International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 697-706, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.056

Feature Selection Using Different Transfer Functions for Binary Bat Algorithm

Omar Saber Qasim
Department of Mathematics,
University of Mosul, Mosul, Iraq.
Corresponding author: omar.saber@uomosul.edu.iq

Zakariya Y. Algamal
Department of Statistics and Informatics,
University of Mosul, Mosul, Iraq.
E-mail: Zakariya.algamal@uomosul.edu.iq

(Received January 20, 2020; Accepted April 1, 2020)

Abstract
The selection feature is an important and fundamental step in the preprocessing of many classification and machine learning problems. The feature selection (FS) method is used to reduce the amount of data used and to create high-probability of classification accuracy (CA) based on fewer features by deleting irrelevant data that often reason confusion for the classifiers. In this work, bat algorithm (BA), which is a new metaheuristic rule, is applied as a wrapper type of FS technique. Six different types of BA (BA-S and BA-V) are proposed, where apiece used a transfer function (TF) to map the solutions from continuous space to the discrete space. The results of the experiment show that the features that use the BA-V methods (that is, the V-shaped transfer function) have proven effective and efficient in selecting subsets of features with high classification accuracy.

Keywords- Feature subset selection, Bat algorithm, Transfer function, Metaheuristic algorithms.

1. Introduction

Feature selection plays an important role in the classification and machine learning, as it aims to remove the irrelevant or redundant features of the data to achieve better classification performance. Several metaheuristic algorithms have been proposed in conjunction with feature selection methods to reduce data processing time and improve the classification accuracy (CA) (Qasim and Algamal, 2018; Dahiya et al., 2019). The methods of feature subset depend on the evaluation of the selected subsets through two basic approaches: filter and wrapper subset methods. The filter relies on the evaluates the subgroups based on the original dataset itself whereas the wrapper method is based on external learning algorithms based on metaheuristic algorithms to evaluate subgroups (Wang and Alexander, 2016; Teng et al., 2017; Al-Tashi et al., 2020).

Evolutionary computing (EC) is nature-inspired techniques and algorithms by mimic the behavior of insects, animals, and birds (such as bats) and depends on the population in the search for possible solutions to the problem (Valdez, 2015). EC has been used extensively in recent years to address problems of feature selection in literature (Emary et al., 2016). For example, there are feature selection approaches based on swarm algorithms (Kashef and Nezamabadi-pour, 2015) , genetic algorithm (GA) (Bhunia et al., 2017) and differential evolution (DE) (Zorarpacı and Özel, 2016). The works in this field can also be found through reference to (Wang et al., 2010; Mafarja and Abdullah, 2013a; Pramy, 2018; Qasim and Algamal, 2018; Al-Thanoon et al., 2019).

Many metaheuristic inspired by the conduct of animals, insects, and birds in nature have been proposed as good methods in the feature selection (Yang, 2010; Mafarja et al., 2020). The bat algorithm (BA) is one of the modern metaheuristic methods inspired by the social behavior of bats. BA is described by the fact that its equations are uncomplicated, therefore can be applied in many fields such as classification, feature selection, data mining, scheduling, and others (Yang and He, 2013).

Transfer functions (TFs) are important and effective parts of a binary metaheuristics algorithm when used in the classification process and feature selection (Mirjalili and Lewis, 2013), where it very affects the process of balance (between exploration and exploitation). In this research, six transfer functions, proposed in (Mafarja and Abdullah, 2013b) , were applied to the BA.

Our paper is organized as follows. The methodology of the BA and the BBA are introduced in Section 2. In section 3, the BBA for feature selection is presented. The Experimental results are listed in Section 4. The conclusion is provided in Section 5.

2. Bat Algorithm (BA)

BA is one of the new metaheuristic methods proposed by Yang in 2010 which have attracted the attention of many researchers from different fields because of their remarkable capability of echolocation. It is based on the echolocation ability of microbats guiding them on their foraging attitude (Yang, 2010). There are various types of bat that differ in size and weight but they have similar behaviors of navigation and hunting. The BA was designed based on the basic advantage of the bats when searching for prey. Bats have a strong sound propagation technique are different from other birds called bio-sonar. Bats when chasing prey tend to increase the rate of sound emitted by the ultrasound and decrease the loudness. The behavior of bats has been modeled as follows (Mirjalili et al., 2014; Yu et al., 2020). BA has a set of vectors representing the location, velocity, and frequency, which can be updated with equations (1), (2), and (3):

$$V_i^{(t+1)} = V_i^{(t)} + (X_i^{(t)} - G_{best})F_i \tag{1}$$

$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)} \tag{2}$$

where, G_{best} represents best solution, F_i is frequency of the i-th bat which is updated through the iterations as follows :

$$F_i = F_{\min} + (F_{\max} - F_{\min})X_{new} = X_{old} + \varepsilon A^t \tag{3}$$

where, β represents a random number of the uniform distribution in [0, 1]. Through Equations (1) & (3) we note that, the various frequencies of bats encourage them to have a tendency to the solution. However, exploitation was carried out through the use of a random walk procedure as follows (Anter & Ali, 2020):

$$X_{new} = X_{old} + \varepsilon A^t \tag{4}$$

where, $\boldsymbol{\varepsilon}$ indicates a random number in [-1,1], and A represents the loudness of emitted sound that bats use. The pulse emission rate $\boldsymbol{r}$ and the loudness A are updated by controlling the balance between these techniques as follows:

$$A_i^{(t+1)} = \alpha A_i^{(t)} \tag{5}$$

$$r_i^{(t+1)} = r_i(0)[1-\exp(-\gamma t)] \tag{6}$$

where α and γ are constants; t is analogous to the cooling factor, lastly, A_i is equal to zero as well as the last value of r_i is $r(0)$. The update of the loudness and rate guarantee that artificial bats move towards the best solutions (Tharwat et al., 2017). The pseudocode of BBA is presented in Figure 1.

```
Start
    Create an initial population of the BA { $x_i$ ,i=1,2,..,n }, $F_i$ , $r_i$ ,and $A_i$ .
  while (t < Max number. of iterations).
      Generate new solutions by updating the following positions and velocities
      in addition to adjusting frequency from Equations (1), (2) and (3).
       If  (rand > $r_i$ )
          Select a random value solution (between the best solutions).
          Create a local solution.
      end if
          Generate a new solution using randomly fly.
             If (rand < $r_i$ & f( $x_i$ )< f($G_{best}$ )).
          Chose  the new solution and reduce $A_i$ and increase $r_i$ .
      end if
      locate the current $G_{best}$ and rank the bats.
  end while
End
```

Figure 1. The pseudocode of the BA

2.1 Binary Bat Algorithm (BBA)

The BBA is conceptually similar to the main bat algorithm (BA) but differs from it in the search space. In the main BA, the search is in continuous space, while in the binary algorithm is searched in the discrete binary space (Mirjalili et al., 2014). The velocity and position in the BBA algorithm does not change because the search space is constrained by 0 and 1. The binary search can be assumed to be a hypercube, In the BBA, it is possible only to shift to the corners of this hypercube by reversing the different bit numbers. Position update in discrete spaces is done by switching the values "0" and "1"(Naik et al., 2020). The position of the agent is changed in discrete space with the probability of its velocity by using a transfer function to map velocity values to probability values (Kennedy and Eberhart, 1997; Mirjalili and Hashim, 2012). This means that the vector values of the position are changing from 0 to 1 and vice versa, through the transfer function, which gives the force to move particles in a binary space. To select the transfer function, consider the following concepts (Rashedi et al., 2010).

- The transfer functions (TFs) must be restricted in [0,1] which is explains the change of position.
- The TFs should supply a great possibility to change the position for a large absolute value of the velocity, because it may be far from the best solution.
- The TFs should supply a small probability to change the position.
- The resumption values of the TFs should rise as the value of the velocity increases.
- The resumption values of the TFs should decrease as the value of the velocity reduces.

Through previous concepts, the search is transformed from continuous space into discrete binary space by the transfer functions (Mohamad et al., 2010; Teng et al., 2017). The selected transport function is defined as follows:

$$V(v_t^k(t)) = \tanh(v_t^k(t)) \tag{7}$$

The probability in the position of an agent in BBA is given as follows:

$$X_t^k(t+1) = \begin{cases} (X_t^k(t))^{-1} & if \quad rand < V(v_t^k(t+1)) \\ X_t^k(t) & if \quad rand \geq V(v_t^k(t+1)) \end{cases} \tag{8}$$

where, $X_t^k(t)$ is position and $v_t^k(t)$ is the velocity of the i^{th} agent in k^{th} dimension at the iteration t . The pseudocode of BBA is presented in Figure 2.

```
Start
     Create an initial population of the BA, { $x_i$ ,i=1,2,..,n }, $F_i$ , $r_i$ , and $A_i$ .
   while (t < Max number. of iterations).
          Updating velocities and adjusting frequency.
          Transfer function values are calculated by Equation (7).
          Updating positions from  Equation (8).
         if  (rand > $r_i$ )
           Select a random value solution and create a local solution.
        end if
           Generate a new solution using randomly fly.
              If ( rand < $r_i$ & f( $x_i$ )< f($G_{best}$ )).
           Chose  the new solution and reduce $A_i$ and increase $r_i$ .
        end if
        locate the current $G_{best}$ and rank the bats.
   end while
End
```

Figure 2. The pseudocode of the BBA

3. Binary Bat Algorithm for Feature Selection

The feature selection depends on the representation of any solution as a binary variable of a N-sized, where N representing the total number of features in the dataset. Meta-heuristics techniques are used in feature selection issues to overcome the problem of generating all possibilities for feature combinations calculated by the equation (Mirjalili and Lewis, 2013; Nakamura et al., 2013):

$$AF = 2^N \tag{9}$$

where, AF indicate the all possible feature in the dataset.

BA is one of the most important metaheuristics algorithms to get the best feature subset by searching for a feature space well and effectively. Each subset of the dataset is evaluated by classification accuracy (CA) obtained from the selected features. The fitness function (Fit) is used to evaluate subgroups according to the accuracy of the classification as follows:

$$Fit = \omega_1 \times CA + \omega_2 \frac{|R|}{N} \tag{10}$$

where, $\omega_1 \in [0\ 1]$, $\omega_2 = 1 - \omega_1$, CA represents the classification, $|R|$ and N respectively are the selected and total features in the dataset (Emary et al., 2016; Alhafedh & Qasim, 2019).

When a feature selection strategy is used in the binary bat algorithm (BBA), the working mechanism of the algorithm is transformed from continuous space into discrete space by applying transfer functions (TFs), which belong to two main categories (V-shaped and S-shaped). The possibility of updating a binary solution (from 0 to 1) and vice versa is done through TFs. In BA, solutions are updated using S-shaped functions based on the following equation (Munshi and Mohamed, 2017; Al-Thanoon et al., 2018):

$$X_t^k(t+1) = \begin{cases} 1 & if \quad rand < SF(v_t^k(t+1)) \\ 0 & if \quad rand \geq SF(v_t^k(t+1)) \end{cases} \tag{11}$$

Also, by using the V-shaped functions, solutions are updated based on the following equation:

$$X_t^k(t+1) = \begin{cases} (X_t^k(t))^{-1} & if \quad rand < VF(v_t^k(t+1)) \\ X_t^k(t) & if \quad rand \geq VF(v_t^k(t+1)) \end{cases} \tag{12}$$

where, $X_t^k(t)$ and $v_t^k(t)$ indicate the position and velocity of the i^{th} agent in k^{th} dimension at the iteration t also SF represent TF of type S-shaped , and VFis TF of type V-shaped. The mathematical representation of BA-V shaped families are defined as:

$$BA - V\,1 = TF(x) = \left| erf\left(\frac{\sqrt{\pi}}{2} x\right) \right| ,\ erf\,(x) = \frac{2}{\pi}\int_0^x e^{-t^2} dt \tag{13}$$

$$BA-V2=TF(x)=\left|\frac{x}{\sqrt{1+x^{2}}}\right| \tag{14}$$

$$BA-V3=TF(x)=\left|\frac{2}{\pi}\arctan\left(\frac{2}{\pi}x\right)\right| \tag{15}$$

The mathematical representation of BA-S shaped families are defined as:

$$BA-S1=TF(x)=\frac{1}{1+e^{-2x}} \tag{16}$$

$$BA-S2=TF(x)=\frac{1}{1+e^{\frac{-x}{2}}} \tag{17}$$

$$BA-S3=TF(x)=\frac{1}{1+e^{\frac{-x}{3}}} \tag{18}$$

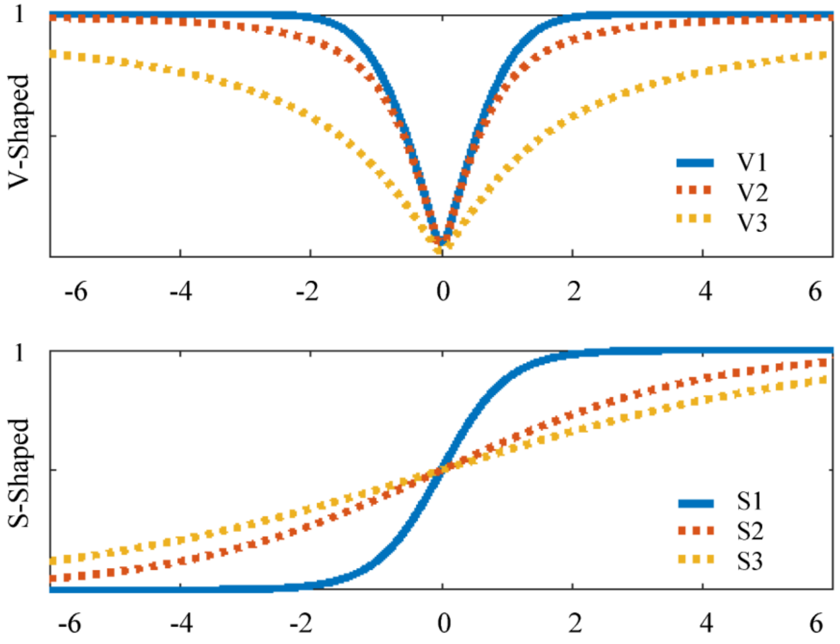

Figure 3. Transfer functions of V-shaped & S-shaped

In Figure 3, six transfer functions (TF$_S$) are used in BA. The three algorithms that use S-shaped of TF$_S$ are named BA-S1, BA-S2, and BA-S3, while the other three algorithms that use V-shaped of TF$_S$ are named BA-V1, BA-V2, and BA-V3 (Beheshti, 2020).

4. Experimental Results and Discussion

Three binary class datasets: antimicrobial agents (pMIC) (Xing et al., 2014), anti-hepatitis C virus activity (Kang et al., 2009a; Kang et al., 2009b; Kang et al., 2010; Khatri et al., 2015) and neuraminidase inhibitors of influenza A viruses (H1N1) (Li et al., 2016), were used for evaluating the suggested algorithm. A brief attributive of the data from the point of view of the number of samples, number of features and class are summarized in Table 1. The datasets have been invested to evaluate the efficiency of the proposed method. Every dataset was split into training dataset which consist of 70% of the total number of samples and testing datasets consisting of 30% of the samples. All algorithms (BA-V and BA-S) were carried out with the Matlab programming and processed on a Quad-Core Intel Core i5 CPU (4G RAM and 3.2GHz processor).

Table 1. Description of the used datasets

Dataset	#Samples	#features	Class
Antimicrobial	212	3657	104 /108 (inactive / active) compounds
Hepatitis	121	2559	31 / 90 (active / inactive)
H1N1	479	2322	266 / 213 (weakly active / active)

Table 2. Methods performance evaluation (on average) according to the training datasets with the standard deviation is between parentheses

Datasets	Antimicrobial		Hepatitis		H1N1	
	training		training		Training	
Functions	CA	#features	CA	#features	CA	#features
BA-S1	91.823 (0.148)	11	94.511 (0.267)	9	91.747 (0.779)	13
BA-S2	88.281 (0.146)	14	90.147 (0.839)	15	90.254 (1.686)	18
BA-S3	83.263 (0.169)	19	86.765 (0.956)	17	84.103 (0.484)	20
BA-V1	90.308 (0.098)	13	91.223 (1.038)	12	89.095 (1.342)	16
BA-V2	93.114 (0.191)	9	96.113 (1.063)	8	93.061 (0.945)	12
BA-V3	87.574 (0.155)	21	89.775 (0.948)	21	88.019 (0.281)	25

Table 3. Methods performance evaluation (on average) according to the testing datasets with the standard deviation is between parentheses

Datasets	Antimicrobial	Hepatitis	H1N1
	testing	testing	testing
Functions	CA	CA	CA
BA-S1	89.841 (1.850)	92.343 (0.917)	90.611 (0.694)
BA-S2	88.146 (0.685)	89.299 (1.437)	88.386 (1.475)
BA-S3	81.349 (0.406)	84.854 (0.534)	82.069 (0.864)
BA-V1	89.673 (1.057)	89.737 (0.978)	87.337 (0.480)
BA-V2	92.402 (1.279)	94.225 (1.534)	90.518 (0.773)
BA-V3	87.022 (0.447)	85.880 (0.520)	86.275 (0.645)

Tables 2 and 3 show the obtained results for BA-V1, BA-V2, BA-V3, BA-S1, BA-S2,, and BA-S3, respectively, in terms of classification accuracy (CA) for both the training and testing datasets (in Table 1) . For comparison, the last column "# features" presents the average number of selected features for each transfer function. The results in Table 2 shows that by the cooperation of the situation with feature selection, the classification accuracy (CA) for all transfer functions (TFs) was greatly improved and the feature dimensionality reduced. The results exhibit that the methods have high efficiency in classifying data. Moreover, BA-V2 and BA-S1 functions attain typical results in a reduction of the features number, which are significantly different from the others.

5. Conclusion

In this paper, several transfer functions (Equations 13-18) have been investigated. To demonstrate the efficiency and effectiveness of transfer functions (V-shape and S-shape), they were used in the binary bat algorithm and made three comparisons of the datasets (in Table 1) by relying on two main scenarios: the number of features specified in each algorithm and the classification accuracy (CA). The experimental results in Tables 2 and 3 have shown that the BA-V2 algorithm has consistency in the feature selection process and that it outperforms other algorithms through the accuracy of the classification results.

Conflict of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

The authors appreciate the effort from editors and reviewers. This paper did not receive any financial assistance/grants from any funding agencies. The authors sincerely appreciate the editor and reviewers for their time and valuable comments.

References

Al-Tashi, Q., Md Rais, H., Abdulkadir, S.J., Mirjalili, S., Alhussian, H. (2020). A review of grey wolf optimizer-based feature selection methods for classification. In: Mirjalili S., Faris H., Aljarah I. (eds) *Evolutionary Machine Learning Techniques. Algorithms for Intelligent Systems*. Springer, Singapore. pp 273-286.

Al-Thanoon, N.A., Qasim, O.S., & Algamal, Z.Y. (2019). A new hybrid firefly algorithm and particle swarm optimization for tuning parameter estimation in penalized support vector machine with application in chemometrics. *Chemometrics and Intelligent Laboratory Systems*, *184*, 142-152.

Al-Thanoon, N.A., Qasim, O.S., & Algamal, Z.Y. (2018). Tuning parameter estimation in SCAD-support vector machine using firefly algorithm with application in gene selection and cancer classification. *Computers in Biology and Medicine*, *103*, 262-268.

Alhafedh, M.A.A., & Qasim, O.S. (2019). Two-stage gene selection in microarray dataset using fuzzy mutual information and binary particle swarm optimization. *Indian Journal of Forensic Medicine & Toxicology*, *13*(4), 1162-1171.

Anter, A.M., & Ali, M. (2020). Feature selection strategy based on hybrid crow search optimization algorithm integrated with chaos theory and fuzzy c-means algorithm for medical diagnosis problems. *Soft Computing*, *24*(3), 1565-1584.

Beheshti, Z. (2020). A time-varying mirrored S-shaped transfer function for binary particle swarm optimization. *Information Sciences*, *512*, 1503-1542.

Bhunia, A.K., Duary, A., & Sahoo, L. (2017). A genetic algorithm based hybrid approach for reliability redundancy optimization problem of a series system with multiple-choice. *International Journal of Mathematical, Engineering and Management Sciences*, *2*(3), 185-212.

Dahiya, B.P., Rani, S., & Singh, P. (2019). A hybrid artificial grasshopper optimization (HAGOA) meta-heuristic approach: a hybrid optimizer for discover the global optimum in given search space. *International Journal of Mathematical, Engineering and Management Sciences*, *4*(2), 471-488.

Emary, E., Zawbaa, H.M., & Hassanien, A.E. (2016). Binary ant lion approaches for feature selection. *Neurocomputing*, *213*, 54-65.

Kang, I.J., Wang, L.W., Hsu, S.J., Lee, C.C., Lee, Y.C., Wu, Y.S., Yueh, A.,Wang, J.C., Hsu, T.A., Chao, Y.S., & Chern, J.H. (2009a). Design and efficient synthesis of novel arylthiourea derivatives as potent hepatitis C virus inhibitors. *Bioorganic & Medicinal Chemistry Letters*, *19*(21), 6063-6068.

Kang, I.J., Wang, L.W., Lee, C.C., Lee, Y.C., Chao, Y.S., Hsu, T.A., & Chern, J.H. (2009b). Design, synthesis, and anti-HCV activity of thiourea compounds. *Bioorganic & Medicinal Chemistry Letters*, *19*(7), 1950-1955.

Kang, I.J., Wang, L.W., Yeh, T.K., Lee, C.C., Lee, Y.C., Hsu, S.J., Wu, Y.S., Wang, J.C., Chao, Y.S., Yueh, A., & Chern, J.H. (2010). Synthesis, activity, and pharmacokinetic properties of a series of conformationally-restricted thiourea analogs as novel hepatitis C virus inhibitors. *Bioorganic & Medicinal Chemistry*, *18*(17), 6414-6421.

Kashef, S., & Nezamabadi-Pour, H. (2015). An advanced ACO algorithm for feature subset selection. *Neurocomputing*, *147*, 271-279.

Kennedy, J., & Eberhart, R.C. (1997, October). A discrete binary version of the particle swarm algorithm. In *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation* (Vol. 5, pp. 4104-4108). IEEE. Orlando, USA.

Khatri, N., Lather, V., & Madan, A.K. (2015). Diverse classification models for anti-hepatitis C virus activity of thiourea derivatives. *Chemometrics and Intelligent Laboratory Systems*, *140*, 13-21.

Li, Y., Kong, Y., Zhang, M., Yan, A., & Liu, Z. (2016). Using support vector machine (SVM) for classification of selectivity of H1N1 neuraminidase inhibitors. *Molecular Informatics*, *35*(3-4), 116-124.

Mafarja, M., & Abdullah, S. (2013a). Comparison between record to record travel and great deluge attribute reduction algorithms for classification problem. In *International Multi-Conference on Artificial Intelligence Technology* (pp. 111-120). Springer, Berlin, Heidelberg.

Mafarja, M., & Abdullah, S. (2013b). Investigating memetic algorithm in solving rough set attribute reduction. *International Journal of Computer Applications in Technology*, *48*(3), 195-202.

Mafarja, M., Heidari, A.A., Faris, H., Mirjalili, S., Aljarah, I. (2020). dragonfly algorithm: theory, literature review, and application in feature selection. In: Mirjalili S., Song Dong J., Lewis A. (eds) *Nature-Inspired Optimizers. Studies in Computational Intelligence*. Springer, Cham, pp. 103-116.

Mirjalili, S., & Hashim, S.Z.M. (2012). BMOA: binary magnetic optimization algorithm. *International Journal of Machine Learning and Computing*, *2*(3), 204-208.

Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, *9*, 1-14.

Mirjalili, S., Mirjalili, S.M., & Yang, X.S. (2014) Binary bat algorithm. *Neural Computing & Application*, *25*, 663–681. doi:10.1007/s00521-013-1525-5.

Mohamad, M.S., Omatu, S., Deris, S., & Yoshioka, M. (2010). Particle swarm optimization with a modified sigmoid function for gene selection from gene expression data. *Artificial Life and Robotics*, *15*(1), 21-24.

Munshi, A.A., & Mohamed, Y.A.R.I. (2017). Comparisons among bat algorithms with various objective functions on grouping photovoltaic power patterns. *Solar Energy*, *144*, 254-266.

Naik, A.K., Kuppili, V., & Edla, D.R. (2020). Efficient feature selection using one-pass generalized classifier neural network and binary bat algorithm with a novel fitness function. *Soft Computing*, *24*(6), 4575-4587.

Nakamura, R.Y.M., Pereira, L.A.M., Rodrigues, D., Costa, K.A.P., Papa, J.P., & Yang, X.S. (2013). Binary bat algorithm for feature selection. In *Swarm Intelligence and Bio-Inspired Computation* (pp. 225-237). Elsevier. doi:10.1016/b978-0-12-405163-8.00009-0.

Pramy, F.A. (2018). An approach for solving fuzzy multi-objective linear fractional programming problems. *International Journal of Mathematical, Engineering and Management Sciences*, *3*(3), 280-293.

Qasim, O.S., & Algamal, Z.Y. (2018). Feature selection using particle swarm optimization-based logistic regression model. *Chemometrics and Intelligent Laboratory Systems*, *182*, 41-46.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2010). BGSA: binary gravitational search algorithm. *Natural Computing*, *9*(3), 727-745.

Teng, X., Dong, H., & Zhou, X. (2017). Adaptive feature selection using v-shaped binary particle swarm optimization. *PloS one*, *12*(3). e0173907. doi: 10.1371/journal.pone.0173907.

Tharwat, A., Hassanien, A.E., & Elnaghi, B.E. (2017). A BA-based algorithm for parameter optimization of support vector machine. *Pattern Recognition Letters*, *93*, 13-22.

Valdez, F. (2015). Bio-inspired optimization methods. In: Kacprzyk J., Pedrycz W. (eds) *Springer Handbook of Computational Intelligence*. Springer, Berlin, Heidelberg.

Wang, J., Li, T., & Ren, R. (2010, August). A real time IDSs based on artificial bee colony-support vector machine algorithm. In *Third International Workshop on Advanced Computational Intelligence* (pp. 91-96). IEEE. Suzhou, China.

Wang, L., & Alexander, C.A. (2016). Machine learning in big data. *International Journal of Mathematical, Engineering and Management Sciences*, *1*(2), 52-61.

Xing, J.J., Liu, Y.F., Li, Y.Q., Gong, H., & Zhou, Y.P. (2014). QSAR classification model for diverse series of antimicrobial agents using classification tree configured by modified particle swarm optimization. *Chemometrics and Intelligent Laboratory Systems*, *137*, 82-90.

Yang, X.S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies For Optimization (NICSO 2010)* (pp. 65-74). Springer, Berlin, Heidelberg.

Yang, X.-S., & He, X. (2013). Bat algorithm: literature review and applications. *International Journal of Bio-Inspired Computation*, *5*(3), 141-149.

Yu, H., Zhao, N., Wang, P., Chen, H., & Li, C. (2020). Chaos-enhanced synchronized bat optimizer. *Applied Mathematical Modelling*, *77*, 1201-1215.

Zorarpacı, E., & Özel, S.A. (2016). A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Systems with Applications*, *62*, 91-103.