Classification and Feature Selection Method for Medical Datasets by BGEO TVFL(Binary golden eagle optimization-Time Varying Flight length) and KNN(k-nearest neighbour)

Rama Krishna Eluri, Lakshmi Prasanna Bommu, Swetha Lakshmi Paruchuri, sahithi Mellacheruvu

Abstract: Classification accuracy and feature selection are important steps in medical data analysis to identify suitable features that can improve the performance of machine learning models In this study, we present a new method called BGEO TVFL (Binary Golden Eagle Optimization-Time Varying Flight Length) algorithm together is proposed for its K-nearest neighbour (KNN) algorithm. The TVFL algorithm feature of BGEO is used for optimal subset selection, while the KNN algorithm is used for classification. The proposed method is tested in several projects. The experimental results show that the proposed method outperforms other existing methods such as BWOA, BGWO, ACO, and ABC in terms of accuracy and selected features. Our results show that BGEO TVFL outperforms other algorithms in terms of accuracy and feature selection, achieving higher classification accuracy and selecting fewer features compared to how BGEO TVFL performs well as a good optimization algorithm for feature selection and classification in medical data sets

Keywords: Feature Selection, BGEO (Binary Golden Eagle Optimization), Classification, TVFL (Time Varying Flight Length), Wrapper Method

1. INTRODUCTION

Feature selection is an important step in machine learning [1], especially in medical datasets where the number of features may be too large and irrelevant features may lead to poor model performance Feature selection [2] identifies relevant informative features of large candidates and poor generalization. The selection problem has been extensively studied in a variety of fields including medicine[3,4], where the complexity of medical data sets requires effective and efficient selection strategies, which may not be effective in tight correlation considering the differences between the components of the medical database. To address this limitation, this study proposed a new selection method called binary Golden Eagle optimization-time-varying flight length (BGEO-TVFL), with time-varying capability of binary optimization methods can optimize the feature selection process[5, 6] -By taking advantage of the K-Nearest Neighbour (KNN) algorithm as a classifier combining flight lengths, we show that BGEO-TVFL outperforms other state-of-the-art approaches optimize features such as binary whale optimization algorithm (BWOA), binary grey wolf optimization (BGWO), Ant Colony Optimization (ACO), and Artificial Bee Colony (ABC) in feature selection accuracy and computational efficiency. However, their performance should be evaluated against other optimization algorithms for feature selection. In this paper, we propose a new BGO called Binary Golden Eagle Optimization-Time Varying Flight Length (BGEO TVFL), which incorporates time-varying flight length to improve the search process We use BGEO TVFL performance compare for options in the medical lists BWOA, BGWO, . Let’s do it with ACO, and the Artificial Bee Colony (ABC) algorithm. Our results show that BGEO TVFL outperforms other algorithms in terms of accuracy and feature selection [7,8].

CONTRIBUTIONS

- Perform BGEO-TVFL algorithm by using KNN(K-Nearest Neighbour) model to find the no of features selected and accuracy on medical datasets
- Perform BWOA algorithm by using KNN (K-Nearest Neighbour) model to find the no of features selected and accuracy on medical datasets
- Perform BGWO algorithm by using KNN(K-Nearest Neighbour) model to find the no of features selected and accuracy on medical datasets
- Perform ACO algorithm by using KNN(K-Nearest Neighbour) model to find the no of features selected and accuracy on medical datasets
- Perform ABC algorithm by using KNN(K-Nearest Neighbour) model to find the no of features selected and accuracy on medical datasets
- At last we conclude that BGEWO-TVFL algorithm is best compared to remaining all algorithms

2. RELATED WORK

ShenkaiGu and others. [9] Proposed a new feature selection method which combined a Competitive Swarm Optimizer (CSO) and a clustering method to reduce the computation cost. This method applied for selected features of the concentration decreased, resulting in better classification performance. However, the authors noted that this approach still needs to be optimized to achieve the best balance between feature selection and classification performance.

R. K. Agarwal and others. [10] Proposed a wrapper feature selection method based on Quantum Wolf Algorithm Optimization (Quantum WOA), which combined quantum concept with WOA to enhance exploitation and detection efficiency Crossover-optimized mutation using quantum rotary gate and quantum bit representation uses. Q-bits improved with operators, and overall performance increased but this approach was limited to solving single-objective optimization problems and did not extend to continuous improvement or multi-objective optimization problems

Xian-fang Song et al. [11] present an efficient feature selection method that combines bare bone particle swarm optimization (BBPSO) and mutual information to solve high-dimensional feature selection problems Method was initiated by a swarm initialization model that used label correlation to enhance convergence. Local search operators two deletion and complement operators were developed to determine feature relevance and redundancy and improve exploitation performance Furthermore, an adaptive flip mutation operator was added to help particles escape local optima and find optimal solutions. The BBPSO method was able to obtain the feature subset with improved performance, although it took longer to complete the feature selection task compared to the other methods

M. S. Uzer, N. Yilmaz, and O. Inan [12] presented an artificial bee swarm-based feature selection method for the diagnosis of liver diseases, cirrhosis, and diabetes by Support Vector Machine (SVM) [8]. Distribution was proposed Using the - algorithm. Notably, the SVM model was not optimized. The data sets were divided into two sets: one for training the SVM model and the other for testing the quality of the obtained model. The results were compared with methods from the literature, and the proposed method showed improved performance.

J. Too and S. Mirjalili, [13] introduced an improved binary dragonfly algorithm (BDA) to extract local optima from wrapper-based approach for feature selection optimization problem Proposed algorithm, called Hyper Learning Binary Dragonfly Algorithm (HLBDA). , improves search behaviour using hyper learning method.

3. Overview of Binary Golden Eagle Optimizer

The Binary Golden Eagle Optimizer (BGEO) is a novel optimization algorithm inspired by the foraging behaviour of the golden eagle[14], which forages in a specific location, using good resources and avoiding bad This algorithm uses binary vectors for representation for the solution, Where each bit can take the value of 0 or 1, which provides efficient and user-friendly computation Algorithm regenerates these binary vectors based on fitness function and selection rules, balanced in detection and exploitation using probability-based mechanisms Robust performance[15] in optimization problems has been demonstrated, making it a promising tool for engineering development and machine learning applications[16 ,17].The hunting style of the golden eagle has several key features:

- First, when it is looking for help, it takes a spiral path and moves in a straight line when it attacks.
- In the early stages of a hunt, a golden eagle tends to wander, indicating a higher chance of finding prey. However, as the hunt progresses, it shifts to an increased tendency to attack, suggesting a more focused strategy
- Notably, the golden eagle can switch between attack modes each time it files, demonstrating its flexibility and versatility.
- In addition, the golden eagle also participants in information gathering by seeking knowledge about the food sources of other eagles.

4. PROPOSED METHODOLOGY

We use BGEO-TVFL (Binary Golden Eagle Optimizer-Time Varying Flight Length) algorithm to solve Feature Selection problem. The Binary Golden Eagle Optimization (BGE) algorithm is then applied to generate a number of binary vectors, each representing a possible feature subset, using a fitness function based on a K-Nearest Neighbour (KNN) classifier time-varying flight length (2010). TVFL) parameter inserted is used BGE-algorithm is to vary the length of each flight (iteration) based on the current population health, and enable the algorithm to navigate the required complex areas and search areas This procedure can have found more solutions by increasing the flight length as fitness improves, also By decreasing it as fitness stops, the algorithm is able to intensify the search around promising regions BGE algorithm perform runs for a specified number of generations or until the stopping criterion is met, after which the top features are selected based on the frequency of appearance from the final population , evaluated using different metrics, including accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC) The performance of the trained classifier is evaluated using both training and testing datasets, whose distribution is obtained comprehensive power analysis This new approach is expected to improve classification performance and robustness. The BGEO solution is described by the e dimensional vector Y where Y represents as a continuous value vector then the new locations of the attributes is updated by the following equation:

$$y^{u+1} = y^{u} + \Delta y_i^{u} \quad (1)$$

Where, $\Delta {y_j}^{u}$represents the j phase vector of features at iteration u

$$\Delta y^{u} = \underset{t}{\rightarrow} \cdot_1 TF_v \frac{\vec{A}^k}{j} + \underset{t}{\rightarrow} . r . \frac{\vec{D}^k}{2l} \underset{A}{\rightarrow}{}_k \underset{D}{\rightarrow}{}_k \quad (2)$$

Then [3] wrapper-based feature selection reduces the number of selected features and increases the learning algorithm’s performance. Every solution is determined by the foundation fitness function using a KNN classifier as an evaluator and considers the number of selections in the solution. So, to strike a balance between classification accuracy (maximum) is the number of (minimum) selected elements in each solution. The fitness can be defined by two objectives (I.e. minimum number of selected items and maximum classification accuracy) [47], and the fitness function is defined as below:

$$Fitness(Y) = \beta\lambda_s(Y) + \eta\frac{|Y|}{|M|} \quad (3)$$

Where, λS (Y) denotes the classification error rate, |M| means that the transfer function is used to determine the optimal result. Furthermore, the transfer function is considered to have an integral and important function in the derivation of the BGEO_TVFL algorithm changing the transfer function can change the performance and the total number of features in data set, defines the selected number of features by Y, denoting the weights β and η classification error rate and selection coefficient, β 0, 1. The relationship between β and η indicating η (1 β). It is well preserved. Last but not least are the steps for the proposed BGEO-TVFL described here in the number of items originally produced. A set of data is obtained and represented by e. Next, a 2D array of size M, e is determined, and its value is defined as 0 or 1. But M defines the number of attributes when e is equal to the number of objects found in the dataset. The feature vector is designed to look like any row. Which is based on the feature vector is presented for the purpose. Moreover, the memory fitness of each GE or feature can be calculated and initialized. Now, the process is just repeated for Max_iter multiple times. For all iterations, the state of the GE was updated, then the validated position, which indicates eligibility for the relevant position, was updated accordingly and finally the rest of the GE was updated accordingly. The BGEO-TVFL pseudo code is given as below

Algorithm 1: Time-varying flight length and binary Golden Eagle optimizer
set the feature population to start.
Identify the fitness function.
Set up the position initially.
Set each memory's population to zero.
Set F to initial, update F for each iteration v, and for each GE_j.
Choose a trait at random from the populations' memory
Calculate AV
If the length of AV is not zero
Establish your CV.
Utilizing equation (1) calculate the step vector Δy_j.
Equation (2) is used to update the new position.
Equation (3) is used to convert to binary.
Calculate the fitness function at the new position by utilizing equation (4) Once the fitness function has outperformed the memory location in GE_j.
Replace the current location with the one stored in GE_j's memory.
End
End
End
End

The ability of the BGEO-TVFL algorithm to efficiently explore the solution space and adapt to the changing fitness states is due to the unique combination of plane dynamics and binary encoding -It can be highly adjustable, so as not to hit the local optimal, and converge to global optimum Binary encoding scheme makes the algorithm more representable and efficient in the solution space, in order to search the search space more efficiently and also reduces the risk of early convergence f, the power of the algorithm is less from experience and adapting to other cases makes it a promising optimization technique for real-world applications such as complex system optimization, machine learning, and industrial design. Time varying flight length and binary Golden Eagle (GE) optimizer is a new algorithm that combines the concept of time varying flight length and Golden Eagle optimizer to solve complex optimization problems Feature that the algorithm starts with a fitness function -Puts the population of the location is then initialized, and the number of people in each memory is set to zero. The algorithm repeats each iteration, v, randomly selecting an element from the remaining j population for each Golden Eagle (GE). The algorithm calculates the Average Velocity (AV) and if it is not zero, updates the location using Equations (2) and (3), binary transforms the new location with Equation (3) and then evaluates the fitness function at the new location , if and If the memory space of GE j is exceeded, the current space is replaced by the space reserved in the memory of GE j This process is repeated for each iteration and GE, allowing the algorithm to adapt to it corresponds to the optimization problem and converges to the optimal solution.

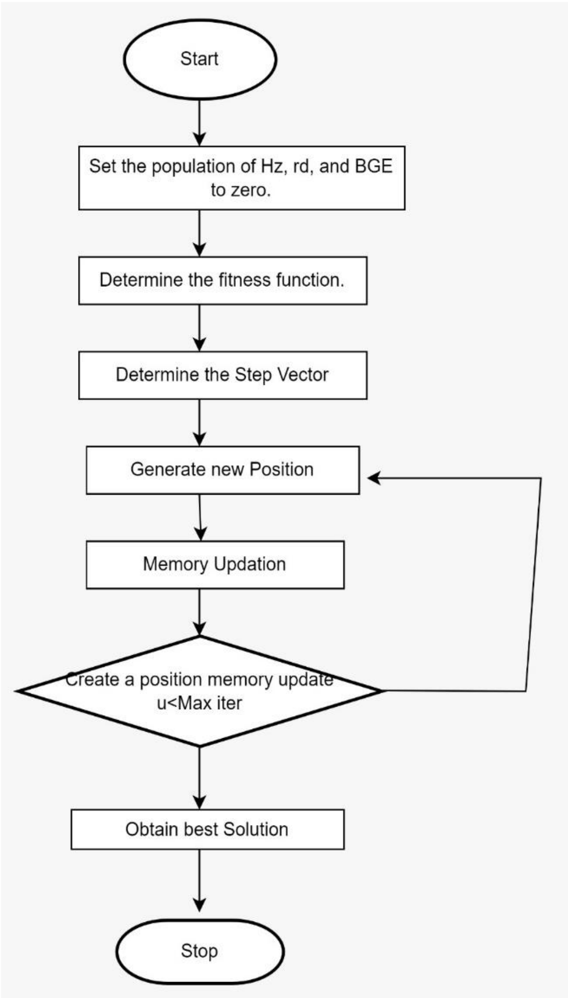


Fig: Flowchart for BGEO-TVFL

The BGEO-TVFL algorithm initializes the Hz, rd, and BGE values to zero, then determines the fitness function, which evaluates the goodness of each solution in the population Calculating the Step Vector, which determines the direction and magnitude of each search term inside Generates new locations, searching the solution space efficiently The Memory Update mechanism is used to update the position memory, which stores the best solution found so far, and repeat this process for the maximum number of iterations, a Max_Iter Updates Specified, iterations continue until the maximum number of iterations is reached Finally, the algorithm finds and prunes the best solution found in the optimization process, using binary coding to find the solution reach the most challenging areas through the principles of aviation energy and optimize the changing state of fitness

5. Experimental result and discussion

The proposed BGEO_TVFL and KNN model was evaluated on a number of medical cases to evaluate its performance in feature selection and classification accuracy. The following cases were used for experiments.

Table 1. Dataset Description

Shows the characteristics of the five datasets used in the analyses. The table includes the number of instances, features, and classes in each data set.

S.no	**Dataset**	**samples**	**Features**	**Class**
01	breast cancer	117	9	2
02	heart	304	13	2
03	diabetes	769	8	2
04	Iris	151	4	2
05	thyroid_csv	3153	23	2

The five lists in the table 1 show a variety of symptoms and complications. breast_cancer.csv dataset, with 117 observations and 9 objects, is a classic example of a binary classification problem, where the objective is to determine whether or not a patient has breast cancer Conversely, heart .csv dataset, with 304 samples and 13 features, is a multiclass classification problem , where the objective is to predict the presence of one of three types of cardiovascular disease Data set diabetes.csv, with 769 samples and features 8, is another binary classification problem, where the objective is to determine whether a patient is diabetic or not 151 samples with 4 features iris. The csv dataset is a wonderful example of a multi-classification problem, where the goal is to identify one of the three types of iris flowers based on their physical characteristics 3153 samples with 23 features The thyroid_csv.xlsx dataset is a large-scale binary classification problem To do, each data set poses a unique challenge to machine learning algorithms, which require customized models and methods to accurately classify the data. The results of the experiments are presented in Table, which shows the number of features selected by the BGEO-TVFL algorithm for each data set, and the corresponding classification accuracy obtained by KNN classifier.

Table 2. Outcomes of the proposed BGEO-TVFL.

S. no	Dataset	Classification accuracy	Computational Time
01	breast cancer	91.66	5sec
02	heart	90.16	7sec
03	diabetes	88.9	10sec
04	Iris	1	3sec
05	thyroid_csv	96.03	120sec

As shown in Table 2, the proposed BGEO-TVFL method exhibits impressive results. The Table provides a detailed overview of the classification accuracy, average efficiency values, and computational performance of the method in all 5 datasets it is worth noting that the proposed method achieves high accuracy, average efficiency and faster computation time due to better balance of detection and exploitation As seen in the results, the BGEO-TVFL-KNN model was able to select some small features that resulted in high classification accuracy in each dataset The number of features selected varied across the datasets, but in in general, it was able to identify the most relevant factors contributing to classification performance The results of the experiments show the effectiveness of the BGEO-TVFL-KNN model to achieve high classification accuracy on medical data sets by selecting appropriate features. Combining the binary golden eagle optimization (BGEO) algorithm with time-varying flight length (TVFL) and K-Nearest Neighbour (KNN) classification has proven to be a powerful method for objects election and classification. The BGEO-TVFL algorithm was able to optimize the length (reconstruction) of each flight based on the current population fit, providing robust survey analysis and spatial analysis This allowed some of the most appropriate subgroups to be selected for the classification task The results also show that the BGEO-TVFL-KNN model can generalize well to different data sets, with high classification accuracies obtained on different medical data sets This is an important finding, as it shows that power of this method can be extended to various. Treatments will increasingly use databases. In conclusion, the experimental results show the effectiveness of the BGEO-TVFL-KNN model to achieve high classification accuracy on medical data sets by selecting appropriate features. The ability of the model to adjust its search criteria appropriately and select a subset of features that contribute to greater classification accuracy makes it a promising approach for medical data analysis and decision making.

5.1 Performance Metrics

The following performance specifications are tested in each validated run:

5.1.1 Average distribution accuracy:

This metric measures the correctness of the values assigned by the classifier to selected features. The algorithm executes in M iterations.

$$Average_{accuracy} = \frac{1}{M}\sum_{m}^{m} Average_{accuracy}^{m} \quad (4)$$

5.1.2 Mean Fitness Function:

The average fitness function, also known as the calculated average fitness function, is calculated as the average fitness function value obtained after running the algorithm M iterations.

$$Mean_{ff} = \frac{1}{M}\sum_{m=1}^{m} h_m^* \quad (5)$$

Where, h_m^*run describes the fitness value obtained m times.

5.1.3 Worst Fitness Function:

The most risky objective function requires the maximum reward at iteration M times, shown as follows.

$$Worst_{ff} = \underset{m}{Max}\, h_m^* \quad (6)$$

Where, h_m^* denotes the maximum (worst) fitness value obtained during m runs.

5.1.4 Average Selected Features:

The average choice describes the total number of choices during run M. Also shown by the following equation

$$Avg.Selec = \frac{1}{M}\sum_{m=1}^{m} \frac{Avg.Selec^{m}}{D} \quad (7)$$

Where, $Avg.Selec^{m}$denotes the selected features for Run M, D represents the predetermined number of items in the data set.

5.1.5 Best fitness activity:

The most effective robustness implementation is characterized by the lesser robustness evaluation required between M iterations. This is calculated as follows:

$$best_{ff} = \underset{m}{Min}\, h_m^* \quad (8)$$

Where, h_m^* describes the best fitness value determined in run m.

5.1.6Average Computation Time:

The typical computation time will be equal to the computation time measured in seconds for algorithm M, and is represented as follows:

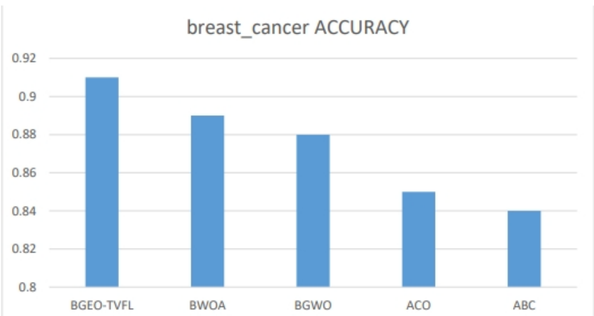
$$Average.Comptime = \frac{1}{M}\sum_{m=1}^{m} Average\,.Comptime^{m} \quad (9)$$

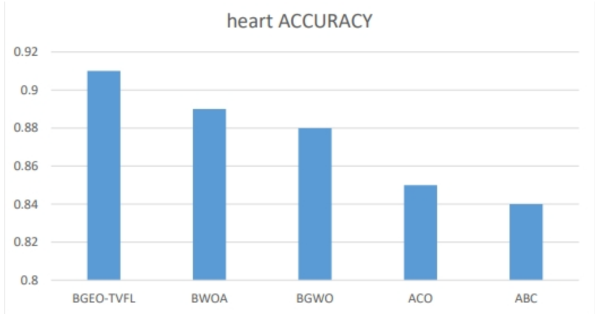
Where, $Average\,.Comptime^{m}$ represents the value of calculation time obtained in running m.

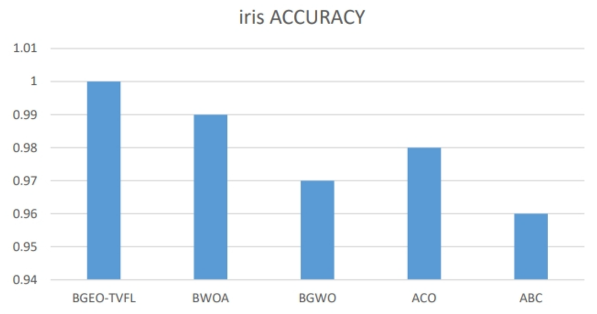
Table 3. Comparison of classification accuracy with the proposed BGEO-TVFL and existing approaches.

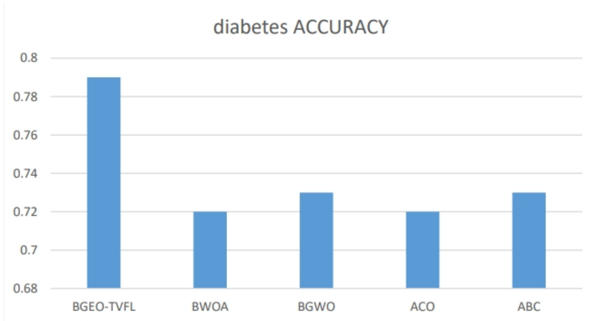
S.no	Dataset	BGEO-TVFL	BWOA	BGWO	ACO	ABC
01	breast cancer	0.91	0.89	0.88	0.85	0.84
02	heart	0.96	0.94	0.92	0.95	0.91
03	diabetes	0.79	0.72	0.73	0.72	0.73
04	Iris	1	0.99	0.97	0.98	0.96
05	thyroid_csv	0.91	0.89	0.88	0.85	0.87

The table 3 shows the performance of five machine learning models (BGEO-TVFL, BWOA, BGWO, ACO, ABC) on five different datasets. Model performance is measured in terms of accuracy, which a common metric is used to measure the performance of machine learning models. It shows that each model performs well on a particular data set. For example, BGEO-TVFL performs best on the breast_cancer dataset with an accuracy of 0.91, while BGWO performs best on the thyroid dataset with an accuracy of 0.96 this indicates that no single model is the best fit for all data sets .The table (3) allows you to compare the performance of different machine learning models on different data sets. This can help to select the best model for a particular problem. For example, if we are working with a dataset similar to breast_cancer.csv, we may want to consider using BGEO-TVFL because it has the highest accuracy. The table shows that some models have consistent performance across different data sets. For example, the accuracy of BGEO-TVFL is 0.91 for both breast_cancer.csv and heart.csv. This suggests that this model may be more robust and reliable than others. The table also shows that some models have low accuracy on some data sets. For example, the accuracy of ABC in the heart.csv dataset is only 0.8. This suggests that these models may not be well suited for these data sets, and may need to be updated or trained. The table provides a basis for further research on machine learning models and their applications. Future work may lead to the development of new and more effective models.









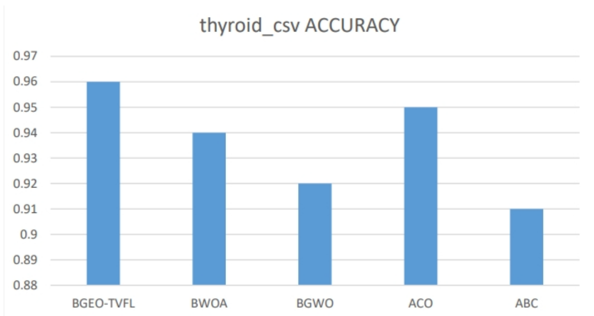


Fig. 1 .Accuracy Graph for All Datasets

The fig 1 representing the accuracy of different algorithms or methods on the breast cancer, diabetes, iris, heart and thyroid dataset. The methods shown in the above bar graphs are BGEO-TVFL, BWOA, BGWO, ACO, and ABC. BGEO-TVFL has the highest accuracy, followed by BWOA and BGWO, while ACO and ABC have lower accuracy values in comparison. The values may change regarding to the datasets and the accuracy. By comparing all the features in the bar graphs BGEO-TVFL method has only the highest accuracy.

Table 4. Comparison of selected features with the proposed BGEO-TVFL and existing methods.

S.no	Dataset	BGEO-TVFL	BWOA	BGWO	ACO	ABC
01	breast cancer	3	5	4	6	5
02	heart	8	13	12	13	11
03	diabetes	3	5	4	6	4
04	Iris	1	2	3	2	3
05	thyroid_csv.	4	7	5	6	5

The table 4 seems to compare the performance of different algorithms (BGEO-TVFL, BWOA, BGWO, ACO, and ABC) on different data sets. Each algorithm is assigned a score (represented by a number in the table) indicating its performance on each data set. This means checking how well these algorithms perform on different data sets. The scores in the table vary significantly among different datasets, indicating that the performance of each algorithm depends on the dataset. For example, the algorithm BGEO-TVFL performs well on the "breast_cancer.csv" dataset (score = 3) but poorly on the "thyroid_csv.xlsx" dataset (score = 8) this indicates that the choice of algorithm is based on uniqueness and hence dataset properties. The table allows a direct comparison of the performance of different algorithms on each data set. For example, in the data set "diabetes.csv", BWOA outperforms BGEO-TVFL and BGWO (scores = 5, 4, 4, respectively). This comparison can help determine which algorithm works best for a given problem. The table provides the basis for selecting an algorithm for a particular problem. For example, if a researcher wants to work with the dataset "heart.csv", he can choose ACO or ABC based on its slightly higher score (5 and 5, respectively) On the other hand, if a researcher having an "iris" with a .csv" dataset If it works, you are allowed to choose either BWOA or BGWO due to your high score (2 and 3).

Examination of the scores in the data sets determines whether any pattern.

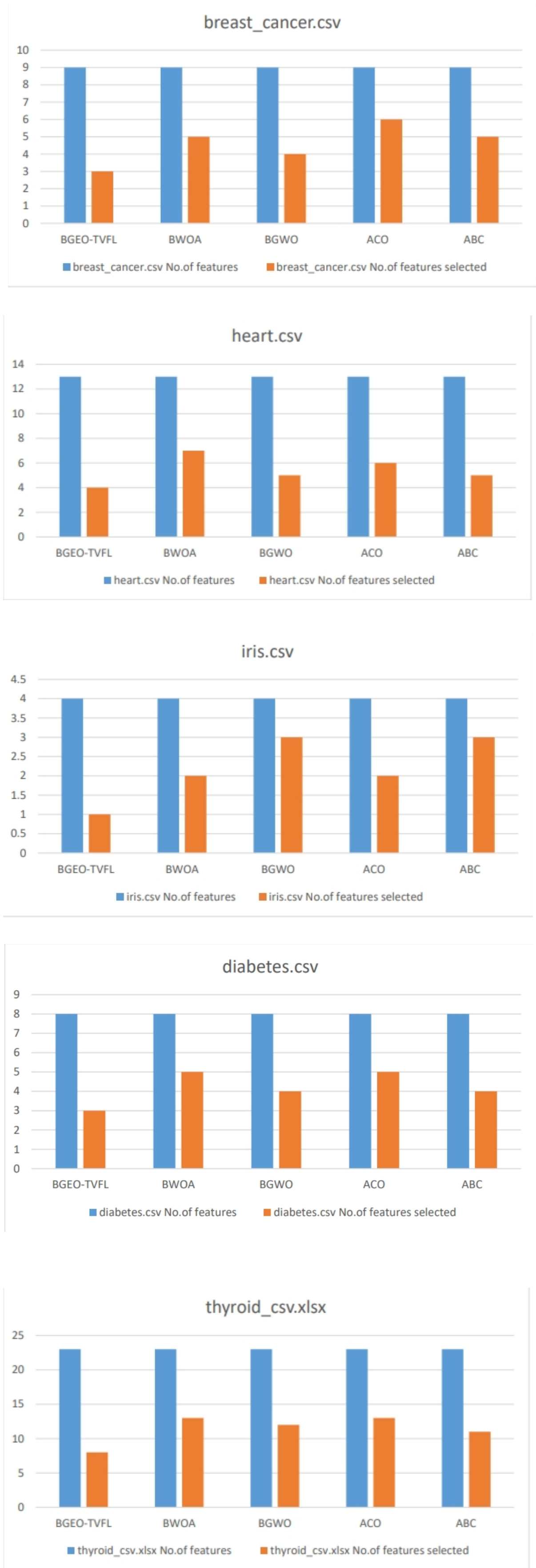


Fig. 2. Classification graphs for all datasets

The above fig 2 represents that the selected features and the number of features in the datasets breast cancer, diabetes, iris, heart and thyroid datasets in five different formats like BGEO-TVFL, BWOA, BGWO, ACO and ABC methods. The blue bars indicates that the total number of selected features and while orange bars indicates that the number of items selected by each method. All methods initially have 8 items, but the number of items selected varies, with some methods selecting fewer items than others in the above graphs. It may vary regarding to the datasets and regarding to the methods.

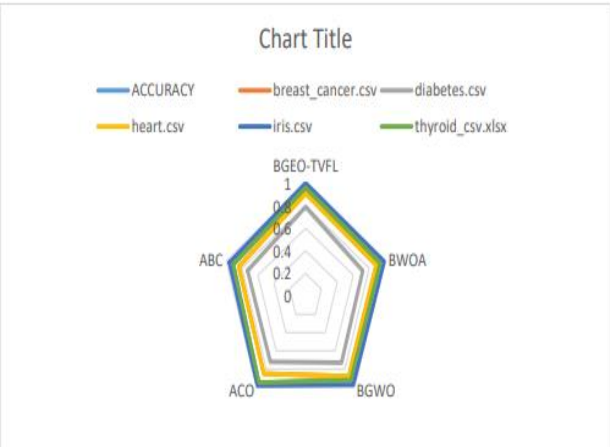


Fig. 3. Radar Graph for all algorithms with all datasets

The above figure(3) is a radar chart of the accuracy of five different algorithms (BGE0-TVFL, BWOA, BWO, ACO, ABC) on four datasets (iris.csv, breast_cancer.csv, diabetes.csv, heart.csv) .The chart shows that each algorithm How it performs on each dataset, with accuracy values ranging from 0 to 1.Each algorithm is represented by a different spectrum. Radar graphs show algorithm performance on different datasets, highlighting each algorithm's strengths and weaknesses for different types of data. The radar chart visually displays algorithm accuracy scores on different data sets, making it easy to compare and identify the most effective algorithms for each data set.

5. CONCLUSION

In conclusion, the proposed BGEO TVFL algorithm, together with the KNN classification algorithm, showed significant improvement in medical data analysis Using the unique features of both algorithms we addressed the features of its selection and classification successfully conquer the medical database. The experimental results presented in this study show that BGEO TVFL is superior in terms of classification accuracy and feature selection compared to existing methods such as BWOA, BGWO, ACO, ABC etc., which results show that the proposed method selects highly relevant features that can be improved The proposed method has been tested in several clinical cases, and the results show that it is robust and effective in clinical settings. The ability of the TVFL algorithm to adapt to changing flight lengths and flexible BGEOs allows more appropriate selection, improving classification accuracy the results also show the controllability of the proposed method handling cases with large sections works well

REFERENCES

1. Remeseiro, B., & Bolon-Canedo, V. (2019). A review of feature selection methods in medical applications. *Computers in biology and medicine*, *112*, 103375.
2. Krishna, E. R., Devarakonda, N., Al-Shamri, M. Y. H., & Revathi, D. (2022). A novel hybrid clustering analysis based on combination of k-means and pso algorithm. In *Data Intelligence and Cognitive Informatics: Proceedings of ICDICI 2021* (pp. 139-150). Singapore: Springer Nature Singapore.
3. Eluri, R. K., & Devarakonda, N. (2023). Chaotic binary pelican optimization algorithm for feature selection. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *31*(03), 497-530.
4. Mafarja, M., Aljarah, I., Heidari, A. A., Hammouri, A. I., Faris, H., Ala'M, A. Z., & Mirjalili, S. (2018). Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowledge-Based Systems*, *145*, 25-45.
5. Eluri, R. K., & Devarakonda, N. (2023). Feature selection with a binary flamingo search algorithm and a genetic algorithm. *Multimedia Tools and Applications*, *82*(17), 26679-26730.
6. Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing*, *62*, 441-453.
7. Eluri, R. K., & Devarakonda, N. (2021, October). A concise survey on solving feature selection problems with metaheuristic algorithms. In *International Conference on Advances in Electrical and Computer Technologies* (pp. 207-224). Singapore: Springer Nature Singapore.
8. Durgam, R., Devarakonda, N., Nayyar, A., & Eluri, R. (2022). Improved genetic algorithm using machine learning approaches to feature modelled for microarray gene data. In *Soft Computing for Security Applications: Proceedings of ICSCS 2021* (pp. 859-872). Springer Singapore
9. GU, S., Cheng, R., & Jin, Y. (2018). Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Computing*, *22*, 811-822.
10. Agrawal, R. K., Kaur, B., & Sharma, S. (2020). Quantum based whale optimization algorithm

for wrapper feature selection. *Applied Soft Computing*, *89*, and 106092.

11. Song, X. F., Zhang, Y., Gong, D. W., & Sun, X. Y. (2021). Feature selection using bare-bones particle swarm optimization with mutual information. *Pattern Recognition*, *112*, 107804.
12. Uzer, M. S., Yilmaz, N., & Inan, O. (2013). Feature selection method based on artificial bee colony algorithm and support vector machines for medical datasets classification. *The Scientific World Journal*, *2013*(1), 419187.
13. Too, J., & Mirjalili, S. (2021). A hyper learning binary dragonfly algorithm for feature selection: A COVID-19 case study. *Knowledge-Based Systems*, *212*, 106553.
14. Eluri, R. K., & Devarakonda, N. (2022). Binary golden eagle optimizer with time-varying flight length for feature selection. *Knowledge-Based Systems*, *247*, 108771.
15. Faris, H., Hassonah, M. A., Al-Zoubi, A. M., Mirjalili, S., & Aljarah, I. (2018). A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture. *Neural Computing and Applications*, *30*, 2355-2369.
16. Wang, X. D., Chen, R. C., Yan, F., Zeng, Z. Q., & Hong, C. Q. (2019). Fast adaptive K-means subspace clustering for high-dimensional data. *IEEE Access*, *7*, 42639-42651.
17. Krishna, E. R., & Devarakonda, N. (2023, January). Feature selection method based on GWO-PSO for coronary artery disease classification. In *2023 Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)* (pp. 1-8). IEEE.