A Feature Weighting Particle Swarm Optimization Method to Identify Biomarker Genes

Xubin Wang
Guangdong Key Lab of AI and Multi-Modal Data Processing
BNU-HKBU United International College
Institute of Artificial Intelligence and Future Networks
Beijing Normal University
Zhuhai, China
wangxb19@mails.jlu.edu.cn

Weijia Jia*, Fellow, IEEE
Institute of Artificial Intelligence and Future Networks
Beijing Normal University
Guangdong Key Lab of AI and Multi-Modal Data Processing
BNU-HKBU United International College
Zhuhai, China
jiawj@bnu.edu.cn

2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 978-1-6654-6819-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/BIBM55620.2022.9995376

***Abstract*—The discovery of biomarker genes from gene expression data is a hot topic for understanding the mechanisms underlying disease etiology. However, while the collection of high-dimensional gene expression data has been made possible by the adoption of technologies such as DNA microarray, it also poses challenges for the identification of key disease-causing genes due to its high-dimensional nature. To address this problem, we propose a feature weighting particle swarm optimization method (FWPSO) for efficiently identifying biomarker genes from high-dimensional microarray data. Specifically, there are two significant phases in FWPSO: *1) Feature Weighting Phase:* Features will be discriminated into relevant and irrelevant based on the evolutionary performance of individuals in the PSO population in each generation, and features will be assigned weights based on this. *2) Feature Selection Phase:* By focusing the search on a feature set that have been determined to be relevant based on the results of the previous phase, the PSO population will improve the efficiency of removing redundant features and discovering the most related genes. Both phases work together and operate in synergy to achieve the optimized results. The experimental results on four microarray datasets shows that FWPSO not only reduces the number of feature dimensions to a large extent, but also achieves higher classification accuracy compared to other methods, demonstrating the effectiveness of our method. Our implementation of FWPSO is available at https://github.com/wangxb96/FWPSO.**

***Index Terms*—Feature Selection; Feature Weighting; Particle Swarm Optimization; Microarray; Classification**

I. Introduction

Thanks to the development of microarray technology, the collection of large-scale gene expression data has become a reality. Especially for research, the study of microarray data is crucial for the identification of biomarkers and the analysis of key gene functions [1], thus attracting many researchers into this field [2]. In recent years, the use of artificial intelligence methods to assist microarray data research has become a valuable supplement [3] [4]. However, due to the high dimensionality of microarray data, it is easy for existing AI models to fall into the 'curse of dimensionality', resulting in high training costs and unpredictability of training results [5]. Therefore, it is imperative to consider more focused and efficient methods to reduce the effects of these problems.

* To whom correspondence should be addressed

Feature selection has proven to be one of the effective ways to alleviate the 'curse of dimensionality' [6]. Generally, there are three main categories of feature selection methods: filter, wrapper and embedded. The filter method has the advantages of generality and low complexity, and is suitable for pre-screening features on large-scale data, however, it is often inferior to the wrapper method in terms of classification performance [7]. The wrapper method usually achieves near-optimal solutions with good feature subset performance and, unfortunately, is usually very computationally intensive as it trains a new model for each chosen feature subset [6]. The embedded method treats feature selection as part of the model construction, which performs better than the filter method and has less time complexity than the wrapper method. However, embedded methods rely on the associated machine learning model and can be over-fitted [1].

Due to the high accuracy required for key gene identification, many studies in recent years have used the wrapper method for feature selection to identify biomarker genes. For example, Sayed *et. al.* [8] proposed a nested genetic algorithm to perform feature selection on high-dimensional microarray data. Jain *et. al.* [9] presented a two-stage hybrid approach for cancer classification, which combines correlation-based feature selection with improved binary particle swarm optimization. To address the shortcomings of the bacterial algorithm, Wang *et. al.* [10] proposed a feature selection algorithm based on bacterial colony optimization to improve the search capability and validated it on 15 cancer microarray datasets. Apolloni *et. al.* [11] designed two hybrid feature selection algorithms based on binary differential evolution for high-dimensional microarray data that were shown to reduce over 99% of the features. Lu *et. al.* [12] combined mutual information maximization and adaptive genetic algorithm to devise a hybrid feature selection method, which can achieve better classification accuracy than traditional methods.

Despite the positive results of the aforementioned studies in their respective fields, subjecting the feature space to a full domain calculation during the course of the evolutionary cycle also increases the computational cost to an unmanageable level at some stages (e.g. high dimensionality and large sample size). At the same time, Jain *et. al.* [9] and Lu *et. al.* [12]'s study,

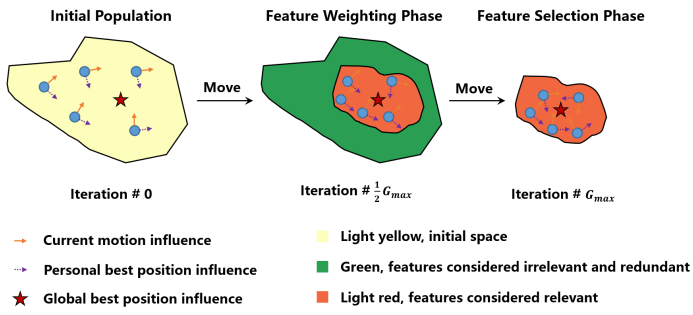

Fig. 1. Schematic diagram of FWPSO. In the feature weighting phase, individuals in the population search for the global optimum and are given weights based on the importance of the features in the search space. Based on the performance of the features in the feature weighting phase, the search space is narrowed to improve the efficiency of the search for the optimal position.

in which the filter method was used to pre-select features and then the wrapper method was used to further select features, also shows us a direction. Besides, according to Wang *et. al.* [6]'s research, assigning weights to features during evolution can also help us comprehend the significance of features.

Motivated by these observations, we propose a feature weighting particle swarm optimization method called FWPSO in this paper to deal with the challenges of high-dimensional gene expression data to further search for biomarker genes. Significantly, FWPSO uses two phases to assign weights to features (*Feature Weighting*) and narrow the feature search space (*Feature Selection*). In feature weighting phase, FWPSO distinguishes the significance of relevant features based on changes in performance caused by changes in individual features within each generation. After that, we re-initialize the PSO population in the feature selection phase based on the assignment results of each feature in the feature weighting phase, while narrowing the feature search space to achieve efficient identification of biomarker genes. The two phases work collaboratively to improve the FWPSO method's search ability to find key features.

The main contributions of the proposed method are as follows:

- We propose a feature weighting particle swarm optimization method named FWPSO, which consists of two phases, feature weighting and feature selection, to identify the importance of features and narrow the feature search space to improve the search efficiency of biomarker genes.
- FWPSO assigns weights to corresponding features through the evolutionary performance brought about by changes in the selected features of individuals in each generation of the PSO population.
- We demonstrate that the proposed FWPSO method significantly outperforms the comparison methods on four microarray datasets.

II. The Proposed FWPSO Method

In this section, we introduce the proposed FWPSO model in detail. Specifically, there are three main parts in our method: particle swarm optimization initialization, feature weighting phase and feature selection phase. The thoughts of the proposed method is outlined in Fig. 1.

A. Particle Swarm Optimization Initialization

Particle swarm optimization (PSO) [13], a swarm intelligence algorithm developed by J. Kennedy and R. C. Eberhart, has the advantages of fewer parameters and fast convergence. Typically, particles in the particle swarm optimization algorithm only have two characteristics: velocity ($\vec{V}$) and position ($\vec{X}$), where $\vec{V}$ stands for the particle's speed and $\vec{X}$ for its direction of motion. During evolution, each particle looks for the best answer on its own in the search space, saves it as the current individual best value $\vec{p}_{i,best}$, and then distributes the value to the other particles in the entire particle swarm. All of the particles in the swarm adjust their speed and position in accordance with the current individual best value $\vec{p}_{i,best}$ they discover and the current global optimal solution $\vec{p}_{best,g}$ shared by the entire particle swarm. The individual best value that is determined to be optimal is taken as the current global optimal solution $\vec{p}_{best,g}$ for the entire particle swarm.

PSO Population Initialization: In the first phase, the PSO population is initialized at random, where each domain of each individual is represented as a real number. For each individual $\vec{p}_i$, it can be coded as $\vec{p}_i = \{F_1, F_2, ..., F_{dim}\}$, where F denotes the feature search space and dim is the size of F. To achieve binarization of the feature domain, we use a threshold θ to transform the original feature domain. Specifically, the transformation process is as follows:

$$F_n = \begin{cases} 1, & F_n \geq \theta \\ 0, & F_n < \theta \end{cases} \tag{1}$$

As previously demonstrated, feature domains greater than the threshold θ are chosen and marked with the number 1, whereas feature domains that are not chosen are denoted with the number 0.

PSO Population Updating: As the swarm evolves, individual particles adjust their position and speed based on their inertia (the degree to which they trust their prior movements), cognitive state (their learned experience), and the exchange of information about the socialization of optimal particle. The update steps are as follows:

$$\vec{V_{id}} = w\vec{V_{id}} + c_1 r_1 (p_{id,\vec{best}} - \vec{X_{id}}) + c_2 r_2 (p_{d,\vec{best},g} - \vec{X_{id}}) \tag{2}$$

$$\vec{X_{id}} = \vec{X_{id}} + \vec{V_{id}} \tag{3}$$

where w is the inertia weight, c_1 is the cognitive factor, c_2 is the social factor, r_1 and r_2 are random numbers between $[0, 1]$, $p_{id,\vec{best}}$ denotes the d-th dimension at $p_{\vec{best}}$ of the i-th individual, and $p_{d,\vec{best},g}$ denotes the d-th dimension at the global optimal individual $\vec{p}_{best,g}$.

Objective Function: In our study, we focus on discovering the biomarker genes. During the training process, there are two aspects to evaluate our algorithm, namely the classification accuracy and the number of genes selected. Theoretically, fewer genes chosen should result in a higher classification accuracy, demonstrating the greater relevance of the genes chosen. Consequently, we take both into account when designing the objective function:

$$f = \alpha Error + \beta \frac{fnum}{|F|} \tag{4}$$

where $Error$ is the classification error rate, $fnum$ is the number of selected features, $|F|$ is the size of feature space, α and β are the control weights.

B. Feature Weighting Phase

Motivation: There are typically three basic categories of features in high-dimensional data: relevant, irrelevant, and redundant [7]. The evolutionary process of a particle swarm optimization algorithm is one of continuous selection of features. Therefore, our intuition is that the effectiveness of features for this task can be evaluated by statistically measuring the changes in classification performance caused by the selected features. Specifically, to simplify the feature evaluation process, we simply classify features into two groups in the feature weighting phase: relevant and irrelevant.

How to identify relevant and irrelevant features? We assess the relevance of features by the changes in performance caused by changes in features over the evolutionary process. Concretely, when individuals evolved better, we judged their emerging features as relevant and their disappearing features as irrelevant. Conversely, when individuals evolve worse, we classify emerging features as irrelevant and disappearing features as relevant.

How to assign weights to features? Overall, we create a feature evaluation matrix W that has the same number of dimensions as the feature space F to evaluate each feature F_i, and all features are given zero weight at the beginning. Here, we let the binary vector of individuals $\vec{p_i}$ in the population be $p_{i,\vec{old}}$ before evolution and become $p_{i,\vec{new}}$ after one generation of evolution. Then the evolutionary changes can be described as follows:

$$\vec{change} = p_{i,\vec{new}} - p_{i,\vec{old}} \tag{5}$$

There are three situations in vector $\vec{change}$: -1, 0 and 1, where -1 means that the feature in the corresponding domain is discarded, 0 means that the feature has not changed, and 1 means that the feature in the corresponding domain is newly selected. Specifically, there are two scenarios to assess these changed features: *1) The $p_{i,\vec{new}}$ becomes better.* In this case, the weight value of emerging features is added by one, the weight value of discarded features is subtracted by one, and no operation is performed on features that have not changed. *2) The $p_{i,\vec{new}}$ becomes worse.* In this situation, we treat discarded features as relevant and increase their weight value by one, while we treat emerging features as irrelevant and decrease their weight value by one. Features that have not changed remain unchanged in their weight values.

C. Feature Selection Phase

Feature Ranking: The feature ranking is based on historical statistics from the feature weighting phase. Weights for features are assigned during the prior training phase based on how well the newly changed features are performed. After the initial training phase, we ordered all features from most important to least important based on the weight values assigned to each feature in the weight matrix W.

Feature Screening: In general, three forms of weight values are obtained: less than 0, equal to 0 and greater than 0. At the same time, two categories of features—invalid features and redundant features—need to be weeded out. In this study, all features with weight values less than or equal to 0 are determined to be irrelevant. Therefore, we discard these features and update the feature space F:

$$F = F - F_i, \quad \forall F_i \in F, W(F_i) \leq 0 \tag{6}$$

where $W(F_i)$ is the weight values of feature F_i.

At this point, all the weights in the feature space have values greater than 0. After that, we pre-screened out redundant features by using the following criteria δ:

$$\delta = \frac{\sum_{i=1}^{|F|} W(F_i)}{|F|} \tag{7}$$

At this step, features with performance below average are regarded as redundant features. Then, we update the feature space F by discarding the features with a weight value lower than δ:

$$F = F - F_i, \quad \forall F_i \in F, W(F_i) \leq \delta \tag{8}$$

PSO Searching: After the above operation, we narrowed down the feature search space F. In this phase, the best individual from the previous phase will be retained to continue participating in the search (individual $\vec{p_1}$). We also reinitialize the PSO population based on the feature ranking results obtained in the previous phase. By sequentially choosing the amount of features in accordance with the results of the feature ranking, the individual in the PSO population is encoded until all of the individuals are encoded. Specifically, $\vec{p_2}$ selects the highest-ranked feature. Following this example, eventually $\vec{p_{NP}}$ selects the top NP-1 features. If the number of features does not encode all individuals, the remaining individuals are encoded randomly. Thereafter, the PSO population performs search operations in a reduced feature space to improve the efficiency of the search for biomarker genes.

D. Time Complexity Analysis

Since each dataset is classified via KNN, the time complexity of our proposed FWPSO method is $O(NP \times (n \times |F| + K \times n))$, where NP is the size of PSO population, n is the number of samples, $|F|$ is the size of feature space, and K is the parameter of KNN. Specifically, $|F|$ is different in Feature Weighting phase and Feature Selection phase.

III. EXPERIMENTS

A. Datasets

We use four public datasets from [1], which are available at https://github.com/xwdshiwo/MMBDE/tree/main/Datasets. In these data, they all have equal or more than 2000 dimensions, with the largest reaching 12600 dimensions. The details of these datasets are shown in TABLE I.

Dataset	# Samples	# Features	# Classes
Colon	62	2000	Tumor: 40, Normal: 22
Lymphoma	45	6937	ACL: 23, GCL: 22
Leukemia	72	7129	AML: 25, ALL: 47
Prostate	136	12600	Tumor: 59, Normal: 77

TABLE I
FOUR CANCER GENE EXPRESSION DATASETS; EACH DATASET SHOWING THE NUMBER OF SAMPLES, FEATURES, AND CLASSES.

B. Experimental Setup

Our experiments are run on a laptop computer equipped with an Intel(R) Core(TM) i5-6300HQ CPU @2.30GHz, 16GB of RAM, and a 64-bit Windows 10 operating system using Matlab 2021a. In this study, the running time is measured in seconds. Following Xie *et. al.* [1]'s work, we set the size of the population NP to 20, the maximum number of iterations G_{max} to 500, and use the average of five-fold cross-validation as the classification accuracy. The classifier in this study is KNN with K = 3. Meanwhile, as our method is a two-phase model, we set these two phases with the same number of iterations. The rest parameters of PSO [14] are listed in TABLE II.

Parameters	Values	Description
NP	20	Population size
G_{max}	500	Number of iterations
lb	0	Lower bound
ub	1	Upper bound
θ	0.9	Threshold
c1	2	Cognitive factor
c2	2	Social factor
w	0.9	Inertial weight
Vmax	$(ub - lb)/2$	Initial maximum velocity
α	0.01	Accuracy control weight
β	0.99	Feature size control weight
K	3	KNN's parameter K

TABLE II
PARAMETERS OF THE PROPOSED FWPSO METHOD

C. Comparison with Related Methods from the Literature

TABLE III shows the detailed results of all methods, where the best results are in bold. To ensure the stability of the results, we repeated each experiment five times and then averaged the results. From the results we can see that our FWPSO achieves the best classification accuracy on all datasets compared to other methods. In particular, FWPSO shows an improvement of between 1% and 6.5% over the second ranked method. The FWPSO method also has an advantage in the discovery of key genes, with the smallest feature set obtained on half of the data and very close to the best results of these methods on the other datasets.

Dataset	Methods	Accuracy	Genes
Colon	Gao [15]	0.9032	3
	Sun [16]	0.8430	5
	Lu [12]	0.8909	19
	Wang [17]	0.8570	11.1
	Lu [18]	0.8400	3
	Vanitha [19]	0.7419	3
	BDE [1]	0.9500	70
	MMBDE [1]	0.9500	4
	FWPSO (Ours)	**0.9976**	**2.4**
Leukemia	Aziz [20]	0.9868	12
	Tumuluru [21]	0.9459	NAN
	Sun [16]	0.9273	**3**
	Lu [12]	0.9762	7
	Wang [17]	0.961	8.1
	Lu [18]	0.952	9
	BDE [1]	0.9723	48
	MMBDE [1]	0.9724	5
	FWPSO (Ours)	**0.9985**	4.4
Lymphoma	Moradi [22]	0.8771	50
	Vanitha [19]	0.9090	4
	BDE [1]	0.9333	185
	MMBDE [1]	0.9556	4
	FWPSO (Ours)	**0.9978**	**3.0**
Prostate	Canedo [23]	0.9060	25
	Jinthannasatian [24]	0.8743	5
	Wu [25]	0.9044	NAN
	Wang [17]	0.904	9
	Lu [18]	0.916	**4**
	BDE [1]	0.9314	89
	MMBDE [1]	0.9124	**4**
	FWPSO (Ours)	**0.9978**	5.6

TABLE III
COMPARISON OF FWPSO AND PUBLISHED METHODS

D. Ablation Study

In this section, we analyzed the effect of each component of the FWPSO method. In particular, we discuss two cases, pure PSO and FWPSO without feature weighting (FWPSO without FW), in terms of classification accuracy, number of genes selected and run time. TABLE IV tabulates the results for all cases. As our FWPSO focuses on searching for key genes, it can be seen that in the absence of the feature weighting operation, it is far worse than pure PSO in terms of classification accuracy, although its results are better than pure PSO in terms of the number of genes obtained and the run time. At the same time, we can also see that by adding the feature weighting operation, the second phase of FWPSO searches through a more relevant and smaller set of genes, achieving not only higher classification accuracy, a smaller subset of features, but also much less run time. In particular, the runtime decreases more significantly with increasing dimensionality of the data features. In the Prostate dataset, for example, with 12,600 features, the FWPSO runtime is reduced by nearly 61% compared to pure PSO. The above results all show the effectiveness and efficiency of FWPSO in identifying key genes.

IV. CONCLUSION

In this paper, we proposed a two-phase feature weighting particle swarm optimization method FWPSO to identify biomarker genes in microarray data. FWPSO starts with a feature weighting operation in the first phase, where the importance of the features that have changed during an evolution is

Dataset	Methods	Accuracy	Genes	Run time
Colon	Pure PSO	0.9138	845.0	440.1
	FWPSO without FW	0.6992	603.6	**404.1**
	FWPSO	**0.9976**	**2.4**	428.8
Leukemia	Pure PSO	0.9893	2743.0	765.8
	FWPSO without FW	0.6907	2224.4	694.7
	FWPSO	**0.9985**	**4.4**	**493.9**
Lymphoma	Pure PSO	0.9511	1699.4	482.8
	FWPSO without FW	0.6909	1248.2	**439.7**
	FWPSO	**0.9978**	**3.0**	443.7
Prostate	Pure PSO	0.8997	5476.6	1545.4
	FWPSO without FW	0.6850	3992.4	1317.8
	FWPSO	**0.9978**	**5.6**	**607.7**

TABLE IV
ABLATION STUDY MEASURED BY ACCURACY, GENES AND RUN TIME

judged based on the change in individual performance, and the cumulative result of all the evolutions in this phase is the final weight of the corresponding feature. Based on the results of the previous phase, FWPSO screens out features judged to be irrelevant and redundant through feature screening to narrow the solution search space and improve the identification of key genes. Experiments with multiple methods on four microarray data demonstrate the effectiveness of our proposed FWPSO method in terms of classification accuracy and identification of key genes.

V. ACKNOWLEDGMENTS

Firstly, we would like to thank the anonymous reviewers for their constructive suggestions to make this paper better. Secondly, this work was supported in part by the Guangdong Key Lab of AI and Multi-modal Data Processing, United International College (UIC), Zhuhai under Grant 2020KSYS007 sponsored by Guangdong Provincial Department of Education; in part by the Chinese National Research Fund (NSFC) under Grants 62272050, 61872239; in part by Institute of Artificial Intelligence and Future Networks (BNU-Zhuhai) and Engineering Center of AI and Future Education, Guangdong Provincial Department of Science and Technology, China; Zhuhai Science-Tech Innovation Bureau under Grants ZH22017001210119PWC and 28712217900001, and in part by the Interdisciplinary Intelligence SuperComputer Center of Beijing Normal University (Zhuhai).

REFERENCES

[1] W. Xie, Y. Chi, L. Wang, K. Yu, and W. Li, "Mmbde: A two-stage hybrid feature selection method from microarray data," in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2021, pp. 2346–2351.

[2] S. Li, S. Teng, J. Xu, G. Su, Y. Zhang, J. Zhao, S. Zhang, H. Wang, W. Qin, Z. J. Lu *et al.*, "Microarray is an efficient tool for circrna profiling," *Briefings in bioinformatics*, vol. 20, no. 4, pp. 1420–1433, 2019.

[3] M. Daoud and M. Mayo, "A survey of neural network-based cancer prediction models from microarray data," *Artificial intelligence in medicine*, vol. 97, pp. 204–214, 2019.

[4] M. E. Maros, D. Capper, D. T. Jones, V. Hovestadt, A. von Deimling, S. M. Pfister, A. Benner, M. Zucknick, and M. Sill, "Machine learning workflows to estimate class probabilities for precision cancer diagnostics on dna methylation microarray data," *Nature protocols*, vol. 15, no. 2, pp. 479–512, 2020.

[5] V. Berisha, C. Krantsevich, P. R. Hahn, S. Hahn, G. Dasarathy, P. Turaga, and J. Liss, "Digital medicine and the curse of dimensionality," *NPJ digital medicine*, vol. 4, no. 1, pp. 1–8, 2021.

[6] X. Wang, Y. Wang, K.-C. Wong, and X. Li, "A self-adaptive weighted differential evolution approach for large-scale feature selection," *Knowledge-Based Systems*, vol. 235, p. 107633, 2022.

[7] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[8] S. Sayed, M. Nassef, A. Badr, and I. Farag, "A nested genetic algorithm for feature selection in high-dimensional cancer microarray datasets," *Expert Systems with Applications*, vol. 121, pp. 233–243, 2019.

[9] I. Jain, V. K. Jain, and R. Jain, "Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification," *Applied Soft Computing*, vol. 62, pp. 203–215, 2018.

[10] H. Wang, X. Jing, and B. Niu, "A discrete bacterial algorithm for feature selection in classification of microarray gene expression cancer data," *Knowledge-Based Systems*, vol. 126, pp. 8–19, 2017.

[11] J. Apolloni, G. Leguizamón, and E. Alba, "Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments," *Applied Soft Computing*, vol. 38, pp. 922–932, 2016.

[12] H. Lu, J. Chen, K. Yan, Q. Jin, Y. Xue, and Z. Gao, "A hybrid feature selection algorithm for gene expression data classification," *Neurocomputing*, vol. 256, pp. 56–62, 2017.

[13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[14] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*. IEEE, 1998, pp. 69–73.

[15] L. Gao, M. Ye, X. Lu, and D. Huang, "Hybrid method based on information gain and support vector machine for gene selection in cancer classification," *Genomics, proteomics & bioinformatics*, vol. 15, no. 6, pp. 389–395, 2017.

[16] L. Sun, X.-Y. Zhang, Y.-H. Qian, J.-C. Xu, S.-G. Zhang, and Y. Tian, "Joint neighborhood entropy-based gene selection method with fisher score for tumor classification," *Applied Intelligence*, vol. 49, no. 4, pp. 1245–1259, 2019.

[17] A. Wang, N. An, J. Yang, G. Chen, L. Li, and G. Alterovitz, "Wrapper-based gene selection with markov blanket," *Computers in biology and medicine*, vol. 81, pp. 11–23, 2017.

[18] L. Sun, X. Zhang, Y. Qian, J. Xu, and S. Zhang, "Feature selection using neighborhood entropy-based uncertainty measures for gene expression data classification," *Information Sciences*, vol. 502, pp. 18–41, 2019.

[19] C. D. A. Vanitha, D. Devaraj, and M. Venkatesulu, "Gene expression data classification using support vector machine and mutual information-based gene selection," *procedia computer science*, vol. 47, pp. 13–21, 2015.

[20] R. Aziz, C. Verma, and N. Srivastava, "A novel approach for dimension reduction of microarray," *Computational biology and chemistry*, vol. 71, pp. 161–169, 2017.

[21] P. Tumuluru and B. Ravi, "Goa-based dbn: Grasshopper optimization algorithm-based deep belief neural networks for cancer classification," *International Journal of Applied Engineering Research*, vol. 12, no. 24, pp. 14 218–14 231, 2017.

[22] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Applied Soft Computing*, vol. 43, pp. 117–130, 2016.

[23] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "An ensemble of filters and classifiers for microarray data classification," *Pattern Recognition*, vol. 45, no. 1, pp. 531–539, 2012.

[24] P. Jinthanasatian, S. Auephanwiriyakul, and N. Theera-Umpon, "Microarray data classification using neuro-fuzzy classifier with firefly algorithm," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–6.

[25] S.-J. Wu, V.-H. Pham, and T.-N. Nguyen, "Two-phase optimization for support vectors and parameter selection of support vector machines: two-class classification," *Applied Soft Computing*, vol. 59, pp. 129–142, 2017.