➔ *Regular Research Paper – NS*

A New Approach for Determining Hyperparameters in Artificial Neural Networks: Enhanced Black Hole Optimization Algorithm

Mehmet BILEN
Mehmet Akif Ersoy University, Golhisar School of Applied Science, Turkey
mbilen@mehmetakif.edu.tr

Abstract

Artificial Neural Networks (ANN), a simple copy of the neurons in our brain, have been used for many years to bring people's problem-solving skills to computers. Although we have been able to run these networks faster with the help of developing technology, our need to run them better has created a challenging study area. Determining the hyper-parameters of the ANN has critical importance in this direction. In this study, an approach was inspired by Black Holes to determine the ANN hyperparameters. The Black Hole Algorithm (BHA), which is used as an optimization algorithm in the literature, is used for determining the hyperparameters of the ANN. The disadvantages of the BHA algorithm were identified and enhanced, a new approach called the Enhanced Black Hole Algorithm (EBHA) was proposed to the literature. Performance values obtained by the test processes because the parameters selected with this algorithm are compared with other algorithms frequently used in the literature, it has been seen that the developed method achieves the most successful performance values.

Keywords: *hyperparameter optimization, ann, black hole algorithm, hepatitis, random search, grid search, Bayesian optimization*

1. INTRODUCTION

Artificial intelligence algorithms that work intuitively have high computational costs. This difficulty has negatively affected the number and success of studies in the field of artificial intelligence in the past century. Today, processors that reach higher speeds with the developing technology have started to be used in the development and training of artificial intelligence algorithms. Thus, there has been a significant increase in the number of successful studies in this field. Although this situation contributes to the studies carried out to determine the hyperparameters of algorithms such as ANN with black box architecture, the repetitive and complex structure of the ANN makes it very difficult to determine the best ones among the hyperparameter space, which has an infinite number of possibilities.

Hyperparameters are the parameters that manage the training process of ANN, as in many heuristic algorithms. Therefore, determining the best combination of parameters is critical for maximizing the potential of the network. In the network development process, it is not very efficient and successful for the developer to manually change and observe these parameters and try to obtain the most appropriate parameters. Because it can only observe many parameter combinations. In this context, many different approaches in the literature such as Grid Search (GS), Random Search (RS), Bayesian Optimization (BO), Black Hole Optimization Algorithm (BHA) to determine the most suitable parameters with the least computational cost. has been suggested.

In the second part of this study, the hyperparameters of ANN and the details of the studies in the literature are mentioned, and then the Enhanced Black Hole Algorithm (EBHA) developed to contribute to the literature is presented. In the title of Research Findings, the algorithms commonly used in the literature and the performance values obtained with the developed method are shared and the results are compared. In the last title, the study was concluded by giving the results when all the findings were examined.

2. MATERIAL AND METHODS

2.1. ANN and Hyperparameters

Artificial neural networks are computer programs that imitate people's problem-solving skills, inspired by biological nerves in the human brain, and are thus used to solve many problems that people can easily solve in daily life [1]. The structure of a simple neural network is given in Figure 1.

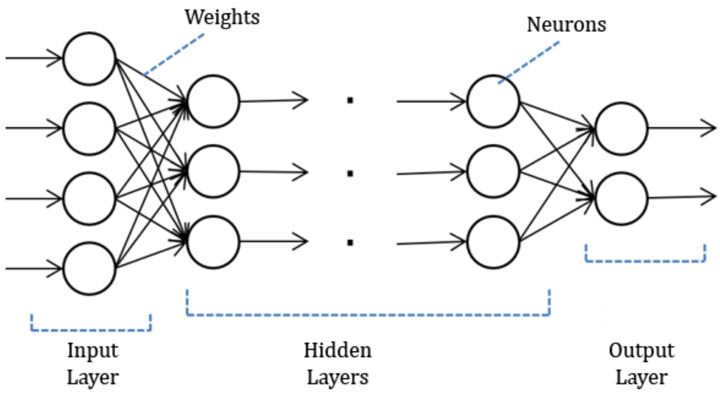

Figure 1. Structure of ANN

When the structure in the given figure is examined, it is seen that a simple ANN network consists of a series of layers. The number of neurons in the output layers is determined depending on the type and structure of the data. However, there is no clear approach in the literature to determine the number of layers in the input layer and intermediate layers, and the number of nerves in each layer. In addition, the determination of many parameters such as the approach to be used when learning the data by the weights between layers and neurons, the number of repetitions that the network will make during the learning phase, are subjects of study in itself. In this study, the Activation Function, Learning Rate, Number of Neurons in the Input Layer, Number of Hidden Layers, Number of Neurons in the Hidden Layers and Batch Size are the parameters tried to be determined by hyperparameter optimization algorithms.

Activation Function: Activation function is the method that determines how the data reaching a neuron in ANN networks will be processed and how it will produce an output based on this data [2]. In this study, it was tried to determine the activation function, which works in harmony with other parameter combinations and provides the best contribution to the performance of the network, among the Relu [3], SoftMax [4] and TanH [5] activation functions that are frequently used in the literature. The equations used to calculate the Relu, SoftMax and TanH activation functions are given in Eq [1-3], respectively.

Eq. 1.

$$relu(x) = \begin{cases} 0 \; if \; x \leq 0 \\ x \; if \; x > 0 \end{cases}$$

Eq. 2.
$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

Eq. 3.
$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Learning Rate: For learning to take place at each step of the ANN, the weights between the neurons on the network must be updated. The learning rate is used to determine the sensitivity of this update. If this rate is chosen too high, it may be possible to miss the optimum points by the network, and if it is chosen too low, the learning progresses very slowly, and the optimum point is never reached.

The number of Layers and Neurons: There are two points to be considered when determining the number of layers and neurons to be used in the network. Similar to the learning rate, if very few layers and neurons are used, the network will never have enough computational power to solve the complexity depending on the data. In cases where too many neurons and layers are used, a network with a very high computational cost will emerge and this cost will make it difficult to reach the number of repetitions required for the network to learn.

Batch Size: In traditional neural networks, updating the weights occurs because of completing a repetition by evaluating each data sample. However, depending on the size of the data, the data set can be divided into subsets and the learning can contribute positively to the performance of the network after examining the samples in each set. Still, determining the size of these subsets requires evaluation in terms of performance, cost, and efficiency, as in other parameters.

2.2. Grid Search

Grid search (GS) is one of the most preferred hyperparameter optimization methods in the literature [6-7]. GS simply tries to obtain the best result by testing the probability pool created by the Cartesian product of the predefined parameters one by one. Since predefined and specific points are investigated, the probability of reaching the optimum point at once with the GS algorithm is low (Figure 2). For this reason, it is better to determine the best of the first researched points and to create a narrower guide that explores these points and repeat this process. However, in cases where the number of possibilities resulting from Cartesian multiplication is high and this algorithm is run repeatedly, the computational cost is very high, which is one of the biggest disadvantages of this algorithm.

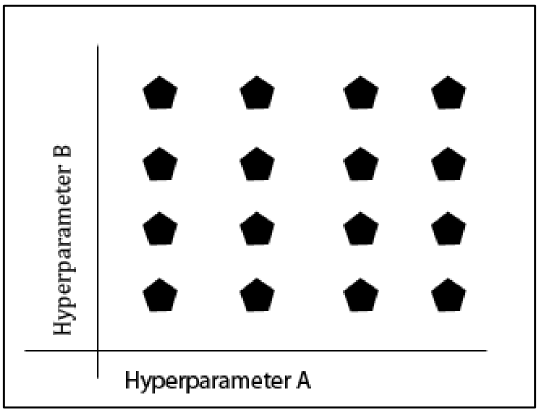

Figure 2. Hyperparameter solution space of GS

2.3. Random Search

Random search has entered the literature to overcome the disadvantage of the guide search algorithm with a large number of possibilities and to increase the sensitivity of the points to be investigated in the solution space by going beyond certain points [8] (Figure 3). The algorithm measures the success of random locations determined in the solution space. Since the determination of these locations is completely random, the probability of obtaining successful results in problems with a very large solution space is very low.

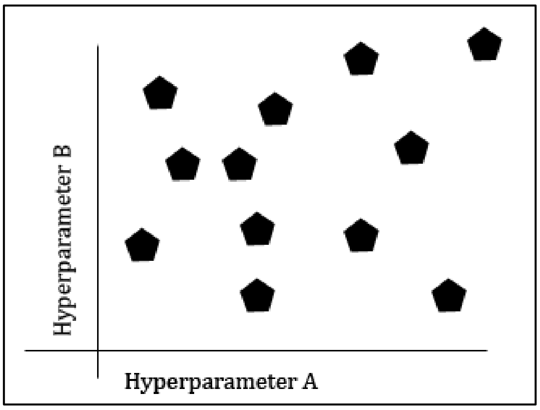

Figure 3. Hyperparameter solution space of RS

2.4. Bayesian Optimization

The biggest problem with the GS and RS methods is that historical information is ignored when searching [9]. Algorithms, such as Bayesian optimization, that use iterative and historical information, make the best decision about which parameter combination to investigate in the next iteration [10]. Eq. 4. was used to calculate the parameters that the Bayes algorithm will choose in each next iteration.

Eq. 4.
$$I(x) = max\{0, f_{t+1}(x) - f(x^+)\}$$

While the f function expresses the success achieved by the ANN working with x parameters, x^+ that is the method used to determine the next parameters is given in Eq. 5.

Eq. 5.
$$x^+ = argmax_{x_i \in x_{1:t}} f(x_i)$$

Eq. 6. was used to calculate the probability of obtaining better results with the selected parameters in the next iteration of the BO algorithm [11].

Eq. 6.
$$f(I) = \frac{1}{\sqrt{2\pi}\sigma(x)} exp(-\frac{(\mu(x) - f(x^+) - I)^2}{2\sigma^2(x)})$$

In the equation, μ represents the mean and σ represents the standard deviation.

2.5. Black Hole Optimization

Black Hole Algorithm is one of the swarm-based optimization algorithms. Similar to other swarm-based algorithms, the BHA works with the principle of searching the solution space of many individuals. However, unlike other swarm-based algorithms, individuals in the population (possible solution sets) aim to reach better solutions by moving around the best candidates in each iteration [12]. In this algorithm, which is inspired by the effect of real black holes in our cosmic universe on other space objects, black holes swallow stars that get very close to them or lose their black hole feature when another star with a better performance turns into a black hole. The pseudo-code of the working principle of BHA is given in Table 1.

Table 1. Psuedo code of BHA

1	Generate stars at random points in the solution space.
2	Repeat until a stop criterion is met
3	Calculate the fitness value of each star using the objective function (Eq. 7.)
4	Assign the star with the best fitness value as a black hole
5	Calculate new positions of all-stars (Eq. 8.)
6	If a star gets a better fitness value, replace it with a Blackhole.
7	If a star falls below the event horizon of the black hole, delete it and create a new star at a random location. (Eq. 9).
8	End the loop if the maximum number of iterations or the desired fitness value is reached.

The objective function used to calculate the fitness value of the stars in step 3 varies according to the problem to be optimized. In this study, since it is aimed to improve the performance of the ANN network, the calculation given in Eq. 7., which is inversely proportional to the ANN error, is used.

Eq. 7.
$$A(x_i) = D(x_i, YSA)$$

x_i are the hyperparameters selected in i-th iteration, D shows the classification success rate of the network trained using these parameters. After calculating the fitness value of all-stars and determining the black hole, the calculation method used to move other stars to other points in the solution space by moving with the gravitational force of the black hole is given in Eq. 8.

Eq. 8.
$$x_i(t+1) = x_i(t) + b * \left(x_{KD} - x_i(t)\right), \qquad i = 1,2, \dots, N$$

x_{KD} is the position of the black hole, $x_i(t)$ is represents the current position of the i-th star to be calculated. b represents a coefficient that is randomly determined in each calculation between 0 and 1. Eq. 9. is used to determine the diameter of the event horizon of the black hole.

Eq. 9.
$$R = \frac{A(x_{KD})}{\sum_{i=1}^{N} A(x_i)}$$

2.6. Enhanced Black Hole Algorithm

When other algorithms used for optimizing hyperparameters are examined, it is seen that it contributes to the performance of ANN successfully. Despite this, it is thought that there is more work to be done in this area. Increasing this success and contributing to the literature constitutes the main motivation of this study. For this reason, in this study, it is desired to eliminate the weaknesses of the Black Hole algorithm by making some reinforcements, thus revealing a more successful algorithm than both the Traditional BHA and other algorithms in the literature.

In the BHA, a star replaces the black hole with more successful results, and then other stars are affected by this black hole and forget the previous possible solution. In this direction, a new approach has been proposed in the literature to search for older possible solutions by giving memory to the stars. In this approach, which was created with the principle that more than one black hole affects the stars at the same time, a new calculation method is proposed as an alternative to the calculation given in Eq. 8. to determine the effective rates of black holes on the stars (Eq. 10). Eq. 11 is used to determine which star will turn into a black hole or replace it with an existing black hole. The new pseudo-code of the EBHA developed in this direction, different from the BHA, has been updated as given in Table 2.

Table 2. Psuedo code of EBHA

1	Generate stars at random points in the solution space.
2	Repeat until a stop criterion is met
3	Calculate the fitness value of each star using the objective function (Eq. 7.)
4	Assign the star with the best fitness value as a black hole
5	Calculate new positions of all-stars relative to existing black holes (Eq. 10)
6	If a star gets a better fitness value; convert the star to a new black hole if it is far enough from the event horizon otherwise replace it with the existing black hole (Eq. 11.).
7	If a star falls below the event horizon of the black hole, delete it and create a new star at a random location. (Eq. 9).
8	End the loop if the maximum number of iterations or the desired fitness value or the maximum number of the black hole is reached.

Eq. 10

$$x_i(t+1) = x_i(t) + \sum_{j=1}^{m} b * A\left(x_{KD_j}\right) * \left(x_{KD_j} - x_i(t)\right)$$

Eq. 11.

$$S(x_i) = \begin{cases} replace\ if\ u(x_{KD}, x_i) \le 2R) \\ convert\ if\ u(x_{KD}, x_i) > 2R) \end{cases}$$

u is a function that is used to calculate the distance between the black hole and the star. The calculation method of this function, which works with the Euclidean distance calculation principle, is given in Eq. 12.

Eq. 12.

$$u(Y, Z) = \sqrt{\sum_{i=1}^{n} (Y_i - Z_i)^2}$$

In the case of black hole and star whose distance between Y and Z is desired to be measured, the parameters of Y are indicated by Y_i, the parameters of Z are indicated by Z_i. n gives the total number of parameters.

4. FINDINGS

Along with the proposed method, different approaches that have an important place in the literature have been used to determine the hyperparameters of ANN. Algorithms are coded on the Google Colab Notebook platform using the Python programming language. Testing was carried out with an Intel Xeon processor with a dual-core and 2.20 GHz operating frequency allocated by the same platform.

For the classification problem, the "Hepatitis" dataset was used in the training and testing processes of the ANN algorithm, which is required to determine the hyperparameters [13-14]. The dataset contains 19 different biological information obtained from 155 different people with liver disease and 1 class parameter (0:Dies, 1:Alives) that indicates the future survival or death of the person concerned. Since the biological information has different size ranges from each other, all the features have been scaled by being drawn to the range of 0 and 1.

The solution spaces investigated with the GS, RS, BO, BHA, EBHA algorithms are given in Table 3.1. Since there are too many possibilities in the GS algorithm, combinations of fixed parameters were searched instead of choosing a range. Even in this case, 1620 possibilities were evaluated, more than the number of networks trained and tested in other algorithms. For the BHA and EBHA algorithms, a population of 5 stars was created and run for 20 iterations, and the possibilities were tested for 100 different networks.

Table 3. Hyperparameter space

Algorithm	Learning Rate	Activation Function	Neuron Count in the input layer	Hidden Layer count	Neuron count in the hidden layers	Batch Size	Trained ANN count
GS	0.0001, 0.0003, 0.0005, 0.0009	Relu, SoftMax, TanH	50, 250, 500	2-6	50, 250, 500	16,32,64,128	1620
RS, BO, BHA, GBHA	0.0001 – 0.001	Relu, SoftMax, TanH	10-500	2-10	10-500	2-155	100

The hyperparameter/solution spaces given in Table 3. were investigated with the relevant algorithms and used in ANN training. 80% of the dataset was reserved for training and the remaining 20% was used for testing. The most successful parameters obtained for each algorithm and the accuracy values obtained as a result of training and testing the ANN algorithm with these parameters are given in Table 4. The accuracy value was calculated through Eq 13. using the values obtained after the classification of the samples allocated for testing.

Eq. 13.

$$D(x_i, ANN) = \frac{T}{n}$$

The T in the equation represents the number of successfully classified samples and n the total number of samples.

Table 4. Best hyperparameters which determined by each algorithm

Algorithm	Learning Rate	Activation Function	Neuron Count in the input layer	Hidden Layer count	Neuron count in the hidden layers	Batch Size	Accuracy
GS	0,0009	Relu	50	4	500	16	0,8387
RS	0,000749	Relu	79	9	470	46	0,8065
BO	0,0001	Relu	348	10	279	3	0,8387
BHA	0.0008	SoftMax	34	5	285	73	0,8065
EBHA	0,00031	Relu	118	4	103	10	**0,8709**

When the parameters obtained because of the optimization algorithms and ANN training are examined, it is seen that the EBHA algorithm is more successful than both the traditional BHA and other algorithms in the literature. While the GS algorithm can select parameters that can obtain an accuracy value of 0.8387 by testing 1620 different networks, the EBHA achieved a higher success value with only 5 different stars and 100 repetitions. When the EBHA algorithm is compared with the RS and BO algorithms, it is seen that it achieves more successful results with fewer neurons and fewer layers.

The pair plot produced by using all parameters and accuracy values selected with different algorithms is given in Figure 4.

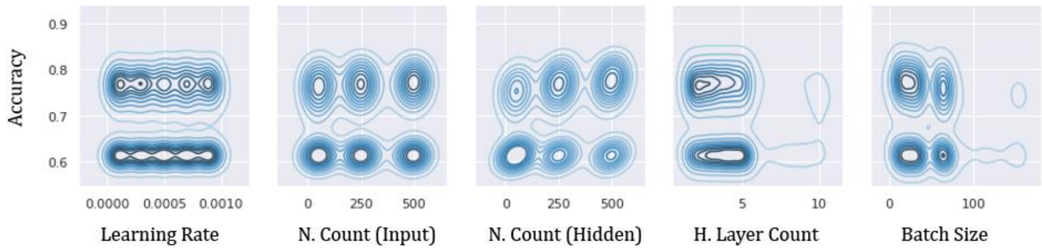

Figure 4. Pair plot of hyperparameters and accuracy values

When the pair plot is examined, it is seen that each parameter has its optimum range, but better results are produced when it comes to the most appropriate range for other parameters of this range. In other words, although it seems possible to increase the performance of the network by changing the value of only one parameter, it is necessary to obtain the best combination of all parameters to obtain the best fitness value.

The graph showing the accuracy values of the network trained using the best parameters obtained with EBHA at each iteration during the training and validation phase is given in Figure 5.

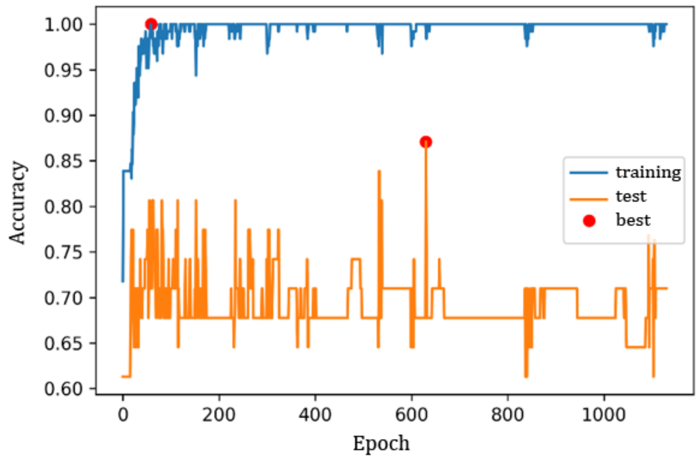

Figure 5. Training and Validation Iterations of EBHA-ANN

As seen in the training and validation graph, the trained ANN reached the maximum accuracy level in a very short time, but the highest success was achieved in the validation after 600 iterations. The findings obtained from the test procedures performed while preserving the learning obtained in this iteration are given in Table 5.

Table 5. Performance values

Class	Specificity	Sensitivity	F1-score	Sample Count
All	0.82	0.83	0.82	155
1	0.88	0.90	0.89	123
0	0.59	0.53	0.56	32

The ROC curve calculated as a result of the test process performed by ANN's classification of the samples of the Hepatitis dataset is given in Figure 6.

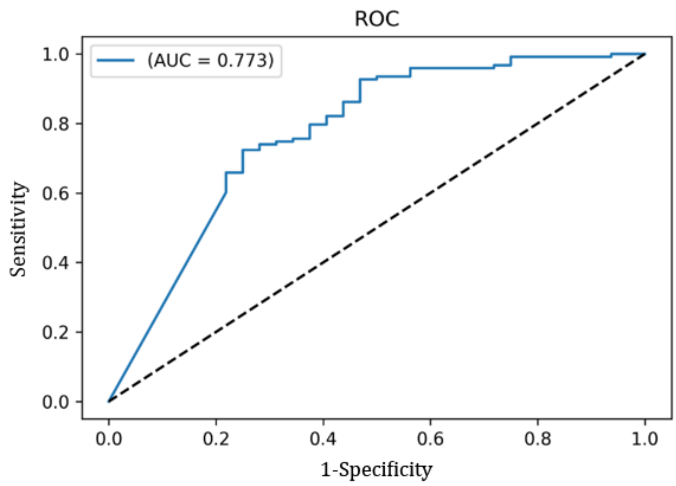

Figure 6. ROC curve of BHAE-ANN

4. CONCLUSION

In this study, for the optimization of the hyperparameters of the ANN, a new algorithm (EBHA) has been proposed to the literature by enhanced the BHA. It is seen that the proposed approach achieves better performance values than the algorithm in the literature with less computation cost and time.

The most suitable hyperparameters for training the Hepatitis dataset with ANN with the proposed EBHA algorithm were determined as Learning rate 0.00031, Activation Function as Relu, number of neurons in the input layer 118, number of hidden layers 4, number of neurons in each hidden layer 103, batch size 10. In the training process with these parameters, the highest accuracy value was 0.8709, while testing the ANN trained with these hyper-parameters revealed 0.82 Specificity, 0.83 Sensitivity and 0.82 f1 score values. When the results are examined, it is seen that the EBHA has achieved high success in both training and testing processes.

This study shows that the solution space used for the determination of hyperparameters in the literature always has the possibility of containing a more successful solution. Therefore, all studies in the literature, including the approach proposed in this study, bring researchers one step closer to designing an optimum ANN.

REFERENCES

[1] Blanton, H. (1997). An Introduction to Neural Networks for Technicians, Engineers and Other non PhDs. *Proceedings of 1997 Artificial Neural Networks in Engineering Conference.* St. Louis.

[2] Hinkelmann, K. (2021, 7 16). *Neural Networks*. University of Applied Sciences Northwestern Switzerland School of Business: Accessed on July 7, 2021, from http://didattica.cs.unicam.it/lib/exe/fetch.php?media=didattica:magistrale:kebi:ay_17 18:ke-11_neural_networks.pdf

[3] Nair, V., & Hinton, G. E. (2010, June). Rectified linear units improve restricted Boltzmann machines. *ICML'10: Proceedings of the 27th International Conference on International Conference on Machine Learning* (s. 807-814). Haifa: Omnipress, United States.

[4] Hinton, G., Deng, L., Deng, L., Yu, D., & Dahl, G. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine, 26*(6), 82-97.

[5] Goodfellow, I., Bengio, Y., & Courville, A. (2017). *Deep Learning (Adaptive Computation and Machine Learning series).* Massachusetts: Cambridge.

[6] Injadat, M., Moubayed, M., Nassif, A. B., & Shami, A. (2020). Systematic ensemble model selection approach for educational data mining. *Knowledge-Based Systems, 200*, 105992.

[7] Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing, 415*, 295-316.

[8] James, B., & Yoshua, B. (2012). Random search hyper-parameter optimization. *Journal of Machine Learning Research, 13*(2).

[9] Sameen, M. I., Pradhan, B., & Lee, S. (2019). Self-Learning Random Forests Model for Mapping Groundwater Yield in Data-Scarce Areas. *Natural Resources Research, 28*, 757-775.

[10] Sameen, M. I., Pradhan, B., & Lee, S. (2020). Application of convolutional neural networks featuring Bayesian optimization for landslide susceptibility assessment. *CATENA, 186*, 104249.

[11] Kouziokas, G. (2020). A new W-SVM kernel combining PSO-neural network transformed vector and Bayesian optimized SVM in GDP forecasting. *Engineering Applications of Artificial Intelligence, 92*, 103650.

[12] Hatamlou, A. (2013). Blackhole: A new heuristic optimization approach for data clustering. *Information Sciences, 222*, 175-184.

[13] Dua, D., & Graff, C. (2019). Irvine, CA: The University of California, School of Information and Computer Science. http://archive.ics.uci.edu/ml.

[14] *Machine Learning Repository*. (2021). Accessed on May 26, 2021, from http://archive.ics.uci.edu/ml

JOMUDE
http://www.jomude.com