# Logarithm Decreasing Inertia Weight Particle Swarm Optimization Algorithms for Convolutional Neural Network

Murinto[1], Miftahurrahma Rosyda[2]

[1,2]*Informatics Department, Universitas Ahmad Dahlan Yogyakarta, Indonesia*

[1]murintokusno@tif.uad.ac.id, [2]miftahurrahma.rosyda@tif.uad.ac.id

**Abstract - The convolutional neural network (CNN) is a technique that is often used in deep learning. Various models have been proposed and improved for learning on CNN. When learning with CNN, it is important to determine the optimal parameters. This paper proposes an optimization of CNN parameters using logarithm decreasing inertia weight (LogDIW). This paper is used two datasets, i.e., MNIST and CIFAR-10 dataset. The MNIST learning experiment, the CIFAR-10 dataset, compared its accuracy with the CNN standard based on the LeNet-5 architectural model. When using the MNIST dataset, CNN's baseline was 94.02% at the 5th epoch, compared to CNN's LogDIWPSO, which improves accuracy. When using the CIFAR-10 dataset, the CNN baseline was 28.07% at the 10th epoch, compared to the LogDIWPSO CNN accuracy of 69.3%, which increased the accuracy.**

**Keywords: CIFAR-10, CNN, logarithm decreasing inertia weight, MNIST**

## I. INTRODUCTION

Deep learning is a set of algorithms in machine learning that tries to learn at several levels, according to different levels of abstraction. This is usually done using an artificial neural network (ANN). The levels in the statistical model studied correspond to different concept levels, where higher-level concepts are defined from lower-level concepts, and lower-level concepts can help define some higher-level concepts. Deep learning is one of the fields of research that works based on artificial intelligence (AI). According to [1] , deep learning can be grouped into 3 models, namely: discriminatory models, generative models and hybrid models. The discriminatory model consists of deep neural network (DNN), convolutional neural network (CNN) and recurrent neural network (RNN). Generative models include deep belief networks (DBN), restricted Boltzmann machines etc. While the hybrid model is a structure that combines identification models and generative models.

In several studies such as that conducted by [2] who proposed an algorithm based on particle swarm optimization, to automatically perform searches to be applied to the deep convolutional neural network architecture for image classification, which is called psoCNN. However, when learning to use machine learning such as CNN, it is important to determine the optimal hyperparameters. There are many parameters in deep learning, and it is difficult to determine the optimal value manually, so some studies are trying to do it automatically to determine the optimal value.

One of the known automation methods is the metaheuristic algorithm. Some metaheuristic algorithms as an automation method. Metaheuristic algorithms are a way to solve difficult optimization problems, and in recent years they have been used for CNN optimization [3-6]. Algorithms are inspired by nature and are based on animal behavior, physics, biology, and so on. This algorithm is based on biological phenomena including evolutionary strategies (ES) and genetic algorithms (GA) [7]. Meanwhile, those based on physical phenomena include simulated annealing (SA) [8]. While based on animal behavior, among others: Ant Colony Optimization (ACO) [9], Firefly Algorithm (FA) [10], Bat Algorithm (BA) [11], and Particle Swarm Optimization (PSO) [12].

For some of these metaheuristic algorithms, PSO is widely used for network optimization. PSO has good convergence when compared to GA. Several studies that use PSO for CNN optimization include the research of [13] proposed a hybrid algorithm in which particle swarm optimization (PSO) is used to reduce the complexity of the overall algorithm. PSO used in conjunction with CNN will reduce the number of epochs in the training process and its dependence on the GPU system. The proposed algorithm can increase the accuracy by 3-4% with a smaller number of epochs. Besides that, using this algorithm is able to overcome the local minima problem which is a common problem in the backpropagation training methodology. PSO is also used

in image classification along with CNN, referred to as psoCNN. Where a new strategy of direct coding and operator speed is designed to allow optimization of the use of PSO with CNN. The experimental results show that psoCNN can quickly find the CNN architecture well [14] Likewise, CNN's PSOs were also combined as done by [14-15]. One of the important things in PSO is the inertia weight used. The inertia weight parameter is important for optimization. Some of them are inertia weights that have been introduced in research [16-21]. In this study, the logarithm decreasing inertia weight (LogDIW) of PSO [22] was used to optimize the CNN hyperparameter. The aim was to improve the accuracy of the MINST, CIFAR-10 and Imagenet datasets, which are often used as benchmarks, using a CNN based on the LeNet-5 architecture [23]. In this study, an optimization algorithm for the convolution neural network is proposed to overcome the many parameters that exist.

This paper is structured as follows: Section 1 describes the background and previous research conducted related to metaheuristic algorithms and CNN, especially the use of PSO for network optimization. Section 2 describes the proposed method. Section 3 result and discussion. Section 4 describes the conclusion and future work.

## II. METHOD

In this study, a LogDIWPSO was used to optimize the CNN parameters and improve the accuracy obtained via CNN. Figure 1 shows a flowchart of the proposed method.

### A. *Particle Swarm Optimization*

Ref. [24] introduced the particle swarm optimization (PSO) algorithm which is a stochastic optimization technique. The basic idea of PSO is to involve a flock of birds in search of food sources in a particular area. The birds do not know for sure where the food source is. Through iteration, they will find out how far the food source can be found. The best individual strategy will be followed by the bird that is close to the food source and also from the previous best position that the individual has achieved. PSO looks for the optimal solution by updating the position and velocity of each particle through (1) and (2) [25].

$$v_{id}^{t+1} = w * v_{id}^t + c_1 * r_1 * (pBest_{id} - x_{id}^t) + c_2 * r_2 * (gBest_{gd} - x_{id}^t) \tag{1}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \tag{2}$$

Where t is the-t iteration, while d is the d-the dimension in the optimization search space. $c_1$ dan $c_2$ are acceleration constants, representation the weighting of the stochastic acceleration terms that push each particle to the best ($pBest_{id}$) dan the global best ($gBest_{gd}$). The values $r_1$ and $r_2$ are random numbers that have a uniform distribution in [0,1].

### B. *Logarithm Decreasing Inertia Weight*

From the research that has been done, it shows that PSO with a large inertial weight value (w) has a good ability in global optimal search, compared to a small inertial weight. The more inertial weights tend to have faster convergence. In his research, [22] introduced an inertia weight called the logarithm decreasing inertia weight (LogDIW) which is written as (3)

$$w = w_{max} + (w_{min} - w_{max}) * \log_{10}\left(a + \frac{10t}{T_{max}}\right) \tag{3}$$

Where a is a constant for the evolutionary velocity adjustment, here a=1. $w_{min}$ and $w_{max}$ are minimum and maximum weight, t is iteration

### C. **Convolutional Neural Network**

Several models have been proposed for the CNN architecture, including LeNet-5 [23]. In this study, LeNet-5 is used, which is one of the well-known basic architecture. The LeNet-5 architecture is shown in Fig. 1 and the proposed algorithm's flow chart is presented in Fig. 2.
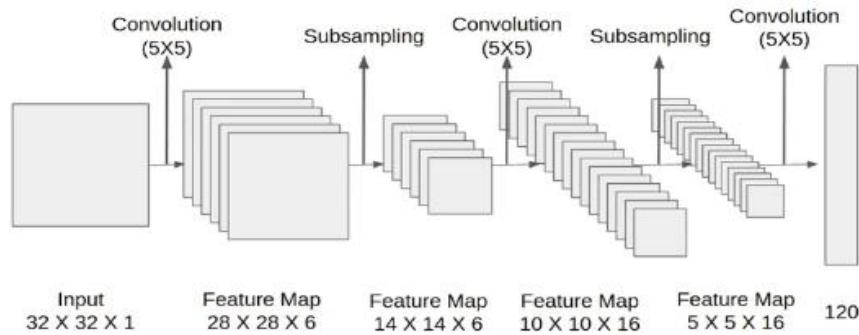
**Fig. 1 The LeNet-5 architecture [23]**

In the LeNet-5 architecture, each convolution layer consists of 3 parts: convolution, pooling and nonlinear activation functions. Convolution is used to extract spatial features and is often referred to as receptive fields originally. In the LeNet-5 architecture, there is also a subsampling average pooling layer, tanh activation function, MLP classifier, and connection space between layers to reduce the complexity of the calculation. There are 7 layers consisting of 3 convolution layers, 2 subsampling layers and 2 fully connected layers.

In the experiment, first determine the position and velocity of each particle. Next, a CNN is run for each particle, and the position, velocity, pBest and best are updated based on the results obtained. This operation is repeated until a certain time and the parameter gBest of the particle in the experiment is cross-entropy and is written as the entropy equation.

The layers in the LeNet-5 architecture as shown in Fig. 1, each consist of: Layer 1 is the input layer. The input layer can support 32 x32 pixels. Layer 2, namely layer C1 is a convolution layer with six convolution kernels measuring 5 x 5, while the size of feature allocation is 28 x 28. Layer 3 is layer S3 which is a grouping layer producing 6 function graphs with a length of 14 x 14. Each cell is a function map mapped to are 2 x 2 on the corresponding function map in C1. Layer 4 is the convolution layer of C3 which consists of 16 convolution kernels of 5 x 5 inputs derived from the 6 main function maps of C3. Layer 5 is layer S4 which is the same as S2 with size 2 x 2 and the output of sixteen function graphs of size 5 x 5. Layer 6 is layer C5 is a convolution layer with 120 core convolutions with a length of 5 x 5, the output length of C5 is 1 * 1 so that S4 and S5 are actually connected. Layer 7 is layer F6 which is linked to C5 and the resulting 84 feature charts. In this study, the activation function of each layer uses Sigmoid, ReLu and Tanh, and the batch size is optimized from 10 to 50, shown in Table I. The optimizer used is Adam or Stochastic Gradient Descent (SGD) with a learning rate of 0.01. In the implementation of the model using google colab.

The main parameters of the LogDWPSO are shown in Table II. The size of the swarm is 10 and the maximum iteration is 20. The cognitive and social parameters are both 2.0 and the weight decreases linearly from 0.9 to 0.4.
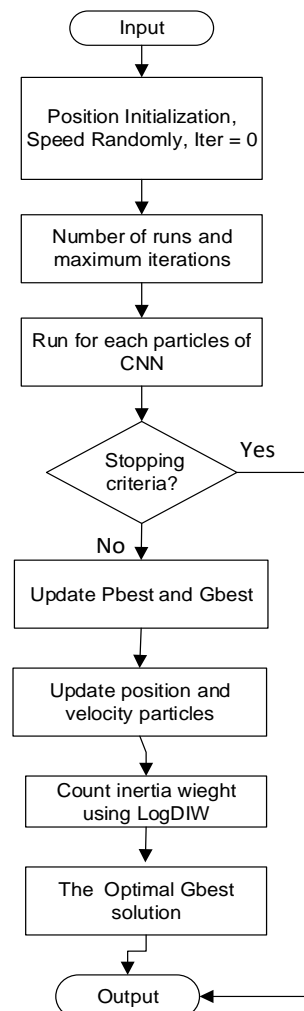


**Fig. 2 Flowchart proposed logarithmic decreasing inertia PSO CNN method**

TABLE I
PARAMETER FOR CNN

| CNN Parameters | CNN | LogDIWPSO CNN |
|---|---|---|
| Learning function | Adam | Adam |
| Activation function | Softmax | Softmax |
| Non-linear activation functions | ReLU, Tanh | ReLU, Tanh |
| Epoch | 10 | 50 |
| Batch Size | 10 | 50 |

TABLE II
PARAMETER FOR LogDIWPSO

| LogDIWPSO Parameters | Value |
|---|---|
| Iteration Number | 20 |
| Population size | 10 |
| $c_1$ | 2 |
| $c_2$ | 2 |
| $w_{max}$ | 0.9 |
| $w_{min}$ | 0.4 |

## III. RESULT AND DISCUSSION

In this research, the experiment with 2 datasets and Lenet-5 CNN architectures. The first experiment was to use the MNIST dataset and the CIFAR-10 dataset on the Lenet-5 architecture. The two image datasets are shown in Fig.3.

MNIST is an image database consisting of 600 handwritten number images for learning and 10000 for testing. Each image is given a label from 0 to 9 In this study, optimization was carried out with LogDIWPSO for every 5 epochs, and learning was displayed based on the parameters obtained. Each experiment was carried out in 30 times, and the average value was used. CIFAR-10 is a labeled subset of 80 million small image datasets. The dataset was collected by [26]. The CIFAR-10 dataset consists of 60,000 32 x 32 color images in 10 classes, with 6000 images per class. In the CIFAR-10 dataset there are 50,000 training images and 10,000 testing images. The dataset is divided into five training batches and one testing batch, each batch consisting of 10,000 images. The accuracy and variance are shown in Table III and IV. At the time of learning 10 epochs, the accuracy of CNN optimized with LogDIWPSO was 98.22%, higher than CNN without optimization which was 97.42%. In Fig.4 and Fig.5 are shown respectively accuracy and loss of CNN and LogDIWPSO CNN using MNIST –CIFAR 10 dataset.
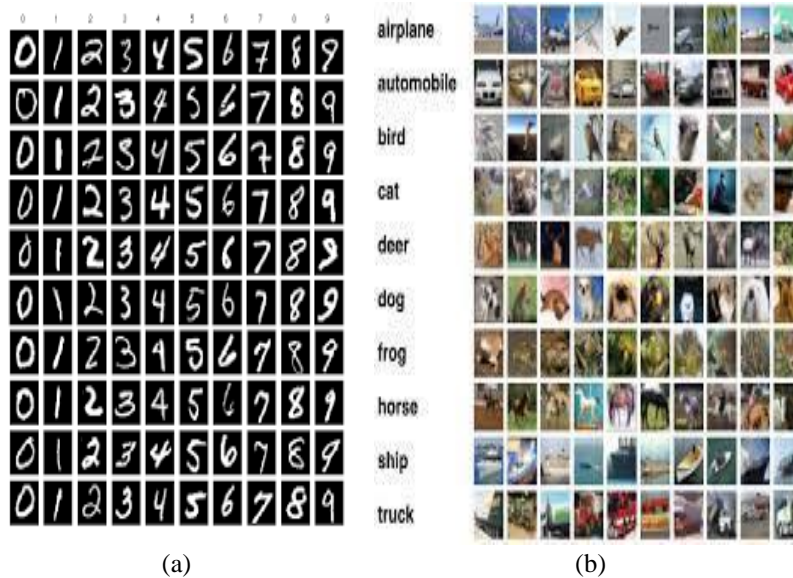


(a)  (b)

**Fig. 3 Examples of Some Image Datasets MNIST(a) [23], CIFAR-10 (b) [24]**
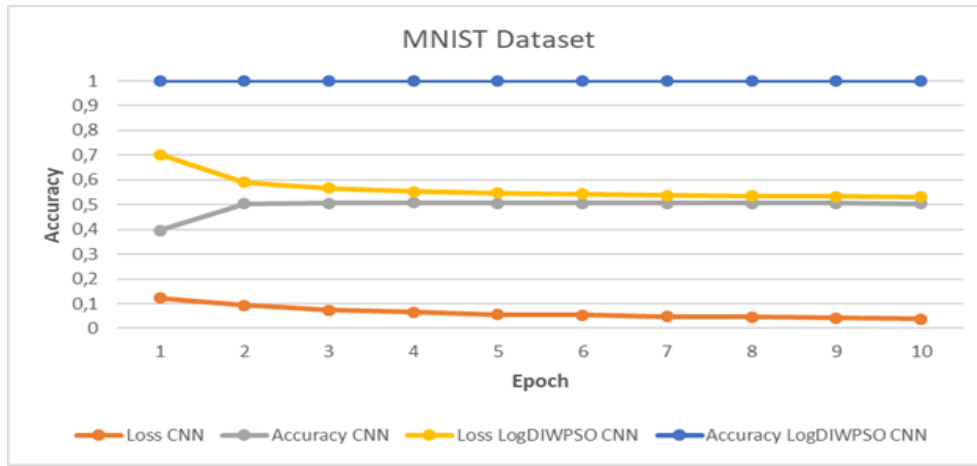
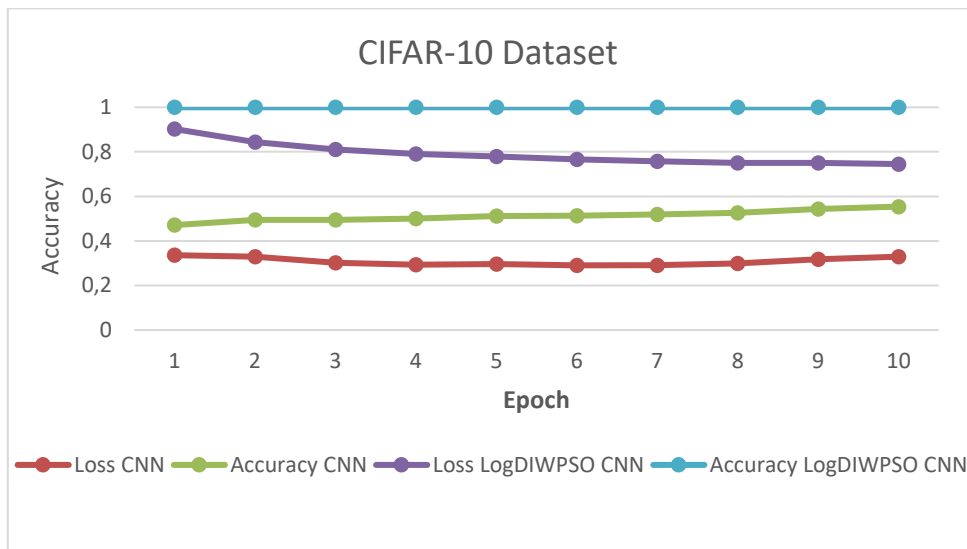**Fig. 4 Accuracy and loss of CNN and LogDIWPSO CNN using MNIST dataset**



**Fig. 5 Accuracy and loss of CNN and LogDIWPSO CNN using CIFAR-10**

TABLE III
ACCURACY AND LOSS OF CNN AND LOGDIWPSO
CNN USING MNIST DATASET
ON LENET ARCHITECTURE

| Epoch | CNN | | LogDIWPSO CNN | |
|---|---|---|---|---|
| | Loss | Accuracy | Loss | Accuracy |
| 1 | 0.4043 | 0.8913 | 1.0063 | 0.9754 |
| 2 | 0.2126 | 0.9391 | 0.1964 | 0.9385 |
| 3 | 0.1630 | 0.9520 | 0.1334 | 0.9581 |
| 4 | 0.1434 | 0.9570 | 0.1006 | 0.9682 |
| 5 | 0.1210 | 0.9626 | 0.0868 | 0.9731 |
| 6 | 0.1149 | 0.9666 | 0.0775 | 0.9762 |
| 7 | 0.1002 | 0.9698 | 0.0697 | 0.9785 |
| 8 | 0.0950 | 0.9708 | 0.0613 | 0.9810 |
| 9 | 0.0877 | 0.9736 | 0.0573 | 0.9819 |
| 10 | 0.0817 | 0.9742 | 0.0577 | 0.9822 |

TABLE IV
ACCURACY AND LOSS OF CNN AND LOGDIWPSO
CNN USING CIFAR-10 DATASET
ON LENET ARCHITECTURE

| Epoch | CNN | | LogDIWPSO CNN | |
|---|---|---|---|---|
| | Loss | Accuracy | Loss | Accuracy |
| 1 | 1.3126 | 0.5278 | 1.6841 | 0.3814 |
| 2 | 1.1726 | 0.5898 | 1.2445 | 0.5584 |
| 3 | 1.0109 | 0.6417 | 1.0575 | 0.6317 |
| 4 | 0.9451 | 0.6687 | 0.9339 | 0.6742 |
| 5 | 0.9417 | 0.6838 | 0.8498 | 0.7034 |
| 6 | 0.9048 | 0.6914 | 0.7829 | 0.7278 |
| 7 | 0.8928 | 0.6987 | 0.7310 | 0.7431 |
| 8 | 0.9114 | 0.6953 | 0.6818 | 0.7641 |
| 9 | 0.9875 | 0.7030 | 0.6463 | 0.7762 |
| 10 | 1.0222 | 0.6993 | 0.5935 | 0.7910 |

## IV. CONCLUSION

In this research, a convolutional neural network method using logarithmic decreasing inertia weight (LogDIW) for PSO has been proposed. LogDIW is used for particle swarm optimization (PSO) hyperparameter optimization. The CNN architecture used is Lenet-5 with two datasets, namely MNIST and CIFAR-10. In this paper is used two datasets, i.e., MNIST and CIFAR-10 dataset. The MNIST learning experiment, the CIFAR-10 dataset, compared its accuracy with the CNN standard based on the LeNet-5 architectural model. When using the MNIST dataset, CNN's baseline was 94.02% at 5th epoch, compared to CNN's LogDIWPSO, which improves accuracy. When using the CIFAR-10 dataset, the CNN baseline was 28.07% at the 10th epoch, compared to the LogDIWPSO CNN accuracy of 69.3% which increased the accuracy.

The next research is to apply the proposed method for different classification of new datasets. The effectiveness of the proposed method is seen in the image classification process. In addition, it is necessary to analyze the conditions for stopping PSO if it is applied in the classification process. Different architectures from those used in this study can also be considered for future research.

## ACKNOWLEDGEMENT

## REFERENCES

[1] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2013.

[2] F. E. Fernandes Junior and G. G. Yen, "Particle swarm optimization of deep neural networks architectures for image classification," *Swarm Evol. Comput.*, vol. 49, no. June, pp. 62–74, 2019.

[3] A. A. Ahmed and S. M. Darwish, "A Meta-Heuristic Automatic CNN Architecture Design Approach Based on Ensemble Learning," *IEEE Access*, vol. 9, pp. 16975–16987, 2021.

[4] B. Wang, B. Xue, and M. Zhang, "Particle Swarm optimisation for Evolving Deep Neural Networks for Image Classification by Evolving and Stacking Transferable Blocks," *2020 IEEE Congr. Evol. Comput. CEC 2020 - Conf. Proc.*, 2020.

[5] Y. Wang, H. Zhang, and G. Zhang, "cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks," *Swarm Evol. Comput.*, vol. 49, pp. 114–123, 2019.

[6] V. Ayumi, L. M. R. Rere, M. I. Fanany, and A. M. Arymurthy, "Optimization of convolutional neural network using microcanonical annealing algorithm," *2016 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2016*, pp. 506–511, 2017.

[7] M. N. Alam, B. Das, and V. Pant, "A comparative study of metaheuristic optimization approaches for directional overcurrent relays coordination," *Electr. Power Syst. Res.*, vol. 128, pp. 39–52, 2015.

[8] L. M. R. Rere, M. I. Fanany, and A. M. Arymurthy, "Simulated Annealing Algorithm for Deep Learning," *Procedia Comput. Sci.*, vol. 72, pp. 137–144, 2015.

[9] M. Dorigo, T. Stutzle, and M. Birattari, "Ant Colony Optimization," *IEEE Comput. Intell. Mag.*, no. November 2006, pp. 28–39, 2006.

[10] M. O. Okwu and L. K. Tartibu, "Firefly Algorithm," *Stud. Comput. Intell.*, vol. 927, pp. 61–69, 2021.

[11] X. S. Yang, "A new metaheuristic Bat-inspired Algorithm," *Stud. Comput. Intell.*, vol. 284, pp. 65–74, 2010.

[12] L. Zhang, Q. Fu, J. Chen, H. Bai, and X. Zhou, "A modified particle swarm optimization algorithm - CPSODE," *Proc. 29th Chinese Control Decis. Conf. CCDC 2017*, vol. 1, no. 2, pp. 6659–6663, 2017.

[13] Y. Chhabra, S. Varshney, and Ankita, "Hybrid particle swarm training for convolution neural network (CNN)," *2017 10th Int. Conf. Contemp. Comput. IC3 2017*, vol. 2018-Janua, no. August, pp. 1–3, 2018.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[15] F. Wang, H. Zhang, K. Li, Z. Lin, J. Yang, and X. L. Shen, "A hybrid particle swarm optimization algorithm using adaptive learning strategy," *Inf. Sci. (Ny).*, vol. 436–437, pp. 162–177, 2018.

[16] F. Qian, M. R. Mahmoudi, H. Parvin, K. H. Pho, and B. A. Tuan, "An Adaptive Particle Swarm Optimization Algorithm for Unconstrained Optimization," *Complexity*, vol. 2020, 2020.

[17] G. Ardizzon, G. Cavazzini, and G. Pavesi, "Adaptive acceleration coefficients for a new search diversification strategy in particle swarm optimization algorithms," *Inf. Sci. (Ny).*, vol. 299, no. December, pp. 337–378, 2015.

[18] H. Cao, N. Feng, and J. Yang, "Improved Chaos Particle Swarm Optimization Algorithm For Wireless Sensor Networks Node Localization," *Revista de la Facultad de Ingenieria,* vol. 32, no. 4, pp. 259–267, 2017.

[19] M. Li *et al.*, "Particle Swarm Optimization Algorithm Based on Chaotic Sequences and Dynamic Self-Adaptive Strategy," *Journal of Computer and Communications* ,

pp. 13–23, 2017.

[20] E. Çomak, "A particle swarm optimizer with modified velocity update and adaptive diversity regulation," *Expert Syst.*, vol. 36, no. 1, pp. 1–18, 2019.

[21] D. Yan, Y. Lu, M. Zhou, S. Chen, and D. Levy, "Empirically characteristic analysis of chaotic PID controlling particle swarm optimization," *PLoS One*, vol. 12, no. 5, pp. 1–24, 2017.

[22] Y. L. Gao, X. H. An, and J. M. Liu, "A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation," *Proc. - 2008 Int. Conf. Comput. Intell. Secur. CIS 2008*, vol. 1, pp. 61–65.

[23] Y. LECUN, S. BOTTOU, Y. BENGIO, and P. HAFFNE, "Gradient-Based Learning Applied to Document Recognition," *Proc. IEEE,* vol. 86, no. 1, pp. 2278–2322, 1998.

[24] E. R. Kennedy J, "Particle swarm optimization," *IEEE Int. Conf. Neural Networks*, pp. 1942–8., 1995.

[25] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," *Proc. IEEE Conf. Evol. Comput. ICEC*, no. February, pp. 69–73, 1998.

[26] A. Krizhevsky, N. Vinod, and H. G. Hinton, "CIFAR-10 Dataset", 2021, *https://www.cs.toronto.edu*. [Accessed: 10-Jun-2021].