

## A Modified Marine Predator Algorithm for an Efficient Feature Selection

<sup>1</sup>Abdulrazaq Ahmed Bello, <sup>2</sup>Muhammad Abdullahi, <sup>3</sup>Salisu Aliyu

<sup>1</sup>Department of Computer Science, Federal Polytechnic Bauchi

<sup>2</sup>Department of Computer Science, Ahmadu Bello University Zaria

### ABSTRACT

Feature Selection (FS) is considered an important but yet a difficult data pre-processing task. Metaheuristics algorithm (MA) such as Particle Swarm Optimization (PSO), Salp Swarm Algorithm (SSA), Dragonfly Algorithm (DA), Grey Wolf Optimization (GWO) and Marine Predator Algorithm (MPA) are used for optimizing feature selection task. The Marine Predator Algorithm is a population based algorithm modeled after the foraging actions of marine predators. MPA is an efficient population based MA but has limitations in FS related task. These limitations include its method of initial population generation which is random initialization and suffers from diversity. Another limitation is exploration/exploitation phase imbalance which result to the search been stuck in local optima, that is, imbalance between the global search and local search. Despite, several attempts by researchers to address this challenges, these problems still persist. This work proposes a novel population initialization mechanism in MPA using logistic chaotic map. Ten benchmark datasets from the University of California in Irvine (UCI) repository were used throughout the experiment. The *k*-Nearest Neighbor (KNN) was used as the base classifier. To address overfitting problem, each of the datasets was divided into training and testing using *K*-fold cross-validation. Results obtained from the experiments showed a significant improvement against the traditional MPA.

### ARTICLE INFO

#### Article History

Received: October, 2024

Received in revised form: December, 2024

Accepted: December, 2024

Published online: January, 2025

### KEYWORDS

Marine Predator Algorithm, Population Initialization, Logistics chaotic map, Feature selection.

### INTRODUCTION

In the last few years, machine learning (ML) which is a subset of artificial intelligence have seen a new wave of publicity fueled by the huge and ever-increasing amount of data and computational power as well as the discovery of improved learning algorithms. However, the idea of a computer learning some abstract concept from data and applying them to yet unseen situations is not new and has been around at least since the 1950s (Badillo et al., 2020). Machine learning is an evolving branch of computational algorithms that are designed to emulate human intelligence by learning from the surrounding environment. They are considered the working horse in the new era of the so-called big data. Techniques based on machine learning have been applied successfully in diverse fields ranging from pattern recognition, computer vision, spacecraft engineering, finance, entertainment, and

computational biology to biomedical and medical applications (El Naqa & Murphy, 2015). Presently, data are recorded from various sources in different form thereby storing both relevant and irrelevant aspect of data not minding the cost of data. ML is not a technique that exist in isolation; it comprises of different steps which include data collection, data preprocessing, feature selection (FS), model training etc. Feature selection, as a data preprocessing strategy, has been proven to be effective (especially high-dimensional data) for various data mining and machine learning problems. The objectives of feature selection include building simpler and more comprehensible models, improving data mining performance, and preparing clean data. The recent proliferation of big data has presented some substantial challenges and opportunities to feature selection (Li et al., 2017).

Corresponding author: Abdulrazaq Ahmed Bello

[aabello.cs@fptb.edu.ng](mailto:aabello.cs@fptb.edu.ng)

Department of Computer Science, Federal Polytechnic, Bauchi.

© 2025. Faculty of Technology Education. ATBU Bauchi. All rights reserved

Bellman (1957) coined the phrase “the curse of dimensionality” to describe the extraordinarily rapid growth in the difficulty of problems as the number of variables (or the dimension) increases. A common experience is that the cost of an algorithm grows exponentially with dimension, making the cost prohibitive for moderate or large values of the dimension (Kuo & Sloan, 2005).

Metaheuristic approaches have been widely used in recent years and are more efficient in dealing with the optimization process of machine learning and FS algorithms/techniques (Holland, 1992). Examples of Metaheuristic algorithm include Genetic Algorithm (GA) (Holland, 1992), Ant Colony Optimization (ACO) (Dorigo & Di Caro, 1999), Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), Salp Swarm Algorithm (SSA) (Mirjalili et al., 2017) and the Marine Predators Algorithm (MPA) (Faramarzi et al., 2020).

In the context of feature selection, different studies have focused on the application of the MPA to handle the feature selection problems. Despite the fact that MPA is effective at solving the FS problem, stagnation in local optima, exploitation and exploration phase imbalance, and the diversity problem, are still challenges affecting the algorithm (Mugemanyi et al., 2023).

This study proposes an improve version of Marine Predator Algorithm for FS, to address the problem of population initialization diversity by initializing the initial population using logistics chaotic mapping, thereby, improving the exploration process of the algorithm.

## METHODOLOGY

This section describes the detailed structure and flow of the proposed technique, design, and algorithms that will be used for the

feature selection task. The methodology consists of the following stages: Data collection and description, data preprocessing, parameter settings used in the proposed and existing algorithms, implementation setup and system specification then integration of marine predator algorithm with logistics chaotic mapping.

In order to evaluate and compare the performance of the existing and proposed models, ten (10) datasets from the University College Irvine (UCI) Machine Learning repository were used, which are available for download from <https://archive.ics.uci.edu/>. The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms.

In this analysis, ten (10) popular datasets from the UCI repository was used for the evaluation of the proposed and existing methods. Several researchers have used these datasets as a benchmark for comparison in feature selection task. The datasets available in the repository are usually grouped into 3 categories based on sizes: small (features < 15), medium ( $15 \leq \text{features} \leq 50$ ), and large (features > 50) (Ghosh et al., 2019). During the training of machine learning model, choosing relevant features for classification is a very important task which is the core problem addressed in this study. Ten (10) datasets were carefully selected from difference literature. The characteristics of the ten (10) datasets are summarized in Table 1.

The experiments were carried out using Google Colab on a PC running Ubuntu 22.04.4 LTS with an Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz, 16 GB of RAM, a 256 GB SSD storage, and other specifications. Training and testing sets were created from the datasets in the ratio of 70:30 respectively.

**Table 1:** Description of the 10 UCI datasets used in the study

S/N	Datasets	No. of samples	No. of classes	No. of features
1	BreastEW	569	2	30
2	CongressEW	435	2	16
3	Ionosphere	351	2	34
4	Sonar	208	2	60
5	SpectEW	267	2	22
6	Vote	300	2	16
7	WaveformEW	5000	3	40

Corresponding author: Abdulrazaq Ahmed Bello

[aabello.cs@fptb.edu.ng](mailto:aabello.cs@fptb.edu.ng)

Department of Computer Science, Federal Polytechnic, Bauchi.

© 2025. Faculty of Technology Education. ATBU Bauchi. All rights reserved

S/N	Datasets	No. of samples	No. of classes	No. of features
8	HeartEW	270	2	13
9	Vowel	528	11	10
10	Glass	214	7	10

A 5-folds cross validation was used to divide the training set into five subsets. To make sure the results are statistically plausible, the experiments were repeated thirty times. The maximum number of search agents in the initialization and the maximum number of iterations for the improved algorithms are 30 and 1000 respectively.

Python programming language is mostly used for the data preprocessing, training, and implementation of the classification model. Python's advantages such as ease of use, huge framework library and flexibility make it the best

choice for creating machine learning models (Raschka et al., 2020). Python offers machine learning settings a remarkable level of power and customization. The language's straightforward syntax makes it easier to validate data and speeds the scraping, processing, refining, cleaning, arranging, and analyzing operations, which reduces the difficulty of working with other programmers (Ranjan et al., 2023). The hardware and software specifications used for the implementation of the model is shown in Table 2.

**Table 2:** Parameter settings of both algorithms for all experiments.

Parameter	Value
Population Size	30
Number of iteration	1000
Number of runs for algorithms	30
$\alpha$	0.99
$\beta$	0.01
The dimension of initial search space (D)	Number of columns in the input data
Search domain (lower and upper bound)	[0,1]
Fish Aggregating Device (FAD)	0.2
P	0.5

**Table 3:** Summary of hardware and software specifications

Software specification	
Language	Python 3.11.0
Platform	Google Colab, RAM 13GB, LibreOffice Writer
Operating system	Ubuntu 22.04.4 LTS
Hardware specification	
Processor	Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz x 8
Computer model	Dell Inc Latitude 5490 Intel(R) at 1.70GHz.
Memory	16.00 GB of RAM
Storage	256.1 GB of Solid State Drive

This proposed method, is based on a modified binary version of MPA that is used to carry out the search procedure. The MPA starts with population initialization using logistic chaotic mapping. This method injects randomness into the initial positions of predators (potential solutions) in the search space. Predators primarily utilize Lévy flights for

long-distance exploration. This mimics how real predators search vast areas for prey. When a predator encounters a promising food source (good solution), it switches to Brownian motion. This simulates the fine-tuned search around a potential solution for further improvement. During the exploitation phase, to prevent premature convergence and encourage

Corresponding author: Abdulrazaq Ahmed Bello

[aabello.cs@fptb.edu.ng](mailto:aabello.cs@fptb.edu.ng)

Department of Computer Science, Federal Polytechnic, Bauchi.

© 2025. Faculty of Technology Education. ATBU Bauchi. All rights reserved

diversification, a bit flipping mutation operation was introduced. This randomly flips bits in a predator's position (solution), potentially leading to the discovery of new potentially better solutions. The MPA evaluates each predator's fitness based on the objective function which is minimizing classification error and number of

features selected. By combining these elements, the MPA aims to achieve a good balance between finding new areas in the search space (exploration) and improving existing solutions (exploitation), leading to a diverse set of solutions which means converging on high-quality solutions.

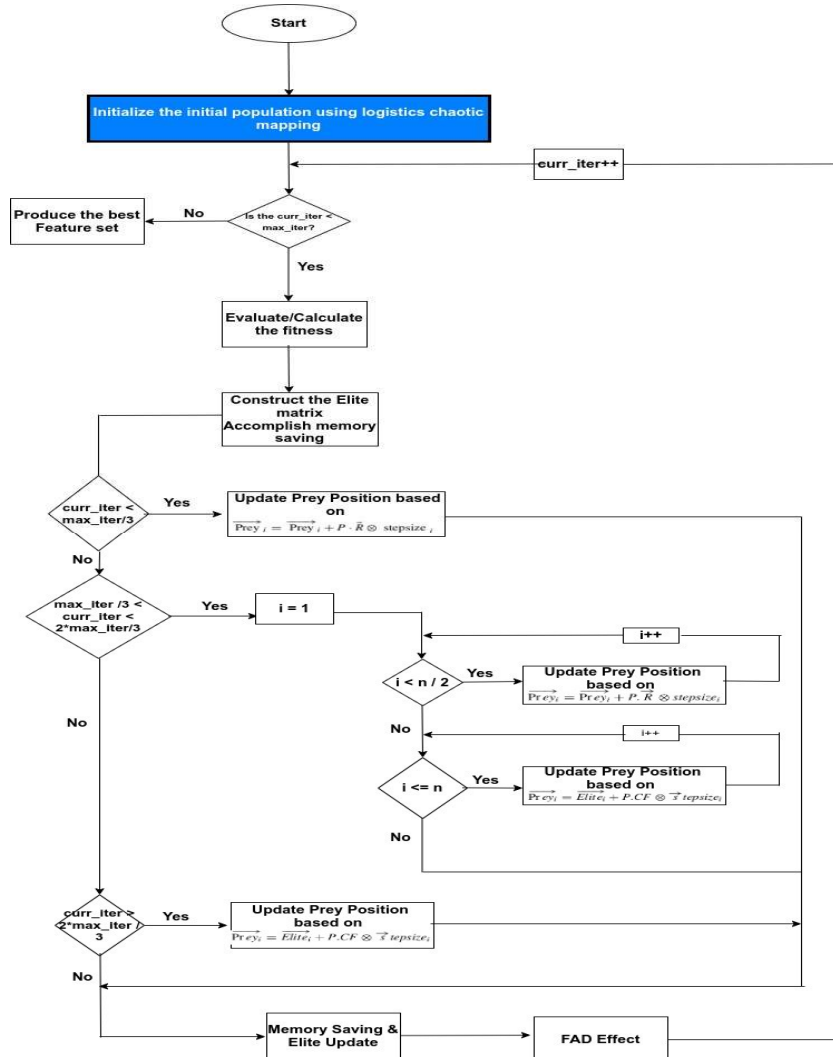


Figure 1: Overview of Modified MPA

Performance evaluation metrics, such as accuracy, the number of features, sensitivity and specificity were used to evaluate the experiments. The description of the evaluation metrics used are given as follows:

- i. Average: The average (mean) of the fitness function values obtained by running algorithm for M times. The Mean fitness function can be calculated by:

$$Mean = \frac{\sum_{i=1}^M fobj}{M}$$

- ii. Standard Deviation (StdDev): It is used to vary the fitness function value computed from the running algorithm M times. It shows an indicator of the stability of an algorithm. It indicates the measure of the spread of values around the mean. Larger values indicate wandering outcomes while smaller value implies the algorithm almost all the time converges towards the same value which is most preferred. It is also used to check the stability of the obtained results. StdDev can be calculated by:

$$Std = \sqrt{\frac{1}{M-1} \sum_{i=1}^M (fobj - mean)^2}$$

- iii. Accuracy: It is a measure that is defined as a total, correctly identified examples out of all the examples. Accuracy is determined as:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

where TP refers to True Positive, which means correctly identified, (FP) refers to incorrectly identified or False Positive, TN is defined as True Negative or correctly rejected. FN refers to False-Negative, which means incorrectly rejected.

- iv. Average Accuracy (AVG<sub>ACC</sub>): It is a measure that is defined as a average, correctly identified examples out of all the examples. Average Accuracy is calculated by:

$$avgAccuracy = \frac{\sum_{i=1}^M Accuracy}{M}$$

where M is the number of run times of the optimization algorithm to select a subset of feature.

- v. Sensitivity: this is the True Positive Rate (TPR). It is the degree to which the learning algorithm is able to identify the data records that have been positively classified; it is used to measure sensitivity. It is calculated by:

$$Sensitivity = \frac{TP}{TP + FN}$$

- vi. Specificity: this is the True Negative Rate (TNR). It defines the proportion of actual negatives that are correctly identified. It is the degree to which the learning algorithm is able to identify records of data with the negative class; it is used to measure specificity. It is calculated by:

$$Specificity = \frac{TN}{TN + FP}$$

The number of features: It is a result of the feature selection process of all algorithms. It usually has the minimal most significant number of features which results in higher classification accuracy rate.

## RESULTS AND DISCUSSION

This section discusses the results obtained from the proposed improvement of the marine predator algorithm versus the original marine predator algorithm used for the feature selection task. The improvement of the algorithm is to have a diverse initial population using logistics chaotic map.

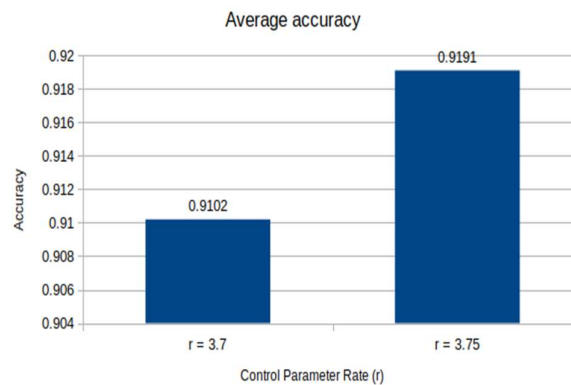
The purpose of this experiment is to compare different control parameter 'r' of logistics chaotic mapping for population initialization against the random initialization that is mostly used for population initialization. The experiment uses two (2) different values of r which is the control parameter of the Equation. The chaotic behavior of logistic map with r = 3.7 and 3.75 respectively, have been leveraged in various applications, such as pseudo-random bit generation and image encryption (Moysis et al., 2020). The experiment uses four (4) evaluation metrics which are average classification accuracy, average sensitivity, average specificity and average number of features selected. Table 4 presents the results of the experiment in terms of average classification accuracy with the standard deviation associated with each accuracy. The table shows the average accuracy of 0.9102 and 0.9191 with standard deviation of 0.002081 and 0.002642 which is also represented in Figure 2 and Figure 3 respectively. Using r = 3.75 yield a higher accuracy of 0.9191 with a lower standard deviation of 0.002642 which is the best result obtained for this experiment. Standard deviation as defined in section 3.10, says that, lower standard deviation suggest

stability of the algorithm. Considering the number of selected features in this experiment, the Table 4.1 presents the results of the experiment in terms of average number of features selected. When the value of  $r = 3.75$ , the experiment produces a lower average

number of selected features which is 9.4 compared to when the value of  $r = 3.7$  which yield a higher average number of selected features (10.9). Figure 2 – Figure 4 shows how using  $r = 3.75$  yield a better result compared to when  $r = 3.7$ .

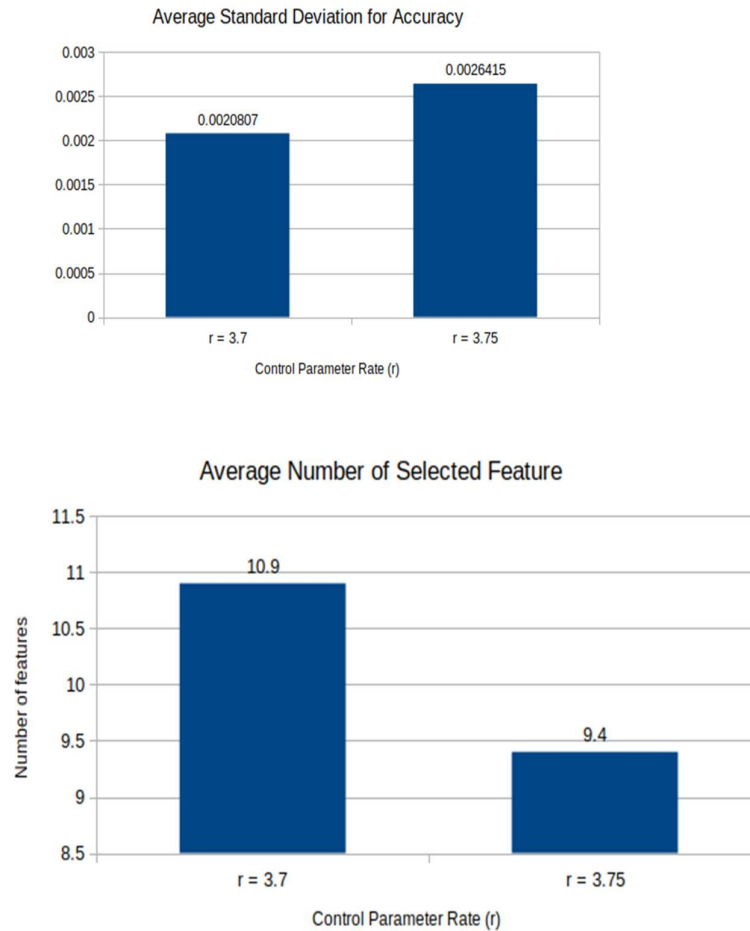
**Table 4:** Comparison based on average accuracy, average standard deviation and number of selected features of the two values of 'r' used in the study.

Dataset	Accuracy		Standard Deviation		Number of features	
	$r = 3.7$	$r = 3.75$	$r = 3.7$	$r = 3.75$	$r = 3.7$	$r = 3.75$
BreastEW	0.983	<b>0.986</b>	0.001455	<b>0.000339</b>	19	<b>11</b>
CongressEW	0.975	<b>0.98</b>	0.000707	<b>0</b>	<b>3</b>	5
IonosphereEW	0.894	<b>0.923</b>	<b>0.000465</b>	0.000595	8	<b>5</b>
Sonar	<b>0.8</b>	0.796	0.002121	<b>0.001414</b>	26	26
SpectEW	0.845	<b>0.848</b>	0.000353	<b>0.000212</b>	9	<b>4</b>
Vote	0.962	<b>0.966</b>	0.004243	<b>0.003536</b>	4	<b>3</b>
WaveformEW	0.897	<b>0.9</b>	<b>0.000707</b>	0.009798	19	19
Glass	0.934	<b>0.955</b>	0.002243	<b>0.001973</b>	7	7
HeartEW	0.812	<b>0.837</b>	<b>0.008513</b>	0.008548	6	6
Vowel	1	1	0	0	8	8
<b>AVERAGE</b>	<b>0.9102</b>	<b>0.9191</b>	<b>0.002081</b>	<b>0.002642</b>	<b>10.9</b>	<b>9.4</b>



**Figure 2:** Average Accuracy using different control parameter rate





**Figure 4:** Average Number of features selected using different control parameter

Table 5 presents the results of the experiment in terms of average sensitivity and specificity with the standard deviation associated with each dataset. The table shows the average sensitivity of 0.8894 and 0.9123 with standard deviation of 0.002390 and 0.002966 which is also represented in Figure 5 and Figure 6 respectively. Using  $r = 3.75$  yield a higher sensitivity of 0.9191 with a higher standard deviation of 0.002642 which is still a good result since the standard deviation is relatively small though it is higher when compared to that of when  $r = 3.7$ . Average Sensitivity of 0.9123 means the model was able to actually identify the data that been positively

classified with a rate of 91.23% which is also known as True Positive Rate (TPR). With  $r = 3.7$ , the experiment produce a lower and more statistically significant standard deviation of 0.002390 but the Recall rate is lower (88.94%). The table shows the average specificity of 0.9023 and 0.9267 with standard deviation of 0.003159 and 0.003281 which is also represented in Figure 7 and Figure 8 respectively. Using  $r = 3.75$  yield a higher specificity of 0.9267 with a higher standard deviation of 0.003281 which is still a good result since the standard deviation is relatively small though it is higher when compared to that of when  $r = 3.7$ .

Corresponding author: Abdulrazaq Ahmed Bello

✉ [abdulrazagabello@fptb.edu.ng](mailto:abdulrazagabello@fptb.edu.ng)

Department of Computer Science, Federal Polytechnic, Bauchi.

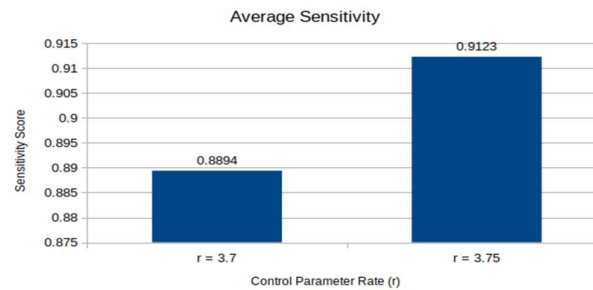
© 2025. Faculty of Technology Education. ATBU Bauchi. All rights reserved

Average Specificity of 0.9267 means the model was able to actually identify the data that been negatively classified with a rate of 92.67% which is also known as True Negative

Rate. With  $r = 3.7$ , the experiment produce a lower and more statistically significant standard deviation of 0.003159 but the Recall rate is lower at 90.23%.

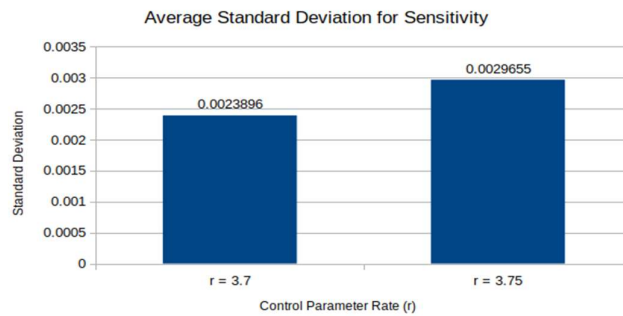
**Table 5:** Comparison based on average sensitivity, average specificity and average standard deviation of the two values of 'r' used in the study.

Dataset	Sensitivity		Standard Deviation		Specificity		Standard Deviation	
	r = 3.7	r = 3.75	r = 3.7	r = 3.75	r = 3.7	r = 3.75	r = 3.7	r = 3.75
BreastEW	0.97	<b>0.981</b>	0.000775	<b>0.000498</b>	0.974	<b>0.991</b>	<b>0.000707</b>	0.007368
CongressEW	<b>0.962</b>	0.96	<b>0.001312</b>	0.001537	0.964	<b>0.98</b>	<b>0.006289</b>	0.006998
IonosphereEW	0.853	<b>0.92</b>	<b>0.003356</b>	0.003377	<b>0.904</b>	0.901	<b>0.005168</b>	0.005884
Sonar	0.786	<b>0.8</b>	0.001922	<b>0.001317</b>	<b>0.797</b>	0.793	<b>0.001486</b>	0.001567
SpectEW	0.746	<b>0.9</b>	<b>0.002002</b>	0.002443	0.76	<b>0.86</b>	0.007402	<b>0.000877</b>
Vote	0.961	<b>1</b>	0.002587	<b>0</b>	0.95	<b>0.973</b>	<b>0.002404</b>	0.002475
WaveformEW	0.876	<b>0.901</b>	<b>0.000707</b>	0.003356	0.874	<b>0.93</b>	<b>0.001767</b>	0.003394
Glass	0.947	<b>0.963</b>	<b>0.002587</b>	0.0072	1	1	0	0
HeartEW	<b>0.822</b>	0.799	0.008648	<b>0.008513</b>	0.8	<b>0.839</b>	0.006364	<b>0.004243</b>
Vowel	<b>0.971</b>	0.899	<b>0</b>	0.001414	1	1	0	0
<b>AVERAGE</b>	<b>0.8894</b>	<b>0.9123</b>	<b>0.0023896</b>	<b>0.0029655</b>	<b>0.9023</b>	<b>0.9267</b>	<b>0.0031587</b>	<b>0.0032806</b>

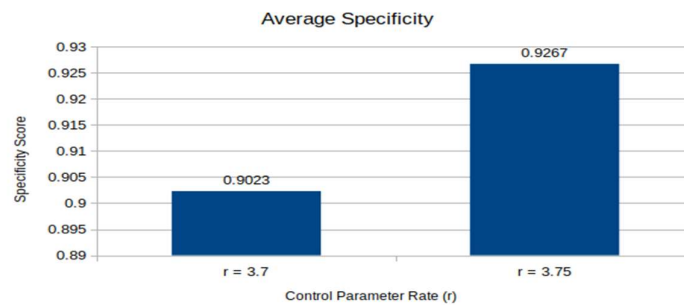


**Figure 5:** Average Sensitivity using different control parameter rate

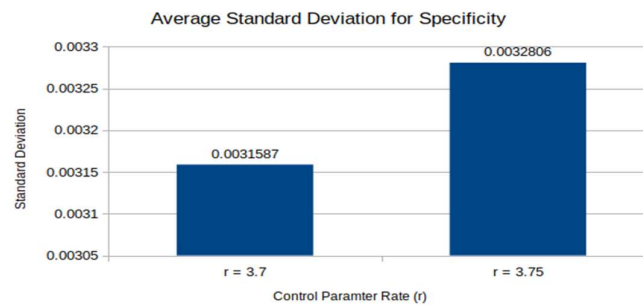




**Figure 6:** Average Standard Deviation for Sensitivity



**Figure 7:** Average Specificity using different control parameter rate



**Figure 8:** Average Standard Deviation for Specificity

The result from the experiment shows how introducing logistic chaotic map to the initial population generation enhances the exploration of the search space. Higher rates were observed from the early stage of the iteration which is where the exploration actually happens. In conclusion, the proposed method of using logistics chaotic mapping to initialize the population with a control parameter of 3.75 is selected as the best algorithm because of its higher statistical results.

## CONCLUSION

In conclusion, this variation of the Marine Predator Algorithm (MPA), incorporating logistic chaotic mapping and bit flipping mutation, proves to be valuable for feature selection in Machine Learning Classification task. The use of logistic chaotic mapping for initialization injects controlled randomness, leading to a diverse initial population. The utilization of K-Nearest Neighbors (KNN) as the classifier ensures the practical applicability of the proposed approach. KNN's simplicity and effectiveness makes it well-suited for feature selection tasks. It is also recommended that study is undertaken to investigate the effect of introducing other chaotic mapping such as circle map, tent map, duffing map, gauss map and exponential map to enhance the initial population of the MPA.

## REFERENCES

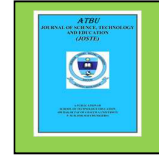
- Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., Siebourg-Polster, J., Steiert, B., & Zhang, J. D. (2020). An Introduction to Machine Learning. *Clinical Pharmacology & Therapeutics*, 107(4), 871–885.  
<https://doi.org/10.1002/cpt.1796>
- Bellman, R. (1957). In *Dynamic Programming*. Princeton University Press.
- Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: A new meta-heuristic. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 1470–1477.  
<https://doi.org/10.1109/CEC.1999.782657>
- El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? In I. El Naqa, R. Li, & M. J. Murphy (Eds.), *Machine Learning in Radiation Oncology: Theory and Applications* (pp. 3–11). Springer International Publishing.  
[https://doi.org/10.1007/978-3-319-18305-3\\_1](https://doi.org/10.1007/978-3-319-18305-3_1)
- Elminaam, D. S. A., Nabil, A., Ibraheem, S. A., & Houssein, E. H. (2021). An Efficient Marine Predators Algorithm for Feature Selection. *IEEE Access*, 9, 60136–60153.  
<https://doi.org/10.1109/ACCESS.2021.3073261>
- Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications*, 152, 113377.  
<https://doi.org/10.1016/j.eswa.2020.113377>
- Ghosh, M., Guha, R., Alam, I., Lohariwal, P., Jalan, D., & Sarkar, R. (2019). Binary genetic swarm optimization: A combination of GA and PSO for feature selection. *Journal of Intelligent Systems*, 29(1), 1598–1610
- Holland, J. H. (1992). Genetic Algorithms. *Scientific American*, 267(1), 66–72.  
<https://doi.org/10.1038/scientificamerican0792-66>
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, 1942–1948.  
<https://doi.org/10.1109/ICNN.1995.488968>
- Kuo, F. Y., & Sloan, I. H. (2005). Lifting the curse of dimensionality. *Notices of the AMS*, 52(11), 1320–1328
- Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature Selection: A Data Perspective. *ACM Comput. Surv.*, 50(6).  
<https://doi.org/10.1145/3136625>
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114, 163–191.

Corresponding author: Abdulrazaq Ahmed Bello

✉ [aabello.cs@fptb.edu.ng](mailto:aabello.cs@fptb.edu.ng)

Department of Computer Science, Federal Polytechnic, Bauchi.

© 2025. Faculty of Technology Education. ATBU Bauchi. All rights reserved



- <https://doi.org/10.1016/j.advengsoft.2017.07.002>
- Moyasis, L., Volos, C., Jafari, S., Munoz-Pacheco, J. M., Kengne, J., Rajagopal, K., & Stouboulos, I. (2020). Modification of the logistic map using fuzzy numbers with application to pseudorandom number generation and image encryption. *Entropy*, 22(4), 474
- Mugemanyi, S., Qu, Z., Rugema, F. X., Dong, Y., Wang, L., Bananeza, C., Nshimiyimana, A., & Mutabazi, E. (2023). Marine predators' algorithm: A comprehensive review. *Machine Learning with Applications*, 12, 100471. <https://doi.org/10.1016/j.mlwa.2023.100471>
- Ranjan, M., Barot, K., Khaimar, V., Rawal, V., Pimpalgaonkar, A., Saxena, S., & Sattar, A. (2023). Python: Empowering Data Science Applications and Research. *Journal of Operating Systems Development & Trends*, 10, 27–33. <https://doi.org/10.37591/joosdt.v10i1.576>
- Raschka, S., Patterson, J., & Nolet, C. (2020). Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. *Information*, 11(4), 193. <https://doi.org/10.3390/info11040193>