SINTEZA 2022
INTERNATIONAL SCIENTIFIC CONFERENCE ON INFORMATION TECHNOLOGY AND DATA RELATED RESEARCH

STUDENT SESSION

ARITHMETIC OPTIMIZATION ALGORITHM FOR SPAM DETECTION

Luka Jovanović,(Student)[1]*,
Bushra Al Barwani[2],
Esraa Al Maani[2],
Miodrag Živković,(Mentor)[1],
Nebojša Bačanin[1]

[1]Singidunum University,
Belgrade, Serbia

[2]Modern Colleague of
Business and Science,
Muscat, Oman

Abstract:

We've all dealt with spam emails, which regularly fill our inboxes and require just a few seconds of our time to remove them. When businesses are forced to develop spam filters and use filtering software, genuine emails may be mistakenly redirected to spam folders. When businesses take on spamming customers, it results in negative consequences for their network and IP reputation, as well as extra expenses associated with employing additional staff to deal with spam and abuse complaints exclusively. When you check off a list of emails that are spam and then delete them each time you log in, it may not seem like a major matter, but there are additional issues involved with sending and receiving spam communications. We do not often consider the expenses connected with spam concerns for organizations or Internet Service Providers, which might be significant (ISP). Non-stop email transmission is disrupted, and an increase in bandwidth utilization, a decrease in in-service performance, and decreased staff productivity are all consequences of this practice. This research paper will explain how the logistic regression linear model determines which emails are spam and which are not by using arithmetic optimization algorithms in machine learning.

Keywords:

Spam email, CSDMC2010, logistic regression, machine learning, swarm intelligence.

INTRODUCTION

A surge in e-commerce has led to an increase in advertising emails, as well as malicious actors attempting to obtain sensitive information by using phishing techniques. Technology experts refer to unsolicited bulk emails as spam. In recent years up to 55% of global email traffic has consisted of spam emails [1]. These unwanted emails waste network bandwidth, storage space, inconvenience the recipients, and may even propagate malware in the attachments [2]. However, spam prevention technology has come a long way in recent years. With many approaches aimed at reducing spam emails being developed [3]. These can roughly be divided as static methods that use pre-defined lists or as dynamic techniques that make use of text categorization approaches developed using statistical techniques or artificial intelligence (AI).

Correspondence:
Luka Jovanović

e-mail:
luka.jovanovic.191@singimail.rs

DOI: 10.15308/Sinteza-2022-406-413

In the field of (AI), one field that has shown several possibilities and continues to do so is machine learning (ML). Due to AI being employed in a broad range of industries, several approaches and algorithms are available within this domain. The most common classifications of AI are metaheuristics and ML. With metaheuristics being problem-independent and applicable over a very broad range of domains. Moreover, metaheuristics may be split into two subcategories: those that are inspired by nature and those that are not inspired by nature, depending on the sort of phenomenon that is emulated. Tabu search (TS) [4] and differential evolution (DE) [5] are two examples of metaheuristics that are not based on natural phenomena. Evolutionary algorithms (EA) [6], which simulate natural evolution, and swarm intelligence, which mimics a group of organisms from nature are two of the most prominent types of nature-inspired metaheuristics.

Swarm intelligence makes use of a random population as an evolutionary unit as well as mechanisms for interpersonal collaboration amongst individual agents to excel at addressing a variety of optimization issues [7]. Additionally, swarm intelligence metaheuristics may improve a wide range of AI approaches and techniques. As a result, hybrid approaches that combine swarm intelligence with several machine learning models and are tailored to a wide range of real-world issues are one of the most current and popular study areas.

The arithmetic optimization algorithm (AOA) presents an exception among swarm intelligence algorithms [8]. While population-based it does not draw inspiration from natural behaviors but rather models behavior on abstract mathematical concepts, making use of arithmetic operators such as multiplication, division as well as addition and subtraction during optimization. Additionally, it shows exceptional results when tackling demanding optimization problems.

This study presents a novel logistic regression (LR) methodology [9], trained with the AOA that aims to combine the advantages inherent to LR, simplicity, efficiency, and fast classification while avoiding the drawbacks of fast convergence to non-optimal local minima by applying the AOA for training purposes.

The contributions of the conducted research may be summarised as the following:

- The first-ever application of the arithmetic optimization algorithm to the problem of filtering spam emails
- A performance evaluation of this application on a real-world data set.
- A comparative analysis with another contemporary algorithm, the ABC algorithm, addresses the same challenge to demonstrate a comparison of their performance.

The remainder of this work is structured per the following: Section 2 provides an overview of LR, swarm intelligence, hyperparameter optimization through a review of related literature; Section 3 provides a detailed description of the proposed approach. Details on the datasets, data pre-processing operations, feature selection methodology, as well as the conducted experiments, are given in Section 4. Finally, Section 5 provides a conclusion to the paper and provides proposals for future work.

2. BACKGROUND AND LITERATURE REVIEW

As an AI subfield ML emphasizes data and algorithm usage to imitate how humans learn. It uses algorithms and statistical approaches to make predictions and classifications, thus uncovering essential insights within information mining projects. The insights gained help in decision-making within businesses and applications. These algorithms predict or classify events depending on input data (labeled and unlabelled). As a result, the algorithm generates an estimate concerning a pattern in the data. In addition, an error function within the machine-learning algorithm assesses the prediction and classification, which can enhance the outcome accuracy. Similarly, a model optimization process ensures data in the training set and the weights are modified to decrease discrepancy between the model estimate and the known example.

Some notable uses of ML algorithms are to predict traffic, recognize speech and images, as well as filter email spam and malware. Traffic predictions happen through real-time location and the average time that is taken. Through supervised machine techniques, machine learning helps solve different problems. For instance, the technique aids in classifying spam in a distinct folder from a person's email inbox. The supervised machine learning methods used in traffic prediction include LR, support vector machines, neural networks, and random forest. In email spam and malware filtering, machine-learning methods use multi-layer perception, naïve Bayes classifier, and decision tree. Nevertheless, ML techniques are vulnerable to high computational costs, slow operational speeds for real-time applications, misclassifications, overfitting in a local minimum, the curse of dimensionality, and sensitivity to feature weights.

2.1. LOGISTIC REGRESSION (LR)

The LR model is often utilized for predictive modeling and analytics. The method extends to ML applications. In the analytic technique, the independent variable is categorical. The model helps in understanding the association between independent and dependent variables by approximating probabilities utilizing an LR equation. This approach reduces mistakes related to the output computed by a logistic activation function. In email classification, the LR model applies a trained online gradient descent algorithm to determine authentic and unauthentic emails. In addition, Predictive models used in the logistic analysis include probity, ordered logit, generalized linear model, multinomial logit, mixed logit, and discrete choice. Predictive models designed and developed using the LR method help in examining different categorical outcomes. Binary LR helps in determining event probability for a categorical reaction variable with two results. Conversely, multinomial LR classifies subjects into separate groups depending on a categorical range of variables to analyze and predict behavior. Consequently, the LR model assists in predicting the probability of an event happening.

2.2. SWARM INTELLIGENCE

Swarm intelligence refers to a group of often nature-inspired artificial intelligence concepts based on collective habits of social colonies. Examples of swarm intelligence applications include artificial bee colonies (ABC) [10], artificial immune systems, cat swarm optimization, particle swarm optimization, and ant colony optimization. The nature-inspired algorithms comprise adaptive features that improve artificial intelligence applications. For example, a swarm intelligence system can combine a negative selection algorithm (NSA) [11], the primary algorithms in the artificial immune system (AIS) [12], with particle swarm optimization (PSO) to detect spam emails [13]. Similarly, ant colony optimization (ACO) uses pheromone laying/pheromone based on the habits of the real ants. The method can solve different optimization problems including numerous medical applications [14] [15] [16], task scheduling [17] [18] [19], wireless sensor network optimization [20] [21] [22]. Swarm intelligence optimization techniques achieve high accuracy. The classification performance relies on the predetermined parameters and the problem type.

2.3. HYPERPARAMETER OPTIMIZATION

Model optimization presents many challenges in machine learning execution solutions. The goal of hyperparameter optimization in machine learning is to deduce the hyperparameters for a given machine learning algorithm so that the algorithm's performance will be efficient. Notably, hyperparameters can influence the training of machine learning algorithms. Engineers need to understand how to optimize the algorithms to achieve optimal functionality. Hyperparameters are utilized by swarm intelligence algorithms to search within a problem domain. These parameters are representative of solutions to the problem being optimized. Swarm Intelligence applications, such as AOA maintain and enhance a collection of viable solutions during the guided search until the user meets some predefined stopping. The Swarm intelligence rule requires solutions to improve their fitness value when they have more space and computational power. By applying swarm intelligence to various established ML algorithms an overall performance increase can be seen [23] [24] [25].

3. OVERVIEW OF THE ARITHMETIC OPTIMIZATION ALGORITHM

Much like other population algorithms, the AOA initializes a random population at the start of an optimization. Following this, the solution set is evaluated through the use of an objective function and gradually refined over numerous iterations. Owing to the stochastic nature of this approach, an optimal solution cannot be guaranteed, however, the chances of locating the global optima improve through repeated iterations. The optimization process is comprised of two stages, exploration that includes covering large areas of the search space with agents in an attempt to avoid local optima and exploitation that involves refining the accuracy of the solutions attained during exploration

The base inspiration for the AOA comes from number theory and calculus, with simple mathematical operations including addition, subtraction, multiplication, and division forming a hierarchy that is the basis for the algorithms function.

After the initial random population creation, the AOA evaluates each agent. This evaluation is repeated following every iteration, and the best performing agent (x) is considered the new optimal solution. However, before the optimization begins phase selection needs to be performed. The math optimizer accelerated (MOA) function shown in Eq. (1) is used.

$$MOA\left(C_{iter}\right) = Min + C_{iter} \times \left(\frac{Max - Min}{M_{iter}}\right)$$

Equation 1 - The math optimizer accelerated (MOA) function

In which C_{Iter} is the current iteration, M^{Iter} determines the maximum number of iterations, while Min and Max are dictated by the minimum and maximum possible values of the accelerated function.

During the exploration phase, the AOA makes use of Division (D) and Multiplication (M), to cover larger sections of the search space. However, high dispersion rates limit search accuracy, and thus during this phase, the algorithm focuses on finding near-optimal solutions, that can be improved upon in later iterations. The models for the main operations used during exploration are shown in Eq. (2).

$$x_{i,j}\left(C_{iter}+1\right) = \begin{cases} best\left(x_j\right) \div \left(MOP+\in\right) \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), r_2 < 0.5 \\ best\left(x_j\right) \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), r_2 \geq 0.5 \end{cases}$$

Equation 2 - Math operations governing exploration

Where C_{Iter} is the current agent of the i-th iteration, $x_{i,j}(C_{Iter})$ defines the jth position of the ith agent, $best(x_j)$ is the current optimal solution, ∊ a small integer value. The upper bound is represented by UB_j, while the lower bound is denoted by LB_j. Finally, μ represents a control parameter used to fine-tune the search process.

The exact operation used in the exploration is determined by the value MOA function, and conditions r1 are a random value. Should the value of the second conditional $r_2 < 0.5$ the division will be used in the exploration, while in the case of $r_2 \geq 0.5$ multiplication will be used instead. The math optimizer probability (MOP) function is shown in Eq. (3).

$$MOP\left(C_{iter}\right) = 1 - \frac{C\,iter^{\frac{1}{a}}}{M\,iter^{\frac{1}{a}}}$$

Equation 3 - The math optimizer probability (MOP) function

Where $MOP(C_{Iter})$ denotes the ith iterations MOP value, C_{Iter} the current iteration, M_{Iter} the maximum number of iterations, and α is the parameter that defines the accuracy over iterations.

The other phase in the AOA algorithms focuses on exploitation. In this phase, the higher density search required for attaining accurate results is met by replacing the previously utilized methods with addition (A) and subtraction (S). With their lower rate of dispersion, an optimum is more easily approached. This mode is entered when the MOA function value for r1 is greater than $MOA(C_{Iter})$. The models for the main operations used during exploitation are shown in Eq. (3).

$$x_{i,j}\left(C_{iter}+1\right) = \begin{cases} best\left(x_j\right) - MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), r_3 < 0.5 \\ best\left(x_j\right) + MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), r_3 \geq 0.5 \end{cases}$$

Equation 4 - Math operations governing the exploitation

With C_{Iter} representing the current agent of the i-th iteration, $x_{i,j}$(CIter) being the jth position of the ith agent, $best(x_j)$ standing in for the current optimal solution. The upper and lower bounds are denoted by UB_j, LB_j respectively. With μ again representing a control parameter used to adjust the search method.

The exact operation used for exploitation is determined by the value MOA function, and conditions r1 are a random value. When the third conditional $r_3 < 0.5$ the subtraction is applied, however, if $r_3 \geq 0.5$ then the addition is used.

With both stages in mind, the full pseudo-code for the AOA can be seen in Listing 1.

4. EXPERIMENTAL SETUP AND RESULTS

To provide valid comparative results, this research relies on the publicly available CSDMC2010 SMAP dataset for performance evaluation. The dataset is comprised of 4327 emails, with 1378 (31.85%) being spam, and 2949 (68.15%) valid emails. With 82148 distinct terms available in the provided dataset. However, the dataset is imbalanced with an evaluated factor of 2.14, as well as sparse with a percentage of 90.48% with a total feature vector size of 1000.

For this research independent implementations of both the ABC algorithm and the AOA have been done, to provide valid groups for result comparison. Due to the metaheuristic nature of swarm intelligence, testing and result representation are adequately adjusted, with multiple repeating independent runs taking place during testing, and results showing statistical results of multiple iterations. Additionally, various parameter settings were tested in search of optimal performance, as shown in Table 1 by the limit parameter having both values of 100 and 200, as well as the SN showing numbers of agents in a given population.

Of note, is that better performance is generally observed when using a limit of 200, as such the results in Table 2 use a limit value of 200.

According to the attained results, the overall performance of the LR model trained with the AOA optimizer is fairly similar to that of LR trained with the ABC algorithm, when considering best results, with appropriate limit and MR values, for the same size population in both algorithms.

With a smaller experimental features size of 500, the ABC algorithm performs slightly better with a smaller population of 40 and 60, while the AOA outperforms it with a population of 80. However, with a feature space of 1000 and a population of 80, the results are virtually identical. Additionally, it is worth noting that the ABC poses a slight advantage over the AOA, as it poses a larger number of adjustable parameters including MR and limit, these allow for a more detailed adaptation to the presented problem.

```
Initialize parameters α, μ.
Initialize random agent positions (Agents: i=1, ..., N .)
while (C_Iter < M_Iter) do
    Evaluate agent fitness (F F)
    Determine fittest agent obtained so far
    Refresh the value of MOA according to Eq. (1).
    Refresh value of MOP according to Eq. (3).
    for (i=1 to Solutions) do
        for ( j=1 to Positions) do
            Generate random values between 0 and 1 for  r 1, r 2, and r 3
            if r 1 > MOA then
                Enter the exploration phase
                if r 2 >0.5 then--
                    (1) Apply the Division math operator (D “ ÷ ”).
                    Update ith agents positions according to rule one in Eq. (2).
                else
                    (2) Apply the Multiplication math operator (M “ × ”).
                    Update ith agents positions according to rule two in Eq. (2).
                end if
            else
                Enter the exploitation phase
                if r 3 > 0.5 then
                    (1) Apply the Subtraction math operator (S “ − ”).
                    Update ith agents positions according to rule one in Eq. (4).
                else
                    (2) Apply the Addition math operator (A “ + ”).
                    Update ith agents positions according to rule two in Eq. (4).
                end if
            end if
        end for
    end for
    C Iter=C Iter+1
end while
Return the best agent (x).
```

Listing 1 – Pseudocode for the arithmetic optimization algorithm.

SN	MR	Limit	Feature vector size = 500					Feature vector size = 1000				
			Best	Worst	Median	Mean	Std.	Best	Worst	Median	Mean	Std.
40	0.05	100	98.18%	97.81%	98.03%	98.04%	0.11	98.57%	98.18%	98.39%	98.38%	0.09
		200	98.32%	97.78%	98.03%	98.01%	0.12	98.66%	98.22%	98.45%	98.44%	0.10
	0.08	100	98.16%	97.86%	97.99%	98.01%	0.07	98.64%	98.15%	98.30%	98.33%	0.11
		200	98.18%	97.81%	98.03%	98.03%	0.09	98.48%	98.11%	98.34%	98.34%	0.11
	0.1	100	98.18%	97.74%	97.93%	97.93%	0.11	98.51%	98.04%	98.27%	98.25%	0.13
		200	98.16%	97.77%	97.94%	97.96%	0.10	98.55%	98.02%	98.24%	98.24%	0.12
	0.2	100	98.06%	97.46%	97.75%	97.75%	0.14	98.36%	97.76%	98.04%	98.03%	0.14
		200	98.01%	97.63%	97.75%	97.77%	0.10	98.32%	97.72%	98.06%	98.03%	0.16
60	0.05	100	98.34%	97.95%	98.09%	98.11%	0.10	98.63%	98.25%	98.45%	98.41%	0.10
		200	98.20%	97.93%	98.10%	98.07%	0.07	98.71%	98.25%	98.45%	98.42%	0.11
	0.08	100	98.22%	97.83%	98.07%	98.05%	0.10	98.55%	98.13%	98.41%	98.39%	0.11
		200	98.25%	97.90%	98.06%	98.08%	0.08	98.60%	98.18%	98.37%	98.36%	0.09
	0.1	100	98.18%	97.81%	97.99%	98.02%	0.10	98.54%	98.15%	98.31%	98.32%	0.11
		200	98.25%	97.86%	98.05%	98.05%	0.11	98.56%	98.11%	98.32%	98.35%	0.12
	0.2	100	98.16%	97.57%	97.85%	97.83%	0.14	98.36%	97.81%	98.11%	98.12%	0.15
		200	98.10%	97.46%	97.82%	97.81%	0.15	98.36%	97.65%	98.12%	98.11%	0.15
80	0.05	100	98.18%	97.76%	98.03%	98.06%	0.09	98.57%	98.20%	98.33%	98.36%	0.09
		200	98.18%	97.88%	98.02%	98.02%	0.08	98.48%	98.11%	98.35%	98.34%	0.09
	0.08	100	98.23%	97.81%	98.03%	98.01%	0.11	98.54%	98.15%	98.38%	98.35%	0.11
		200	98.23%	97.71%	98.04%	98.02%	0.12	98.54%	98.11%	98.39%	98.33%	0.11
	0.1	100	98.29%	97.83%	98.04%	98.03%	0.13	98.53%	98.18%	98.34%	98.34%	0.09
		200	98.22%	97.86%	97.99%	97.97%	0.10	98.53%	98.13%	98.31%	98.33%	0.11
	0.2	100	98.06%	97.54%	97.83%	97.84%	0.15	98.34%	97.86%	98.10%	98.08%	0.11
		200	98.06%	97.66%	97.88%	97.89%	0.11	98.34%	97.74%	98.07%	98.05%	0.14

Table 1 - ABC trained LR classification statistics for the CSDMC2010 dataset

All testing done in this research has been done with population sizes of forty, sixty, and eighty for each algorithm

SN	Limit	Feature vector size = 500					Feature vector size = 1000				
		Best	Worst	Median	Mean	Std.	Best	Worst	Median	Mean	Std.
40	98.45%	97.45%	97.66%	97.71%	0.13	98.52%	98.15%	98.31%	98.33%	0.10	0.09
60	98.19%	97.76%	98.03%	98.11%	0.14	98.61%	98.18%	98.35%	98.39%	0.12	0.11
80	98.26%	97.92%	98.05%	98.13%	0.12	98.54%	98.12%	98.32%	98.32%	0.11	0.11

Table 2 - AOA trained LR classification statistics for the CSDMC2010 dataset

Based on the conducted experiments, it can be deduced that the AOA is perfectly acceptable for addressing the problem of spam filtering, matching the performance of the ABC algorithm, which has been shown to outperform traditional classification methods. Making the AOA a suitable choice for application in advanced spam prevention systems.

5. CONCLUSION

In conclusion, this work presents the first-ever application of the AOA algorithm to the problem of spam filtration found in the literature. The algorithm has been independently implemented and tested on a real-world CSDMC2010 SMAP dataset, and the resulting performance was evaluated in comparison to another popular independently implemented metaheuristic algorithm, the ABC algorithm. Both metaheuristics present similarly admirable results in independent implementations and have similar overall performance when evaluated, however, the AOA algorithm shows great potential for addressing and resolving this kind of challenge. Accordingly, future work will focus on modifying and improving the basic AOA algorithm in hopes of improving overall performance, additionally, applications of the algorithm to similarly difficult challenges will be carried out.

6. REFERENCES

[1] S. R. Department, Statista , [Online]. Available: https://www.statista.com/statistics/1270424/daily-spam-volume-global/. [Accessed 13 3 2022].

[2] S. Savita Kumari and Y. Partibha, "Machine Learning based Optimization Scheme for Detection of Spam and Malware Propagation in Twitter," *Machine Learning*, vol. 12, no. 2, 2021.

[3] R. Sanjeev , V. Anil Kumar and B. Tarunpreet, "A review on social spam detection: Challenges, open issues, and future directions," *Expert Systems with Applications*, vol. 186, p. 115742, 2021.

[4] F. Glover and M. Laguna, "Tabu search," *Handbook of combinatorial optimization*, pp. 2093--2229, 1998.

[5] K. V. Price, "Differential evolution," *Handbook of optimization*, pp. 187--214, 2013.

[6] H. Muhlenbein, M. Gorges-Schleuter and O. Kramer, "Evolution algorithms in combinatorial optimization," *Parallel computing*, vol. 7, pp. 65--85, 1988.

[7] M. Sotomayor, P.-C. J. David and F. Castiglione, "Complex Social and Behavioral Systems: Game Theory and Agent-based Models," 2020.

[8] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer methods in applied mechanics and engineering*, vol. 376, p. 113609, 2021.

[9] R. E. Wright, "Logistic regression," 1995.

[10] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied soft computing*, vol. 8, no. 1, pp. 687--697, 2008.

[11] Z. Ji and D. Dasgupta, "Real-valued negative selection algorithm with variable-sized detectors," *Genetic and evolutionary computation conference*, pp. 287--298, 2004.

[12] S. A. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evolutionary computation*, vol. 8, no. 4, pp. 443--473, 2000.

[13] K. Agarwal and T. Kumar, "Email spam detection using integrated approach of na{\"\i}ve Bayes and particle swarm optimization," *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 685--690, 2018.

[14] S. Prakash, M. V. Kumar, R. S. Ram, M. Zivkovic, N. Bacanin and M. Antonijevic, "Hybrid GLFIL Enhancement and Encoder Animal Migration Classification for Breast Cancer Detection,"*Computer Systems Science and Engineering*", vol. 41, no. 2, pp. 735--749, 2022.

[15] I. Strumberger, A. Rakic, S. Stanojlovic, J. Arandjelovic, T. Bezdan, M. Zivkovic and N. Bacanin, "Feature Selection by Hybrid Binary Ant Lion Optimizer with COVID-19 dataset," *2021 29th Telecommunications Forum (TELFOR)*, pp. 1--4, 2021.

[16] E. Tuba, I. Strumberger, T. Bezdan, N. Bacanin and M. Tuba, "Classification and feature selection method for medical datasets by brain storm optimization algorithm and support vector machine," *Procedia Computer Science*, vol. 162, pp. 307--315, 2019.

[17] N. Bacanin, T. Bezdan, E. Tuba, I. Strumberger, M. Tuba and M. Zivkovic, "Task scheduling in cloud computing environment by grey wolf optimizer," *2019 27th Telecommunications Forum (TELFOR)*, pp. 1--4, 2019.

[18] T. Bezdan, M. Zivkovic, M. Antonijevic, T. Zivkovic and N. Bacanin, "Enhanced flower pollination algorithm for task scheduling in cloud computing environment," *Machine learning for predictive analysis*, pp. 163--171, 2021.

[19] A. Chhabra, K.-C. Huang, N. Bacanin and T. A. Rashid, "Optimizing bag-of-tasks scheduling on cloud data centers using hybrid swarm-intelligence meta-heuristic," *The Journal of Supercomputing*, pp. 1--63, 2022.

[20] N. Bacanin, U. Arnaut, M. Zivkovic, T. Bezdan and T. A. Rashid, "Energy Efficient Clustering in Wireless Sensor Networks by Opposition-Based Ini-

tialization Bat Algorithm," *Computer Networks and Inventive Communication Technologies*, pp. 1--16, 2022.

[21] N. Bacanin, M. Antonijevic, T. Bezdan, M. Zivkovic and T. A. Rashid, "Wireless Sensor Networks Localization by Improved Whale Optimization Algorithm," *Proceedings of 2nd International Conference on Artificial Intelligence: Advances and Applications*, pp. 769--783, 2022.

[22] M. Zivkovic, T. Zivkovic, K. Venkatachalam and N. Bacanin, "Enhanced dragonfly algorithm adapted for wireless sensor network lifetime optimization," *Data intelligence and cognitive informatics*, pp. 803--817, 2021.

[23] N. Bacanin, T. Bezdan, E. Tuba, I. Strumberger and M. Tuba, "Optimizing convolutional neural network hyperparameters by enhanced swarm intelligence metaheuristics," *Algorithms*, vol. 13, no. 3, p. 67, 2020.

[24] N. Bacanin, M. Zivkovic, M. Salb, I. Strumberger and A. Chhabra, "Convolutional Neural Networks Hyperparameters Optimization Using Sine Cosine Algorithm," *Sentimental Analysis and Deep Learning*, pp. 863--878, 2022.

[25] E. Tuba, N. Bacanin, I. Strumberger and M. Tuba, "Convolutional Neural Networks Hyperparameters Tuning," *Artificial Intelligence: Theory and Applications*, pp. 65--84, 2021.