The 19th International Conference of Iranian Aerospace Society
AERO2021
2021 - Tehran
Iran Helicopter Support & Renewal Co.
AERO2021-06190426

# GNSS Interference Classification using Multi-Layer Perceptron Neural Network Trained by AGPSO

**Mohammad Reza Ghasemi[1], Samira Tohidi[2], Mohammad Reza Mosavi[3*]**
1- Bachelor Student, Department of Electrical Engineering, Iran University of Science and Technology, mohammad_qasemi@elec.iust.ac.ir
2- Ph.D. Student, Department of Electrical Engineering, Iran University of Science and Technology, s_tohidi@elec.iust.ac.ir
3- Professor, Department of Electrical Engineering, Iran University of Science and Technology, m_mosavi@iust.ac.ir
*Corresponding Author

**Abstract**
Spoofing attacks are a fundamental threat to civilian Global Navigation Satellite System (GNSS) applications due to their powerful impact on receivers. As a result, plenty of anti-spoofing methods have been developed by researchers in recent years. In this paper, we have proposed a technique for the detection and classification of GNSS interferences based on the so-called Power-Distortion (PD) detector. The PD detector uses received signal power and correlation-profile distortion monitoring for detecting any type of interference including spoofing attacks, jamming, or multi-path. We illustrate that detection and classification can significantly be improved by replacing prior methods of classification with our proposed method. The method uses a Multi-Layer Perceptron Neural Network (MLP NN) trained by Particle Swarm Optimizer with Autonomous Group (AGPSO) in which we will call it MLP NN-AGPSO classifier. The primary usage of this observation is the diagnosis of spoofing attacks among other interferences. The results show that the MLP NN-AGPSO detector exhibits improved detection and classification accuracy. Results obtained from simulation show that AGPSO3 has better classification performance in comparison to AGPSO1 and AGPSO2. More specifically, multi-path signal detection has a great accuracy of 95.33%, and spoofing and jamming are 90.11% 98.67% accurate, respectively.
**Keywords:** *GNSS- Spoofing- Jamming- Multi-path- Classification- MLP NN- PSO.*

## 1. Introduction

The use of Global Navigation Satellite Systems (GNSS) is spreading in today's society. Many critical applications rely on GNSS signals to correctly identify position and timing. However, they are still vulnerable to interference and even low-power signal can deceive GNSS receivers. Interferences have been categorized into two main groups: (1) intentional and (2) unintentional. Jamming and spoofing are placed in the former group and multi-path is in the latter. In jamming attacks, the radio transmitter emits a signal that with higher power to mask the authentic signal and prevent the receiver from obtaining the authentic signal. Multi-path is the result of environmental reflections of the main signal. A spoofing attack consists of transmitting GNSS-like signals, to fool the receiver into false tracking. Because of this, the spoofer can control the Position, Velocity, and Time (PVT) solution of the receiver which it believes to be true. GNSS-based systems security and protection against different interferences are the main issues in this field.

GNSS authentication techniques are generally categorized into three groups: (1) cryptographic, (2) geometric, and (3) signal processing methods. The first group, utilize unpredictable modulation in spreading code or navigation data, although the modulation can be verified to indicate the authentic signal. Geometric method employs the authentic signals' angle-of-arrival diversity. Approaches like the Power-Distortion (PD) detector [1] and parameter estimation methods [2] are placed in the last group. Techniques that are low cost, without any need of supplementary hardware, and those that can be achieved just by software update are practical and beneficial tools for GNSS signal authentication. Monitoring autocorrelation profile distortion and monitoring obtained power from Automatic Gain Control (AGC) set points and falls into this category.

Artificial Neural Networks (ANNs) are taken into consideration because of their prominent features in parameters estimation and pattern recognition. Multi-Layer Perceptron Neural Network (MLP NN) is one of the most common types of ANNs. Recently, many heuristic algorithms like Genetic Algorithm (GA) [3], Particle Swarm Optimization (PSO) [4,5], Ant Colony Optimization (ACO) [6], etc. have been utilized to train the ANNs to improve the functionality of the network. The common issue of all these algorithms is trapping in local optima and convergence speed. In this paper, we employ the AGPSO algorithm to train our MLP NN since we are dealing with a high dimensional problem and also its property in converging to the best solution and not trapping in local optima.

The rest of the paper is organized as follows. The measurement model, the power of the received signal, and

signal distortion are discussed in Section 2. In Section 3, we have overviewed ANN and MLP NNs. Following the PSO and AGPSO algorithms have been introduced in Section 4. Section 5 introduces the proposed MLP NN-Particle Swarm Optimizer with Autonomous Group (AGPSO) method for classification and the simulation results and validation for the proposed method are illustrated in Section 6. The final section provides a conclusion.

## 2. Measurement Models

Here, we discuss the measurement of two proposed metrics that we already mentioned, i.e., power and correlation distortion. Before that, some metrics are introduced for better understanding of the concept.

First, Signal Quality Monitoring (SQM) is used to measure the signal distortion amount. Researchers have introduced many metrics for interference detection in this field [7]. SQM is based on continuously observing received GNSS signals for possible interference and distortion. The SQM is very powerful in detecting matched-powered spoofing attacks, where the spoofing signal is broadcast with only a slight power advantage, so the distortion generated at the correlation function between the signal code and code local replica is maximal. SQM-based methods are not applicable if the shape of the correlation peak is not affected by the spoofing attack.

Second, the Symmetric Difference (SD), which is one of the metrics used for detecting spoof interferences [1]. It measures the distortion that can be caused by eighter spoofing or multi-path. For instance, in presence of noise, multi-path interference, and spoofing-free situations, the symmetric difference is equal to 0. In practice, SD is always non-zero and a large SD denotes the spoofer presence. The detectors based on this metric have two main deficiencies. First, standard SD uses just two taps of correlation function which make it insensitive to correlation-function distortions that are not aligned with these taps. The other deficiency is that the SD is dependent on tracking of the receiver's code to adjust the correlation taps around the authentic correlation peak symmetrically. It should be considered that very low noise interferences, like thermal noise, exclude the code-tracking loop to align taps pair flawlessly symmetric. This issue can be solved by using multiple correlation taps. In this approach, elicited values for the correlation by extra taps are exploited to obtain code and carrier phase and gain-controlled amplitude using maximum-likelihood estimations in the single signal correlation function model. After that, the correlation function estimation model is deducted from the correlation function value obtained by each tap. The remainders of the result are considered as the distortion measurement. In the following, the main metrics are described.

### 2.1. Received Signal Power Measurement

Signal's power measurement was first introduced in [8,9] as an interference detection metric. Authors in [10] have presented this method as a spoof detection strategy. In this method, the AGC outputs are used to distinct any extra signals that is found in receiver antenna except the authentic signal.

The received signal's power is a simple and beneficial interference indicator. The signal power ($P_k$) is measured over $t_k$ in RF band according to Eq. (1). As $P_k$ is very low on the earth surface, even a weak interference signal can affect the authentic signal's power. In these cases, a threshold is usually considered to monitor unusual and unawares changes in $P_k$. Power monitoring with other monitoring techniques is a useful metric for interference detection.

$$P_k = 10log(\frac{1}{T}\int_{t_{k-1}}^{t_k} |\hat{r}_c(t)|^2\, dt) \tag{1}$$

Where $\hat{r}_c(t)$ is the filtered version of $r_c(t)$ and $r_c(t)$ represents the signal that exits from front-end output and $P_k$ is $\hat{r}_c(t)$ average power. In receivers with AGC in their front-end, $P_k$ is computed by the AGC set point.

### 2.2. Distortion Measurements using Multi-Tap Maximum-Likelihood Estimator

The multi-tap maximum-likelihood estimator is used to decompose the raw GNSS in-phase and quadrature samples into two estimated signal models. Authors in [11] describe this estimator in detail. The maximum-likelihood estimator uses $l$ number of signal correlator taps. The tap that is located at the center, follows the assessed correlation function peak of the receiver while the others are aligned symmetrically around it. $\xi_k(\tau)$ illustrates the authentic signal correlation function. The correlation function of $j^{th}$ tap is considered as Eq. (2):

$$\xi_k(\delta_j) = \beta_k[\xi_{Ak}(\delta_j) + \xi_{Nk}(\delta_j)] = a_{Ak} exp(jj_{Ak}) R(\delta_j - \tau_{Ak}) + \beta_k \xi_{Nk}(\delta_j) \tag{2}$$

where $\delta_j$ represent $j^{th}$ tap location, $k$ shows the time index, and $\tau_{Ak}$, $\varphi_{Ak}$ are the authentic signal's code phase and carrier phase, respectively. $a_{Ak}$ is the gain-controlled amplitude. $a_{Ak}$, $\tau_{Ak}$, and $\varphi_{Ak}$ are estimated using a technique adapted from the maximum likelihood described in [11]. Each set of estimates $\{\hat{a}_{Ak}, \hat{\tau}_{Ak}, \varphi_{AK}\}$, has a cost function $J_k$ which is calculated by the Eq. (3):

$$J_k = \left\| \xi_k - H^T(\hat{\tau}_{Ak,\delta}) \hat{a}_{Ak} \exp(j\hat{\varphi}_{Ak}) \right\| \tag{3}$$

where $\|x\|_T^2 = x^T T^4 x$ is the norm $x$ definition. The obtained two sets of estimations return the lowest cost $J_k$. Since $J_k$ has a reverse ratio with likelihood estimation, the lower the value of the cost function, the higher the likelihood estimation will be. At next level, a filtered code-phase is obtained. New carrier phase and amplitude estimations are determined after each split point. These two steps are repeated until $J_k$ is not considerably changed. By $J_k$ convergence, the maximum-likelihood outputs are collected. The final $J_k$ is considered as the

signal distortion value $D_k$. If the cost function $J_k$ is large, it means that the spoofing attack or multi-path interference exists. On the other hand, the correlation function value is proportionate truly if the $J_k$ is scanty. Using $l$ number of correlation taps superior in contrast with the symmetric difference, which uses only two taps. It extracts more information which makes it more insensitive against noise- and dynamics-induced misalignment, unlike the SD that makes false reports in such cases [12].

## 3. Multi-layer Perceptron Neural Network

ANNs are one of the widely used computational tools for estimation and classification problems. Recently, ANNs are taken into consideration for their outstanding achievements. ANN performance is tightly related to choosing the right learning algorithm [13]. Back Propagation (BP) algorithm is one of the most popular learning algorithms that minimize the Mean Squared Error (MSE) between the desired and the estimated outputs for the particular network inputs [14]. For prediction problems, BP learning algorithm is used to train MLP NN by computing the connection weights and biases. Despite BP algorithm benefits, it has some drawbacks. First, the result may be trapped in local optima because of high dependency on initial parameters, which means the convergence is not guaranteed [15]. Second, the slow convergence rate may lead BP to get trapped in many iterations. The BP algorithm deficiencies may result in a slow learning process or sometimes produce less accurate outcomes [13,14].

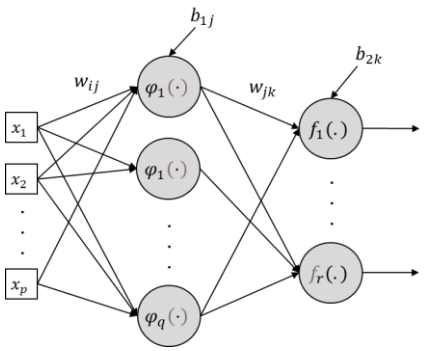


Fig. 1: MLP NN with $(p, q, r)$ structure.

MLP NN is one of the widely used multilayer ANNs. Each MLP NN has at least three layers: (1) the input layer, (2) the hidden layer, and (3) the output layer. Fig. 1 shows a three-layer MLP NN with $p$ inputs, $q$ hidden neurons, and $r$ output neurons. $w_{ij}$ is the weight that connects $i^{th}$ input node and $j^{th}$ node in the hidden layer, $b_{1j}$ is bias weight for $j^{th}$ node in the hidden layer, $w_{jk}$ is the connection weight between $j^{th}$ node in the hidden layer and $k^{th}$ node in the output layer, and $b_{2k}$ is bias weight for $k^{th}$ node in the output layer. In this paper, a MLP NN has been utilized to detect and classify PD dataset into jamming, spoofing, multi-path, and interference-free classes.

The outputs of first hidden layer neurons are computed by Eq. (4):

$$v_j = \varphi\left(\sum_{i=1}^{p} \omega_{ij} x_i + b_{1j}\right) \tag{4}$$

where $p$ is the number of NN inputs, $w_{ij}$ represents the connecting weights between the $i^{th}$ node of the inputs and $j^{th}$ node in the hidden layer, and $x_i$ shows the $i^{th}$ input. After calculating the hidden layer outputs, network estimations would be calculated as follows:

$$y_k = f\left(\sum_{j=1}^{q} \omega_{jk} v_j + b_{2k}\right) \tag{5}$$

where $q$ is the number of hidden layer nodes, $w_{jk}$ represents the weights between $j^{th}$ node of hidden layer and $k^{th}$ node of the output, and $v_i$ represents $j^{th}$ node of the hidden layer output

## 4. PSO Algorithm Overview

PSO is one most common optimization algorithms that is used extensively. The main idea of the PSO algorithms is adopted from the animals' collective behavior [16,17]. It uses sets of particles to find the best solution in the searching area. Each particle has a position and a velocity. Updating particles are mainly based on the particle's distance from its local best solution and general best solution. PSO algorithm is modeled as follows in Eq. (6):

$$\begin{aligned} v_i^{t+1} &= w v_i^t + c_1 \cdot r_1 \cdot (pbest_i - x_i^t) + c_2 \cdot r_2 \\ &\quad \cdot (gbest - x_i^t) \\ x_i^{t+1} &= x_i^t + v_i^{t+1} \end{aligned} \tag{6}$$

where $v_i^t$ is the velocity of $i^{th}$ particle at $t^{th}$ iteration, $w$ is inertia weight to control the PSO algorithm stableness and it is typically chosen between 0.4 and 0.9, $c_1$ and $c_2$ are cognitive and co-operative coefficients respectively, $r_1$ and $r_2$ are random numbers between 0 and 1 to give the PSO algorithm the ability of random search. $x_i^t$ shows the $i^{th}$ particle in time $t$. $pbest_i$ and $gbest$ are $i^{th}$ particle local best solution and general best found. $w$, $c_1$, and $c_2$ are the three main coefficients that the PSO performance is dependent to.

One of the PSO advantages that makes it popular is its simplicity and low-cost computation. However, trapping in local optima and slowness of convergence velocity are two inevitable issues [18], for most evolutionary algorithms. One of the approaches to handle these problems is using dynamic parameter tuning which improves the efficiency of the PSO algorithm without increasing in computational cost. AGPSO algorithm's main idea focuses on $c_1$ and $c_2$ coefficients to improve the convergence of PSO.

### 4.1. Autonomous Group PSO Algorithm

The tricky part of all the optimization algorithms is finding the global minimum [19]. One of the issues in the most minimization algorithm is being trapped in local minima. AGPSO employs different groups of $c_1$ and $c_2$ to solve this problem. In the basic PSO algorithm, particles behave similarly in terms of local and general searching. The resemblance of these strategies means that all the parameters are tuned just like each other with the same strategy, which means that particles are obliged to search without intelligence, but in AGPSO algorithm instead, tries different ways to tune optimization parameters. In this model, $c_1$ and $c_2$ are updated with different strategies which are represented as continuous mathematical functions in Table 1. Figures 3, 4 and 5 illustrate the behavior of $c_1$ and $c_2$ over the iterations. During the iterations, $c_1$ is reduced and $c_2$ is increased. This means that, during first iterations that $c_1$ is bigger than $c_2$, particles tend to explore local range, then in the next iterations, when $c_2$ gets bigger than $c_1$, they search the area generally.

Table 1: Updating strategies in AGPSO.

| AGPSO | | Updating formula | |
|---|---|---|---|
| | | c1 | c2 |
| AGPSO1 | Group 1 | $(-2.05/\mathrm{T})\mathrm{t} + 2.55$ | $(1/\mathrm{T})\mathrm{t} + 1.25$ |
| | Group 2 | $(-2.05/\mathrm{T})\mathrm{t} + 2.55$ | $(2\mathrm{t}^3/\mathrm{T}) + 0.5$ |
| | Group 3 | $(-2\mathrm{t}^3/\mathrm{T}^3) + 2.5$ | $(1/\mathrm{T})\mathrm{t} + 1.25$ |
| | Group 4 | $(-2\mathrm{t}^3/\mathrm{T}^3) + 2.5$ | $(2\mathrm{t}^3/\mathrm{T}^3) + 0.5$ |
| AGPSO2 | Group 1 | $2.5 - (2\log(\mathrm{t})/\log(\mathrm{T}))$ | $(2\log(\mathrm{t})/\log(\mathrm{T})) + 0.5$ |
| | Group 2 | $(-2\mathrm{t}^3/\mathrm{T}^3) + 2.5$ | $(2\mathrm{t}^3/\mathrm{T}^3) + 0.5$ |
| | Group 3 | $0.5 + 2\exp[-(4\mathrm{t}/\mathrm{T})^2]$ | $2.2 - 2\exp[(4\mathrm{t}/\mathrm{T})^2]$ |
| | Group 4 | $2.5 + 2(\mathrm{t}/\mathrm{T})^2 - 2(2\mathrm{t}/\mathrm{T})$ | $0.5 - 2(\mathrm{t}/\mathrm{T})^2 + 2(2\mathrm{t}/\mathrm{T})$ |
| AGPSO3 | Group 1 | $1.95 - 2\mathrm{t}^{1/3}/\mathrm{T}^{1/3}$ | $2\mathrm{t}^{1/3}/\mathrm{T}^{1/3} + 0.05$ |
| | Group 2 | $(-2\mathrm{t}^3/\mathrm{T}^3) + 2.5$ | $(2\mathrm{t}^3/\mathrm{T}^3) + 0.5$ |
| | Group 3 | $1.95 - 2\mathrm{t}^{1/3}/\mathrm{T}^{1/3}$ | $(2\mathrm{t}^3/\mathrm{T}^3) + 0.5$ |
| | Group 4 | $(-2\mathrm{t}^3/\mathrm{T}^3) + 2.5$ | $2\mathrm{t}^{1/3}/\mathrm{T}^{1/3} + 0.05$ |

In Table 1, $t$ and $T$ are the current and the maximum number of iterations, respectively. Based on each particle's group among those in Table 1, elements search the area differently. For instance, in group 1 from AGPSO1, particles have social behavior in the first iterations, while members of group 4 tend to investigate the area exclusively in the most iterations.

## 5. Proposed MLP-PSO Methods

This section introduces the proposed MLP NN trained by AGPSO method for classification.

### 5.1 Multi-Layer Perceptron Neural Network Structure

For the classification problem, we used a four-layer MLP NN with (2,5,4,1) structure. However, the output layer is unsupervised because we just want to map the output from the previous layer to logical values, 0 or 1. Three transfer functions used in the model are defined as the following:

$$\varphi(x) = xe^{-\frac{1}{2}x^2} \quad (7)$$

$$\psi(x) = \frac{1}{1+e^{-x}} \quad (8)$$

$$f([x_1, x_2, \dots, x_k]) = compet([x_1, x_2, \dots, x_k]) \quad (9)$$

The network inputs are the power of the received signal and correlation function distortion. The desired output is the corresponding class according to the inputs. MSE of the MLP NN output is considered as the fitness function for the PSO algorithm and it's calculated as follow:

$$MSE = \frac{1}{n} \times \sum_{1}^{n} \left(Y_{desire,i} - Y_{output,i}\right)^2 \quad (10)$$

where $Y_{output,i}$ represent the identified class detected by the classifier and $Y_{desire,i}$ shows the actual class for $k^{th}$ sample in dataset. In each iteration, the MSE of each particle from PSO is calculated and sent to the PSO to modify its particles' positions and velocities.

### 5.2 Training MLPNN by AGPSO Algorithm

AGPSO algorithm is utilized as the MLP NN training algorithm. The designed classifier is so-called the MLP NN-AGPSO classifier. PSO algorithm has been employed for its properties in fast convergence speed, not being trapped in local minima, etc. Parameters $c_1$ and $c_2$ are chosen from Table 1 according to the AGPSO type. Each of the four groups in the selected AGPSO type is randomly assigned to the PSO particles. $w$ is the weighting function which is linearly decreased from 0.9 to 0.4. Eq. (11) shows the relation of $w$ in each iteration.

$$w = 0.9 - 0.5 * (t/T) \quad (11)$$

where $t$ and $T$ are the current iteration and the maximum number of iterations, respectively.

To train an MLP NN by heuristic algorithms like PSO, the whole structure should be represented as Fig. 2.

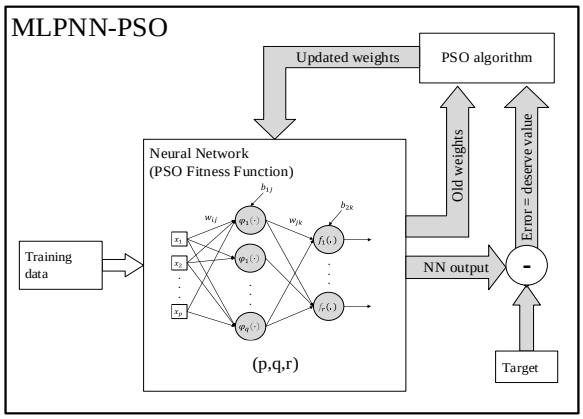


Fig. 2: Training MLP NN by PSO algorithm (MLP NN-PSO).

There are three possible ways to represent a network's weights and biases: vector, matrix, and binary. In vector and matrix representation, each element is represented as a vector and matrix respectively, and in the binary representation, elements are given as a sequence of bits. Here as our NN is not complex, vector representation has been utilized for the connection weights and biases. Eq. (12) shows a sample of vector representation:

$$Particle = [w_1, w_2, w_3, \dots, b_1, b_2, b_3, \dots] \qquad (12)$$

Here, particles are the MLP NN weights. In each iteration, updated particles are applied to the network to calculate the MSE of the predicted output. The MSE is used as PSO particles' fitness so it will be sent to the PSO algorithm to find $gbest, pbest$, and update the position and the velocity of the particle.

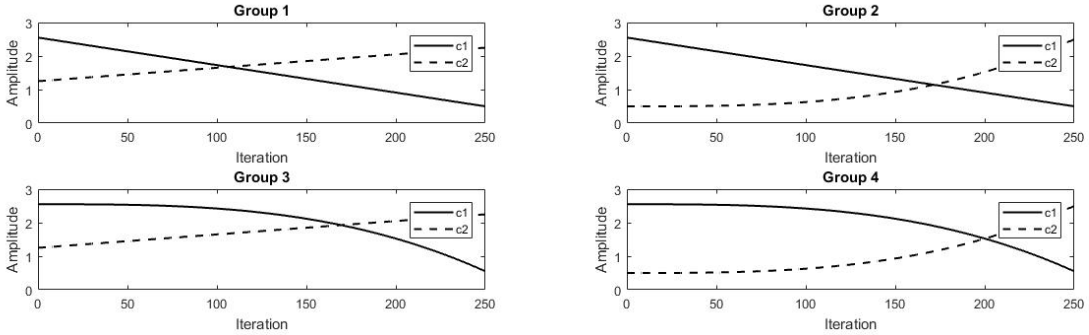


Fig. 3: AGPSO1 autonomous groups behavior during iterations.

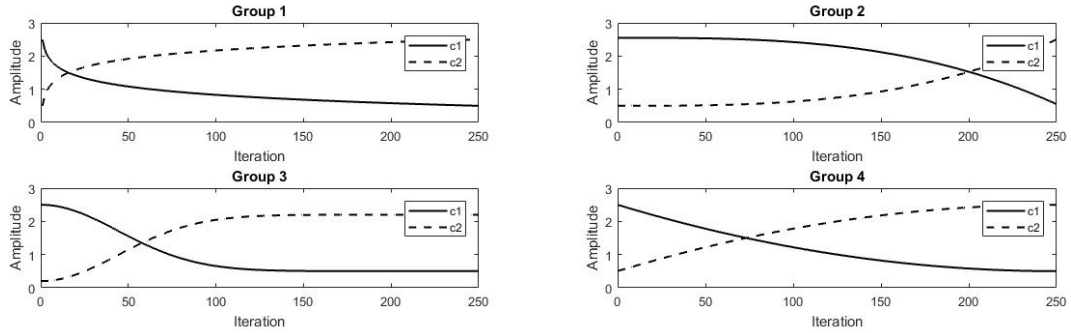


Fig. 4: AGPSO2 autonomous groups behavior during iterations.

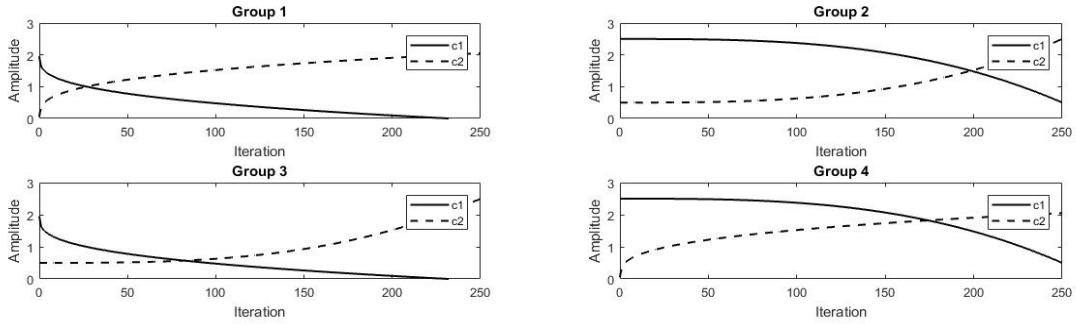


Fig. 5: AGPSO3 autonomous groups behavior during iterations.

The PSO algorithm's particles are initialized with randomly distributed positions and their velocity is considered zero. The values of each dimension are limited distributed positions and the initial velocities for the particles between -1 and 1. Then, the particles will be divided into the AGPSO groups (from group 1 to group 4) accidentally. After each iteration, $pbest$ and $gbest$ for each particle is calculated based on the mean squared error from the MLP NN. Finally, $c_1$ and $c_2$ are calculated according to the group strategy to update the particles' position and velocity using Eq. (6).

## 6. Simulation and Results

This section illustrate the simulation results and validation for the proposed method.

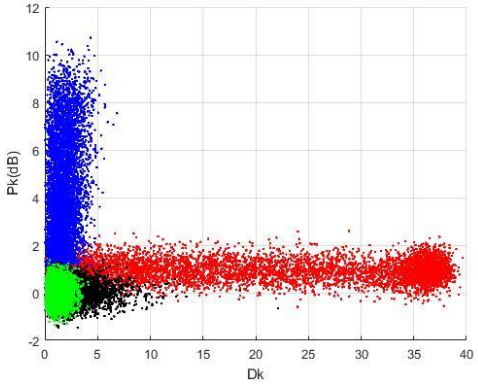


Fig. 6: Simulated distortion ($D_k$) and received power ($P_k$) measurements for interference-free (green), multi-path (black), spoofing (red), and jamming (blue).

All three AGPSO algorithms have been assessed 15 times and the best result of each one is chosen as the final result for comparison. The maximum iteration for the AGPSO algorithm is set to 250. Fig. 6 is the training dataset for MLP NN-AGPSO.

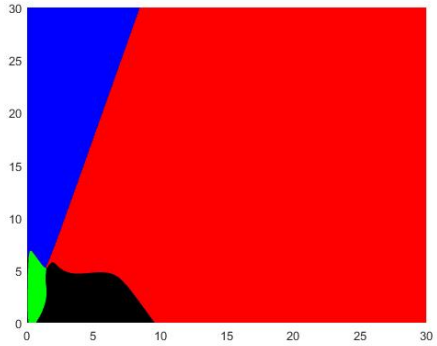

Fig. 7: MLP NN-AGPSO classification region for AGPOS1 for interference-free (green), multi-path (black), spoofing (red), and jamming (blue).

Fig. 7 illustrates the AGPSO1 classification result. Table 2 shows classification results by applying weights trained by MLP NN-AGPSO to the classifier for categorizing the received signals into spoofing, jamming, multipath, or interference-free groups based on their power and correlation distortion. These results show that autonomous group PSO can improve the classification accuracy in contrast with other algorithms mentioned in Tables 3, 4, and 5 in some detection of interference classes.

Table 4 shows the result of the Bayes-optimal classifier reported in [1]. Comparing our proposed method and Table 2 shows that AGPSO1, AGPSO2, and AGPSO3 has resulted in about 4.7%, 6.3%, and 3% more accuracy in detecting spoofing interference respectively. Also, in the detection of multi-path and jamming interference, we improved the result by approximately 95% and 1.6% respectively by employing AGPSO3. The same analysis can be applied in comparison to our proposed method with the results in Tables 3 and 5.

Table 2: Simulation-evaluated by MLP NN-AGPSO classifier.

| **AGPSO** | **Decision** | **True decision** | | | |
|---|---|---|---|---|---|
| | | H0 | H1 | H2 | H3 |
| **AGPSO1** | H0 | 0.8317 | 0.4772 | 0.0261 | 0.0083 |
| | H1 | 0.1550 | 0.4733 | 0.0272 | 0.0006 |
| | H2 | 0.0006 | 0.0383 | 0.9028 | 0.0089 |
| | H3 | 0.0128 | 0.0111 | 0.0439 | 0.9822 |
| **AGPSO2** | H0 | 0.9895 | 0.8210 | 0.0347 | 0.0147 |
| | H1 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | H2 | 0.0062 | 0.1757 | 0.9330 | 0.0110 |
| | H3 | 0.0043 | 0.0033 | 0.0323 | 0.9743 |
| **AGPSO3** | H0 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | H1 | 0.9711 | 0.9533 | 0.0522 | 0.0056 |
| | H2 | 0.0011 | 0.0306 | 0.9011 | 0.0078 |
| | H3 | 0.0278 | 0.0161 | 0.0467 | 0.9867 |

Table 3: Simulation-evaluated classification reported in [12].

| **Decision** | **True decision** | | | |
|---|---|---|---|---|
| | H0 | H1 | H2 | H3 |
| **H0** | 0.9947 | 0.9083 | 0.0670 | 0.0039 |
| **H1** | 0 | 0.0698 | 0.0117 | 0 |
| **H2** | 0.0043 | 0.0214 | 0.8463 | 0.0155 |
| **H3** | 0.0010 | 0.0005 | 0.0750 | 0.9806 |

Table 4: Simulation-evaluated classification reported in [1].

| **Decision** | **True decision** | | | |
|---|---|---|---|---|
| | H0 | H1 | H2 | H3 |
| **H0** | 0.9942 | 0.8809 | 0.0624 | 0.0184 |
| **H1** | 0.0005 | 0.0987 | 0.0234 | 0 |
| **H2** | 0.0001 | 0.0162 | 0.8698 | 0.0017 |
| **H3** | 0.0039 | 0.0028 | 0.0442 | 0.9799 |

Table 5: Simulation-evaluated classification reported in [20].

| Decision | True decision | | | |
|---|---|---|---|---|
| | H0 | H1 | H2 | H3 |
| **H0** | 0.8794 | 0.5004 | 0.0427 | 0.0457 |
| **H1** | 0.1201 | 0.4751 | 0.0511 | 0.0047 |
| **H2** | 0.0001 | 0.0242 | 0.8871 | 0.0056 |
| **H3** | 0.0002 | 0.0001 | 0.0190 | 0.9437 |

H0, H1, H2, H3 are interference-free, multi-path, spoofing, and jamming, respectively.

## 7. Conclusion

In this paper, we utilized a so-called MLP NN-AGPSO classifier for the detection and classification of GNSS possible interferences. To assess the designed classifier, after performance analysis, the obtained results were compared with methods in [1], [12], and [20]. According to the results, we can conclude that the AGPSO algorithm can lead us to better results in terms of classification accuracy compared with the other methods' benchmarks. The results show that multi-path detection has been improved about 95%, and spoofing and jamming detection are roughly 3% and 1% more precise compared with the results in [1].

## 8. References


[1] K. D. Wesson, J. N. Gross, T. E. Humphreys, and B. L. Evans, "GNSS Signal Authentication via Power and Distortion Monitoring," IEEE Transactions on Aerospace and Electronic Systems, vol. 54, no. 2, pp. 739-754, 2017.

[2] M. Mosavi, Z. Nasrpooya, and M. Moazedi, "Advanced Anti-spoofing Methods in Tracking Loop," Journal of Navigation, vol. 69, no. 4, pp. 883-904, 2016.

[3] X. Li and J. Sun, "Genetic Algorithm-Based Multi-Objective Optimization for Statistical Yield Analysis Under Parameter Variations," Journal of Circuits, Systems and Computers, vol. 26, no. 01, pp. 1-21, 2017.

[4] K. Muthulakshmi, R. M. Sasiraja, and V. S. Kumar, "The Proper Location and Sizing of Multiple Distributed Generators for Maximizing Voltage Stability using PSO," Journal of Circuits, Systems and Computers, vol. 26, no. 04, pp. 1-20, 2017.

[5] M. R. Mosavi and M. Khishe, "Training a Feed-Forward Neural Network using Particle Swarm Optimizer with Autonomous Groups for Sonar Target Classification," Journal of Circuits, Systems and Computers, vol. 26, no. 11, pp. 175-185, 2017.

[6] K. Socha and C. Blum, "An Ant Colony Optimization Algorithm for Continuous Optimization: Application to Feed-Forward Neural Network Training," Neural Computing and Applications, vol. 16, no. 3, pp. 235-247, 2007.

[7] M. Moazedi, M. R. Mosavi, and A. Sadr, "Real-time Interference Detection and Mitigation in Robust Tracking Loop of GPS Receiver," Analog Integrated Circuits and Signal Processing, vol. 95, no. 1, pp. 93-113, 2018.

[8] A. Ndili and P. Enge, "GPS Receiver Autonomous Interference Detection," IEEE 1998 Position Location and Navigation Symposium (Cat. No.98CH36153), Palm Springs, CA, USA, pp. 123-130, 1998.

[9] F. Bastide, D. Akos, C. Macabiau, and B. Roturier, "Automatic Gain Control (AGC) as an Interference Assessment Tool," in Proceedings of the 16th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS/GNSS 2003). Portland, OR: Institute of Navigation, September 2003.

[10] D. M. Akos, "Who's Afraid of the Spoofer? GPS/GNSS Spoofing Detection via Automatic Gain Control (AGC)," Journal of the Institute of Navigation, vol. 59, no. 4, Winter 2012.

[11] N. Blanco-Delgado and F. D. Nunes, "Multipath Estimation in Multicorrelator GNSS Receivers using the Maximum Likelihood Principle," IEEE Transactions on Aerospace and Electronic Systems, vol. 48, no. 4, pp. 3222-3233, 2012.

[12] J. N. Gross, C. Kilic, and T. E. Humphreys, "Maximum-Likelihood Power Distortion Monitoring for GNSS-signal Authentication," IEEE Transactions on Aerospace and Electronic Systems, vol. 55, no. 1, pp. 469-475, 2019.

[13] C. Ozturk and D. Karaboga, "Hybrid Artificial Bee Colony Algorithm for Neural Network Training," 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, pp. 84-88, 2011.

[14] M. Yaghini, M. Khoshraftar, and M. Fallahi, "A Hybrid Algorithm for Artificial Neural Network Training," Engineering Applications of Artificial Intelligence, vol. 26, no. 1, pp. 293-301, 2013.

[15] A. Kattan, R. Abdullah, and R. A. Salam, "Harmony Search Based Supervised Training of Artificial Neural Networks," In 2010 International Conference on Intelligent Systems, Modelling and Simulation (ISMS), Liverpool, pp. 105-110, 2010.

[16] R. C. Eberhart and J. Kennedy, "A New Optimizer Using Particles Swarm Theory," In Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39-43, 1995.

[17] R. C. Eberhart and J. Kennedy, "Particle Swarm Optimization," In IEEE International Conference on Neural Network, Perth, Australia, pp. 1942-1948, 1995.

[18] S. Mirjalili and A. Lewis, "S-shaped Versus V-shaped Transfer Functions for Binary Particle Swarm Optimization," Swarm and Evolutionary Computation, vol. 9, pp. 1-14, 2013.

[19] X. Li and J. Sun, "Genetic Algorithm-Based Multi-Objective Optimization for Statistical Yield Analysis Under Parameter Variations," Journal of Circuits, Systems and Computers, vol. 26, no. 1, pp. 1-21, 2017.

[20] S. Tohidi and M. R. Mosavi, "Effective Detection of GNSS Spoofing Attack Using A multi-layer Perceptron Neural Network Classifier Trained by PSO," in Proceedings of IEEE Computer Conference, ComputerSociety of Iran (CSICC), pp. 1-5, 2020.