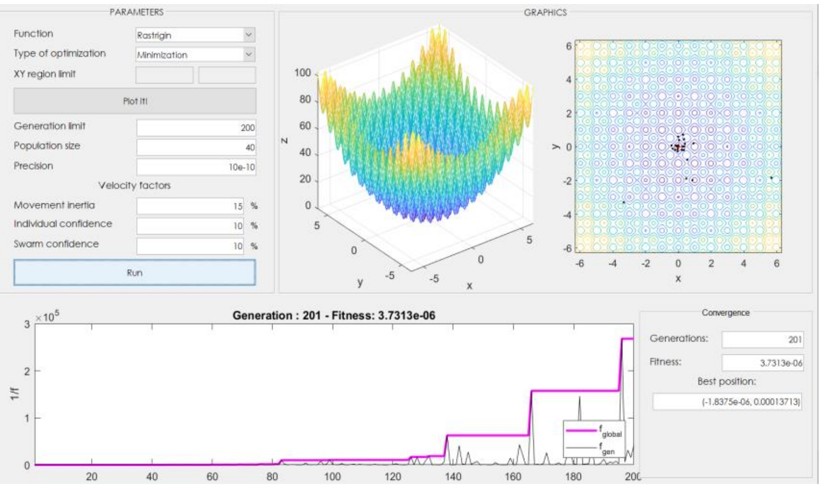


Function: Rastrigin
Type of optimization: Minimization
XY region limit:
Generation limit: 200
Population size: 40
Precision: 10e-10
Velocity factors
Movement inertia: 15 %
Individual confidence: 10 %
Swarm confidence: 10 %

Generation : 201 - Fitness: 3.7313e-06

Convergence
Generations: 201
Fitness: 3.7313e-06
Best position: (-1.8375e-06, 0.00013713)

Figure 5: PSO Convergence with Inertia Weight Equals to 0.15



PARAMETERS

Function: Rastrigin

Type of optimization: Minimization

XY region limit

Plot it!

Generation limit: 200

Population size: 40

Precision: 10e-10

Velocity factors

Movement inertia: 15 %

Individual confidence: 10 %

Swarm confidence: 10 %

Run

Generation : 201 - Fitness: 3.7313e-06

Convergence

Generations: 201

Fitness: 3.7313e-06

Best position: (-1.8375e-06, 0.00013713)

Figure 6: PSO Convergence with Inertia Weight Equals to 0.2

Our Methodology for Anomaly Detection

From a machine learning perspective, the anomaly detection can be approached as a classification or a clustering problem (Eskin, E., Portnoy, L., Stolfo, S. (2001), Bohara, B. (2020), El Bekri, M. (2019)). One way that clustering can be used for anomaly detection is by first grouping the data into normal clusters. Then, the data points that fall outside these normal clusters or far from the cluster centers can be considered as anomalies. For this purpose, we will use DIW-PSO algorithm for clustering. As in other PSO applications, each particle represents a solution to the problem. Through DIW-PSO, we seek finding similarities between data points and forming clusters by minimizing the distance between data points. The fitness function we seek to optimize is the Sum of Squared Error (SSE) (Raitoharju, J. (2017)):

$\sum_{n=1}^{N} d_{nc}^{2}$ where dnc is the Euclidean distance between the centroid and the point. By minimizing SSE, we obtain better results.

In that sense, the anomaly detection system we want to build must follow these steps:

1. Setting the initial clusters;
2. Labeling clusters as ‘normal’ or ‘anomalous’;
3. Using the labeled clusters to classify network data.

We will explain, below, the three steps in detail:

1. Setting the Initial Clusters

In order to determine the optimal number of clusters, we have used the "elbow" method. After, we encode the search space or the dataset. In fact, each particle is 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; which means that each particle represents a given configuration of clusters.

2. Labeling Clusters

After the convergence of the algorithm, we can label the resulting clusters; this is based on the following three assumptions:

- The number of normal instances in a training database is larger than the number of anomalous instances. This is because the goal is to teach the system the normal profile.
- The characteristics of an anomalous instance are different from those of a normal instance.
- Some types of anomalous instances have similar patterns and their characteristics are slightly different.

In other words, instances that appear as small clusters will be labeled as anomalies because the number of normal instances is larger than the number of intrusions according to assumption number 1, which means that normal instances should form large clusters, while anomalous data tend to belong to small clusters.

According to assumption n°2, anomalous and normal instances should not exist in the same cluster; and according to assumption n°3, a cluster containing such anomalous types of instances will be dense. We can, hence, extend theses three assumptions to a fourth assumption which states that: if a cluster has low density, it means that it is a normal cluster.

Thus, the new labeling process will be as follows:

- If a cluster is extremely dense, label it as anomalous;
- If a cluster is small, label it as anomalous;
- If a cluster is large but very dispersed, label it as anomalous
- In other cases, label it as normal;

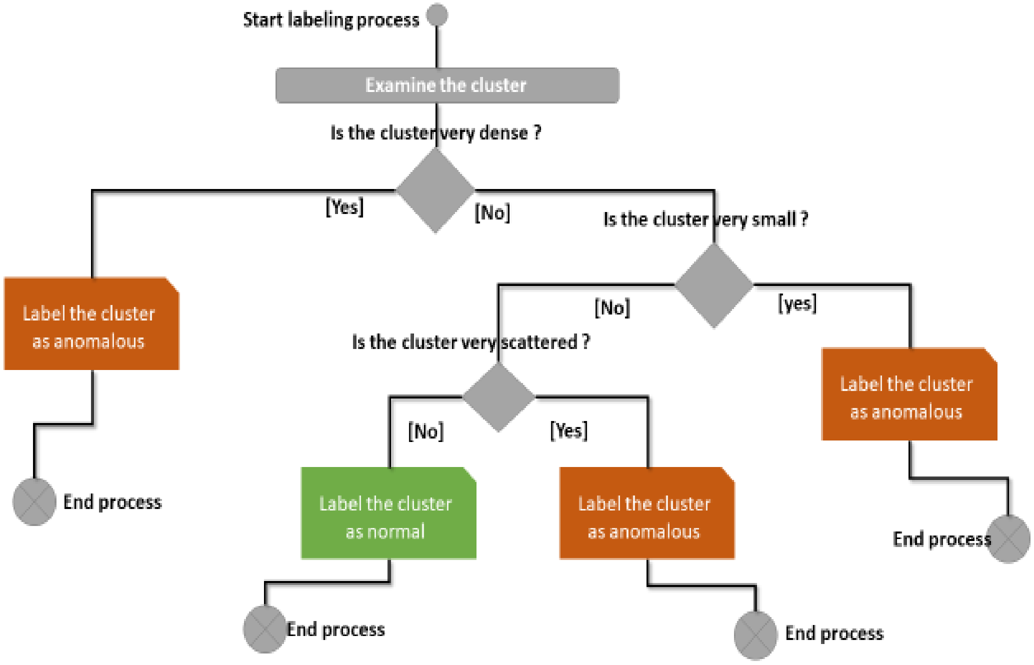

Start labeling process
Examine the cluster
Is the cluster very dense ?
[Yes]
[No]
Is the cluster very small ?
Label the cluster as anomalous
[No]
[yes]
Is the cluster very scattered ?
Label the cluster as anomalous
[No]
[Yes]
End process
Label the cluster as normal
Label the cluster as anomalous
End process
End process
End process

Figure 7: The Labeling Process

3. Detecting Anomalies

Once the clusters are created and labeled from the training set, the system will be ready to detect anomalies. Given x a data point that will be scaled to x', we calculate the distance between x' and the centroids of the different clusters using the distance metric used for cluster construction, the instance will be associated with the closest cluster and will carry its label which can be normal or anomalous.

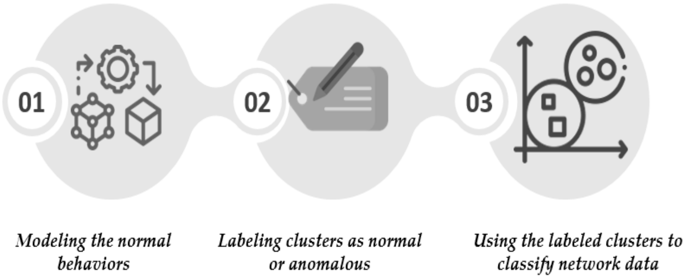

01
02
03
Modeling the normal behaviors
Labeling clusters as normal or anomalous
Using the labeled clusters to classify network data

Figure 8: The Steps of Building the Anomaly Classifier

In other words, the quality of the anomaly detection will depend on the quality of the built classifier and the quality of the formed clusters.

6 A Use Case: Anomaly Detection in Intelligent Irrigation Control System

The Functions Performed by the Intelligent Irrigation Control System

In machine learning, the effectiveness of your models is dependent on how well they perform in actual usage scenarios. The use case scenario we selected is an intelligent irrigation control system for cotton farming in which humidity and temperature sensors implemented in the field transmit information at a well-defined frequency to the microcontroller that controls the flow of water to be distributed through drip irrigation lines. (see Figure 9).

The objective of this system is to automate the irrigation process while optimizing water use. It should perform four main functions:

1) Continuously monitor the amount of moisture, temperature and water available for the plants;
2) Determine whether watering is necessary for the plants based on the information obtained from the sensors;
3) Provide the exact (or approximate) amount of water required by the plants.
4) Interrupt the supply of water when the required amount of water has been delivered to the plants. This feature is important because the amount of water available is often scarce and optimization is essential.

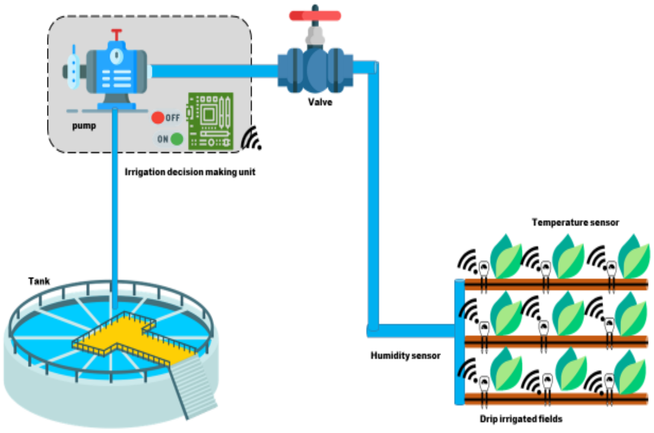


Figure 9: Intelligent Irrigation Control System Diagram

Cotton Cultivation, Optimal Growth Conditions

Understanding the business domain is critical for establishing a baseline of normal behavior in the system or network being analyzed. It gives context and insights into the system's intended functionality, purpose, and expected behavior patterns. Different systems have different functions, and what may be normal for cotton farming for example may be abnormal for corn farming. Furthermore, environmental factors such as seasonal variations, earth condition and geographical locations should be considered to avoid false positives and negatives in anomaly identification.

The cotton plant is a plant that needs a long frost-free period, a lot of heat and a lot of sun. It prefers warm and humid climates. The seeds will have a low germination rate if the soil temperature is below 15°C during active growth, the optimal temperature is between 21 and 37°C. Temperatures well above 37°C are not desirable, however, the cotton plant can survive temperatures approaching 45°C for short periods of time, but this also depends on humidity levels.

Figure 10: Image of a Cotton Field (Source: Pixabay)

International references in terms of optimal humidity levels for cotton widely applicable for the three main types of soil are summarized in table 1 (The unit of measurement is volumetric water content (VWC)):

Table 1: Cotton Moisture Levels

Type of soil	No irrigation needed(bar)	Irrigation to apply	Dangerously low soil moisture
clay	80-100	60-80	Under 60
Loamy soil	88-100	70-88	Under 70
Sandy soil	90-100	80-90	Under 80

How the Anomaly based IDS can Protect the Intelligent Irrigation Control System

The role of the anomaly-based IDS is to continually assess the functioning of the intelligent irrigation control system by tracking the combination of humidity, temperature, and water flow in order to detect any deviation from the normal irrigation pattern that may indicate a problem, such as a broken irrigation line, a clogged water dropper, a hijacked or a damaged sensor, or the decision unit etc.

The Machine Learning Model Used and the Establishment of the Baseline of Normal Behavior

It is important to create a machine-learning model that must be updated on a regular basis to guarantee its continued precision in producing results. In the case of cotton irrigation, it is crucial to remember that the water requirements of plants, soil properties, and weather conditions are all dynamic factors that are subject to change. Failing to update the model can lead to errors and ultimately harm the crop.

Our approach involves the stages below to establish the baseline of normal behavior:

1. Data Preparation: It includes handling missing values, normalizing characteristics, and selecting relevant attributes that are essential for the clustering process. This way the data is refined and made suitable for an effective clustering.
2. Parameter Fine-tuning of the clustering Algorithm: The parameters of the clustering algorithm are refined and optimized to enhance its performance. This involves running benchmakrs under different parameters to identify the best configuration for achieving imporved, accurate and meaningful clustering results.
3. Apply the clustering algorithm, in our case we have used DIW-PSO;
4. Determination of the Number of Clusters: By selecting the appropriate number of clusters, the granularity and level of detail in defining normal behavior patterns can be effectively established, in our case we used the elbow technique to determine the number of clusters.
5. Establishment of the Labeling Procedure for Clustering: A labeling procedure is established to assign meaningful labels to the clusters. This procedure categorizes and interprets the clusters based on their shared characteristics. the labeling process is explained in Figure 7
6. Examination of the resulting clusters: The generated clusters are assessed from both a quality and a business standpoint. The clusters are evaluated in terms of quality using criteria such as minimal overlapping and a balanced combination of compactness and cohesion (we use). This assessment verifies that the clusters correctly reflect separate groupings of data points. Additionally, from a business perspective, domain experts validate the clusters to guarantee their accuracy and utility. Based on their expertise, expert examination helps validate or correct the clusters. For cluster quality assessment, we use SC and Dunn indices; SC primarily focuses on assessing the cohesion and separation of individual data points within clusters, while the DI takes into account both the compactness and separation of clusters as a whole. from the business point of view the clusters are assessed according to international references in terms of optimal humidity levels for cotton (see **Table 1**);
7. Repeat, steps 3 to 6, until no correction is needed;

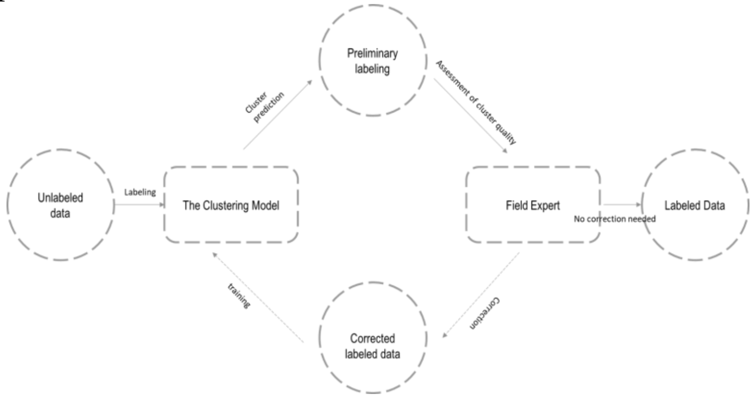


Figure 11: The Machine Learning Model

Analyzing Historical Data

We visualized the historical data through charts and plots (see Figure 12and Figure 13) in order to get a better understanding of patterns and trends; also, we used the correlation matrix (Figure 14) to understand the relationships between variables. We then normalized data to ensure that variables are on a comparable scale, which will, enable fair comparisons and avoid biases based on their original magnitudes.

The Input Data Set

There is a lack of large datasets in the field of intelligent irrigation, making it difficult to find one for this specific case. The dataset used for this study is related to a cotton crop irrigation model and consists of 200 records with three numerical attributes (https://www.kaggle.com/datasets/harshilpatel355/autoirrigationdata). While smaller datasets may result in biased results compared to larger ones, they also allow better visualization of the data and results, enabling further refinement of the model until an appropriate classifier is found. The first attribute represents soil moisture, the second indicates soil temperature, and the third is a boolean variable indicating whether or not water has been pumped. Figure 12 shows the distribution of each attribute:

- The soil moisture attribute: expressed in numerical value, its unit is the centibar, indicating the soil moisture. It is the key indicator for providing the right amount of water to the crop.
- The soil temperature attribute: expressed in numerical values, its unit is Celsius, indicating the soil temperature.
- The pump attribute: expressed in two numerical values 1 or 0, indicating the state of the irrigation pump, whether it is open or closed.

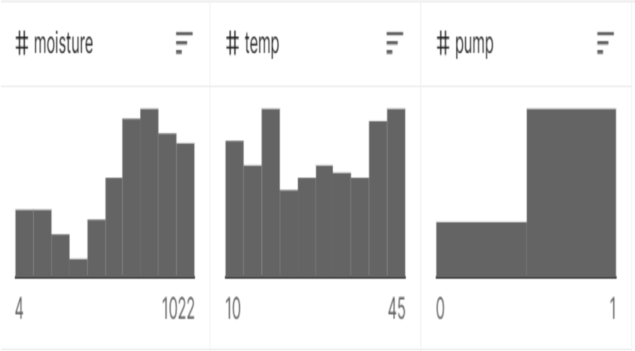


Figure 12: The Input Data Distribution

We have, then, plotted the data on a three-dimensional graph. The data is divided into two groups: data where irrigation is active and data where irrigation is suspended. The color varies according to the degree of temperature. (see Figure 13)

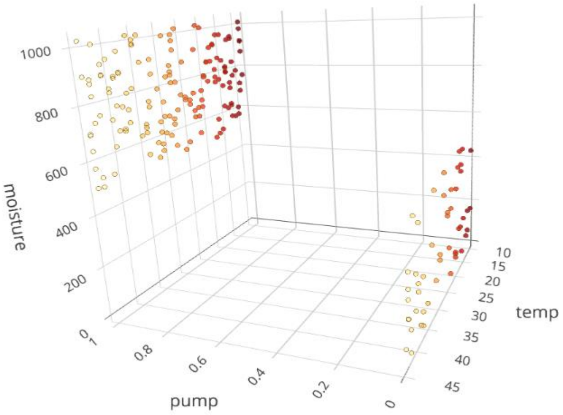


Figure 13: Input Data 3D-Plotting

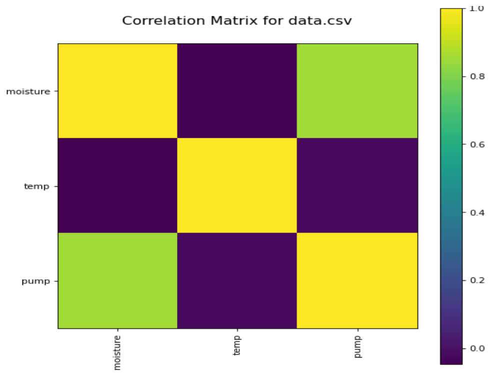


Figure 14: The Correlation Matrix of the Input Data

We then plotted the confusion matrix of the data (see Figure 14), the correlation between humidity and the pump value exceeds 60%, while the correlation between moisture and the temperature attribute is very low, which means that humidity strongly influences the irrigation decision and vice versa.

We have then normalized the data following a standardized scaling, using the following equation:

Equation 2: normalization equation

$$\text{Normalized\_df} = \frac{\text{df} - \text{df.min()}}{\text{df.max()} - \text{df.min()}}$$

We will then apply our classifier to cluster the data.

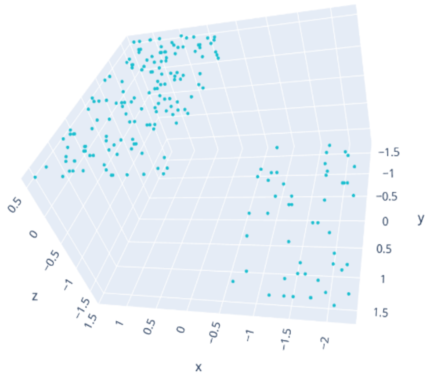


Figure 15: Data Plotting After Normalization. Vue 1

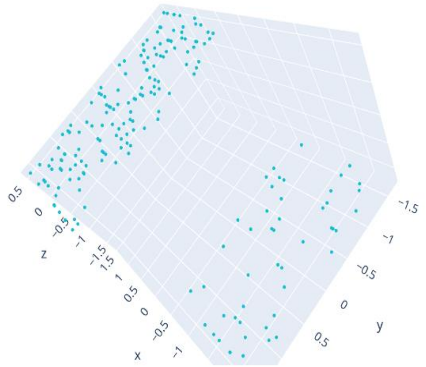


Figure 16 : Data Plotting After Normalization. Vue 2

The Input Parameters of the Simulation

We used the Evo Cluster (Aljarah, Q.R.F.H. (2021)) framework, an open-source framework implemented in Python language, which includes the most known and recent metaheuristic optimizers inspired by nature such as PSO, and we launched the experiment with the following input parameters:

- Environment: Google Colab
- Fram work: Evo cluster;
- Input data: Cotton irrigation dataset;
- DIW-PSO Optimizer parameters;
  - Vmax =6
  - Wmax=0.9
  - Wmin =0.2
  - c1=2
  - c2=2
- Numbers of clusters: Automatically defined using the elbow method;
- Main objective function to optimize: SSE (mean squared error)
- Secondary functions to optimize: The silhouette Coefficient (SC), the Dunn Index (DI)
- Number of executions: 30
- Size of the particle population: 30 particle
- the number of iterations: iters=25

Results

The number of clusters obtained using the elbow method is five; we have selected the five best results, represented in the table 2:

Table 2: Best Five Execution Results

Execution Time	SSE	SC	DI	STDev
2.19	118.0	0.33	0.98	0.45
2.23	118.19	0.33	0.99	0.45
2.25	118.82	0.36	0.98	0.45
2.23	118.1	0.34	0.96	0.45
2.21	118.27	0.33	0.98	0.45

In order to have high quality clusters, it is not enough to have the minimum value of the mean squared error only, but to take into account other metrics including the Silhouette score index and the Dun Index. The silhouette score ranges from +1 to -1, +1 indicates the best score and -1 indicates the opposite. 0 indicates that the cluster overlaps with another cluster while negative values indicate that the point is assigned to the wrong cluster. In the conducted experiment, we haven't obtain negative or null values, which means that no points were assigned to the wrong cluster, and the clusters are disjoint.

On the other hand, a value close to 1 of the Dunn index indicates that the clusters are compact and well separated or distinguished from other clusters. The values obtained in our case are greater than or equal to 0.96, a value that is close to 1 and that the criteria of compactness and separation have been well verified. After a multi-criteria analysis, which minimizes SSE, maximizes SC, and minimizes DI, the best combination fulfilling this condition is presented in table 3:

Table 3: The Best Result

SSE	**SC**	**DI**	**STDev**
118.0	0.33	0.98	0.45

Plotting and Cluster Analysis

We represented the results of the obtained clusters in 3D,

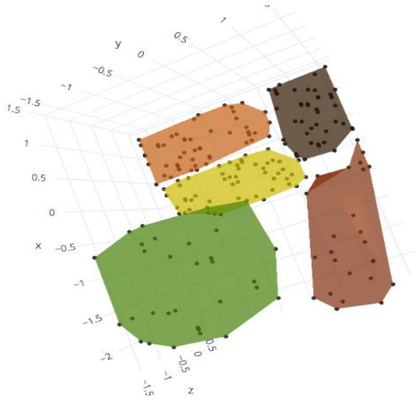


Figure 17: Resulting Clusters, Vue 1

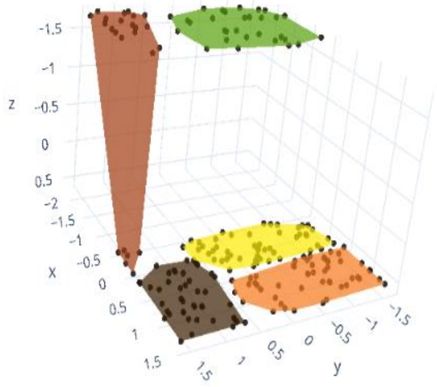


Figure 18: Resulting Clusters, Vue 2

🟧 Cluster 1 🟨 Cluster 2 🟩 Cluster 3 🟫 Cluster 4 🟫 Cluster 5

We examined the clusters, to get an idea of their cardinality; the cardinality of each cluster is represented in the following figure:

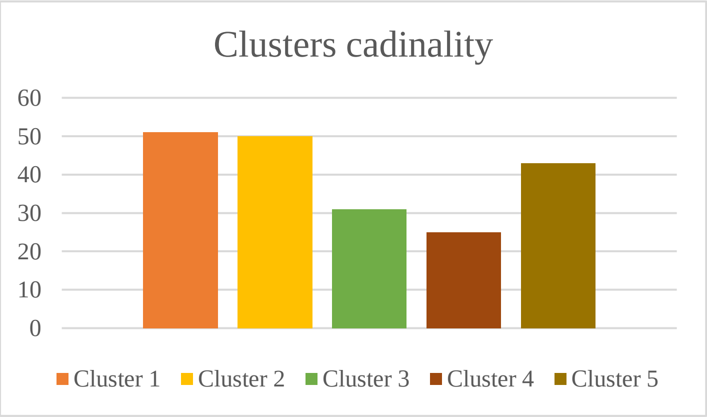


Figure 19: Clusters Cardinality

Labeling of Clusters

It is clear that cluster 4 is the smallest of the five clusters obtained, according to the labeling process; we qualify it as the anomalous cluster. In other words, all data that will be close to the center of this cluster will be qualified as anomalous.

Table 4: Cluster 4

Moisture	Temp	Pump
642	45	1
507	45	1
537	44	1
533	42	1
503	44	1
579	45	1
89	42	0
76	35	0
421	39	0
208	39	0
124	40	0
245	43	0
466	41	0
75	37	0
209	43	0
304	43	0
256	40	0
141	43	0
4	42	0
178	39	0
39	38	0
21	37	0
206	37	0
143	43	0
52	44	0

Analysis and Discussion

At the level of clusters 1, 2, and 5, the data obtained does not indicate any apparent anomaly, indeed, the humidity level is high which indicates that the soil has been well irrigated, which is still consistent with state of the irrigation pump put at the state ‘’ON’’. On the other hand, at the level of cluster 3, the data indicates low humidity levels, which is normal given that the pump is in the "OFF" state. The data on clusters 1, 2, 3, and 5 represent a certain coherence with the requirements of the plant and represent correct irrigation decisions that put the plant in its optimal growth conditions.

In other words, any instance, after scaling, that will be close under the Euclidean metric to the centroids of the mentioned four clusters will contribute to the growth of the cotton crop.

At the scale of cluster 4 (see Table 4), we see that in the face of very low soil humidity levels and high temperature, the pump is “OFF”, also, the data shows disturbance in the water flow and sudden changes, and an acute drop in humidity levels which may indicate faulty behavior of the humidity sensors. Cases of this cluster show that the choices made by the system do not support the plant's optimal growth zone; in other words, any new instance, scaled, that will be close under the Euclidean distance to the centroid of cluster 4 will be considered anomalous, because this decision will only contribute either to the hydric stress of the harvest or to water waste.

7 Conclusion

In conclusion, this paper, which is an extension of our previous work (EL BEKRI, M. (2022)), demonstrates that particle system-based stochastics can be extended to new use cases and can be used to solve complex problems such as optimization of anomaly-based intrusion detection systems.

It is important to recall, that signature-based intrusion detection systems have the largest market share, and the use of anomaly-based intrusion detection systems is still being debated within the academic community and they have not yet reached a level of maturity sufficient for a widespread adoption in the field of cyber security. We wanted to advance this topic and present concrete steps on how to implement anomaly-based IDS. In addition, the anomaly detection problem was most of the times addressed using simple statistical measures to calculate deviations from the center or mean value, however, anomalies remain unknown mysteries that require sophisticated approaches to be detected.

In fact, the particle swarm optimization algorithm coupled with machine learning has shown its effectiveness in various optimization problems and we wanted to take advantage of this combination to optimize anomaly detection. We have chosen intelligent irrigation as a use case to test our proposal; We believe that DIW-PSO in particular and similar particle system-inspired stochastic optimization algorithms have a future in the world of intrusion detection systems. We also believe that building such classifiers addresses the issue of the rarity and the cost of labeled data. Through this mechanism, a self-labeling process of data is set up, favoring the identification of the normal behavior of any intelligent irrigation system.

References

[1] Aboueata, N., Alrasbi, S., Erbad, A., Kassler, A., & Bhamare, D. (2019). Supervised machine learning techniques for efficient network intrusion detection. *In IEEE 28th International Conference on Computer Communication and Networks (ICCCN)*, 1-8.

[2] Afkhami, M., Hassanpour, A., & Fairweather, M. (2019). Effect of Reynolds number on particle interaction and agglomeration in turbulent channel flow. *Powder Technology*, *343*, 908-920.

[3] Al-Imran, M., & Ripon, S.H. (2021). Network intrusion detection: an analytical assessment using deep learning and state-of-the-art machine learning models. *International Journal of Computational Intelligence Systems*, *14*, 1-20.

[4] Bohara, B., Bhuyan, J., Wu, F., & Ding, J. (2020). A survey on the use of data clustering for intrusion detection system in cybersecurity. *International journal of network security & its applications*, *12*(1), 1-18.

[5] Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey.

[6] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 1-58.

[7] Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *In IEEE MHS'95. Proceedings of the sixth international symposium on micro machine and human science*, 39-43.

[8] El Bekri, M., & Diouri, O. (2019). Pso based intrusion detection: A pre-implementation discussion. *Procedia Computer Science*, *160*, 837-842.

[9] El bekri, M.O.H.A.M.E.D., Diouri, o., & Chiadmi, d. (2022). A review of the particle swarm clustering method for intrusion detection in IOT. *Journal of Theoretical and Applied Information Technology*, *100*(9), 2799-2810.

[10] Eskin, E., Portnoy, L., & Stolfo, S. (2001). Intrusion detection with unlabeled data using clustering. In *Proceedings of ACM CSS Workshop on Data Mining Applied to Security*.

[11] Grill, M., Pevný, T., & Rehak, M. (2017). Reducing false positives of network anomaly detection by local adaptive multivariate smoothing. *Journal of Computer and System Sciences*, *83*(1), 43-57.

[12] Giorgi, G., Abbasi, W., & Saracino, A. (2022). Privacy-Preserving Analysis for Remote Video Anomaly Detection in Real Life Environments. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, *13*(1), 112-136.

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

[14] https://www.kaggle.com/datasets/harshilpatel355/autoirrigationdata

[15] Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences*, *9*(20), 1-28.

[16] Moermond, T.C. (1990). A functional approach to foraging: morphology, behavior, and the capacity to exploit. *Studies in Avian Biology*, *13*, 427-430.

[17] Qaddoura, R., Faris, H., Aljarah, I., & Castillo, P.A. (2021). Evo Cluster: an open-source nature-inspired optimization clustering framework. *SN Computer Science*, *2*, 1-12.

[18] Raitoharju, J., Samiee, K., Kiranyaz, S., & Gabbouj, M. (2017). Particle swarm clustering fitness evaluation with computational centroids. *Swarm and evolutionary computation*, *34*, 103-118.

[19] Reeves, W.T. (1983). Particle systems—a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics (TOG)*, *2*(2), 91-108.

[20] Roesch, M. (1999). Snort: lightweight intrusion detection for networks. *In Lisa*, 99(1), 229-238.

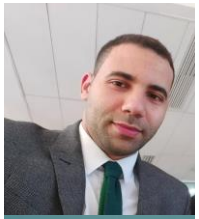
[21] Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A.A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *computers & security*, *31*(3), 357-374.

[22] Smith, J. (2019). Climate Change: A Comprehensive Guide. Publisher.

[23] Sun, D., Toh, K.C., & Yuan, Y. (2021). Convex clustering: model, theoretical guarantee and efficient algorithm. *The Journal of Machine Learning Research*, *22*(1), 427-458.

[24] Wyndham, F. S., & Park, K. E. (2018). “Listen Carefully to the Voices of the Birds”: A Comparative Review of Birds as Signs. *Journal of Ethnobiology*, *38*(4), 533-549.

Authors Biography



Mohamed EL BEKRI, Phd Student, at the Ecole Mohammadia d’Ingénieurs - Department of Computer Engineering - Mohammed V University of Rabat, Morocco.
IT engineer at Data Protection authority, CNDP, Morocco;
Research interests: Swarm Intelligence, Particle systems, IoT, cybersecurity, privacy.

Ouafaa Diouri, Professor since 1988 at Ecole Mohammadia d’Ingénieurs - Department of Computer Engineering - Mohammed V University of Rabat, Morocco
Permanent member SIP research Lab.
Research interests: Application security, cybersecurity, Blockchain, IoT.

Dalila Chiadmi, Full Professor at Ecole Mohammadia d’Ingénieurs, - Department of Computer Engineering - Mohammed V University of Rabat, Morocco
Director of SIP Research Lab
Former head of Department of Computer Engineering
Research interests: Big Data, Big graphs, services Computing, Smart Cities, Intelligent Transport System, and open gov.