Exhaustive Exploitation of Nature-Inspired Computation for Cancer Screening in an Ensemble Manner

Xubin Wang, Yunhe Wang, Zhiqiang Ma, Ka-Chun Wong, and Xiangtao Li

***Abstract*—Accurate screening of cancer types is crucial for effective cancer detection and precise treatment selection. However, the association between gene expression profiles and tumors is often limited to a small number of biomarker genes. While computational methods using nature-inspired algorithms have shown promise in selecting predictive genes, existing techniques are limited by inefficient search and poor generalization across diverse datasets. This study presents a framework termed Evolutionary Optimized Diverse Ensemble Learning (EODE) to improve ensemble learning for cancer classification from gene expression data. The EODE methodology combines an intelligent grey wolf optimization algorithm for selective feature space reduction, guided random injection modeling for ensemble diversity enhancement, and subset model optimization for synergistic classifier combinations. Extensive experiments were conducted across 35 gene expression benchmark datasets encompassing varied cancer types. Results demonstrated that EODE obtained significantly improved screening accuracy over individual and conventionally aggregated models. The integrated optimization of advanced feature selection, directed specialized modeling, and cooperative classifier ensembles helps address key challenges in current nature-inspired approaches. This provides an effective framework for robust and generalized ensemble learning with gene expression biomarkers.**

***Index Terms*—Feature selection, clustering, ensemble learning, grey wolf optimizer, classification.**

I. Introduction

CANCER has become one of the leading causes of mortality worldwide, resulting in over 10 million deaths in 2020 alone [1]. The heterogeneity and complexity of various cancer types poses significant challenges for timely and accurate diagnosis, prognosis, and treatment planning [2], [3]. Precision oncology aims to overcome these difficulties by leveraging molecular biomarkers and omics data to guide personalized therapeutic decisions [4]. In particular, analysis of cancer gene expression data enables identification of discriminative genes and pathways involved in pathogenesis, which can inform diagnostic tests, prognostic indicators, and drug targets [5], [6].

However, several analytical difficulties impose barriers to identifying robust molecular biomarkers from gene expression data. Small sample sizes coupled with extremely high dimensionality and sparsity of the data make computational analysis statistically underpowered [7]. Technical noise, batch effects, tumor heterogeneity, and variability between patients also confound analyses [8], [9]. Effective and robust computational methods are therefore urgently needed to overcome these challenges and accurately detect differentially expressed genes from such complex high-dimensional datasets across diverse cancer types. This can support development of gene expression-based biomarkers for precision oncology applications.

A variety of computational approaches have been applied for cancer gene expression analysis and biomarker identification, including machine learning, deep learning, and nature-inspired optimization algorithms [10], [11], [12]. In particular, swarm intelligence and evolutionary algorithms like particle swarm optimization (PSO) [13], ant colony optimization (ACO) [14], genetic algorithms [15], and enhanced optimizer variants [16], [17], [18], [19] have shown promise. While achieving promising results, further improvements in accuracy, robustness, and generalization ability are still possible. A key limitation is that most methods rely on a single learner algorithm, which makes it difficult to determine the universally optimal learner across diverse cancer types and datasets. Different algorithms have distinct strengths and weaknesses, so their performance varies. Relying on just one also reduces robustness.

Ensemble learning methods which combine multiple diverse base learner models can help address these pitfalls [20]. Strategies like bagging [21] and boosting [22] train multiple base models on randomized or reweighted data versions, then aggregate

Manuscript received 21 November 2021; revised 7 November 2023; accepted 2 April 2024. Date of publication 5 April 2024; date of current version 9 October 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62076109, Grant 62206086, in part by the Natural Science Foundation of Hebei Province under Grant F2023202062, in part by Jilin Province Outstanding Young Scientist Program under Grant 20230508098RC, and in part by the Fundamental Research Funds for the Central Universities, JLU. *(Corresponding authors: Yunhe Wang; Xiangtao Li.)*

Xubin Wang and Xiangtao Li are with the School of Artificial Intelligence, Jilin University, Changchun 130012, China (e-mail: wangxb19@mails.jlu.edu.cn; lixt314@jlu.edu.cn).

Yunhe Wang is with the School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China (e-mail: wangyh082@hebut.edu.cn).

Zhiqiang Ma is with the School of Information Science and Technology, Northeast Normal University, Changchun 130117, China (e-mail: mazq@nenu.edu.cn).

Ka-Chun Wong is with the Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong (e-mail: kc.w@cityu.edu.hk).

Specifically, we have opened EODE source code on Github at https://github.com/wangxb96/EODE.

This article has supplementary downloadable material available at https://doi.org/10.1109/TCBB.2024.3385402, provided by the authors.

Digital Object Identifier 10.1109/TCBB.2024.3385402

1545-5963 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.

predictions to reduce variance and bias. Such ensembles have proven effective for tasks ranging from cancer subtype classification [23], [24] to drug response modeling [25]. However, naively combining all base learner models can limit diversity, leading to redundant representations and suboptimal performance [26]. Recent studies have explored intelligent optimizer-guided selection of ensemble subsets to promote specialization and synergy among members [27], [28], [29], [30], [31]. For instance, genetic algorithms have been applied to search the space of model combinations, selecting only classifiers that maximize validation accuracy through cooperative interactions [32]. While showing promise, these approaches generally utilize the full, high-dimensional feature space, which can retain irrelevant variables that confuse models and constrain diversity. Advanced feature selection is needed to derive maximally informative biomarker subsets tailored for ensemble learning [33]. Furthermore, diversity enhancement techniques like bagging and boosting are insufficient to fully overcome representation redundancies during model training [34]. Novel forms of controlled randomness injection could better promote specialization by guiding different models to focus on distinct explanatory data facets [27], [35]. Overall there remains great opportunity to advance ensemble classifier performance by integrating intelligent feature selection, guided diversity induction, and metaheuristic optimization of cooperative model combinations [26], [36]. This can further evolve the state-of-the-art in ensemble methods for precision medicine applications.

In this work, we propose a novel nature-inspired feature selection algorithm, optimized ensemble classifier, and diversity-enhancing ensemble strategy by integrating the grey wolf optimizer (GWO). Our approach, called Evolutionary Optimized Diverse Ensemble learning (EODE), synergistically combines GWO-based wrapper feature selection, diversity injection via randomized model training, and evolutionary optimization for constructing optimal ensemble classifiers. Specifically, GWO efficiently searches the high-dimensional gene expression space to identify an informative subset of discriminative features for cancer diagnosis. Multiple diverse base classifiers (e.g., SVM, KNN) are trained on these selected features while introducing randomness to increase diversity. Finally, GWO optimizes selection and integration of ensemble members to maximize performance on validation data. EODE enhances generalization ability by leveraging GWO's feature selection, controlled randomness injection, and metaheuristic ensemble optimization. We evaluate EODE on cancer gene expression datasets for tasks including subtype classification, outcome prediction, and the size of feature subset. Results demonstrate EODE significantly improves accuracy and robustness over 23 state-of-the-art methods on 35 cancer gene expression datasets. The integrated strategy advances biomarker discovery and precision oncology by evolving high-performance diverse ensemble classifiers. The main steps of the EODE approach are as follows:

1) *Base classifiers:* The diversity among the base classifiers is crucial to the effectiveness of the ensemble. The base classifiers can be any suitable classification algorithms, such as decision trees, support vector machines, or neural networks. In this study, six base classifiers including Discriminant Analysis (DISCR), Decision Tree (DT), K-Nearest Neighbor (KNN), Artificial Neural Networks (ANN), Support Vector Machine (SVM), and Naive Bayes (NB) are used.
2) *Classifier selection:* To mitigate the high computational cost associated with using ensemble methods in the feature selection training process, all base classifiers are initially trained with five-fold cross validation using the original training data. The best-performing base classifier is then selected to participate in the feature selection stage. This approach ensures that appropriate learners are involved in training for different datasets to a certain extent.
3) *Feature selection:* GWO is employed to search for an optimal subset of genes that are most relevant to cancer diagnosis. The fitness function is designed to evaluate the quality of each feature subset based on classification performance and the size of feature subset. GWO optimizes the feature subset by iteratively updating the positions of grey wolves based on their fitness values.
4) *Ensemble diversity enhancement:* To increase the diversity of ensemble, the techniques such as bagging, boosting, or random subspace method can be employed. Here, we generate multiple random subspaces through K-means clustering to increase the diversity of the ensemble. We use these data clusters to train base classifiers, resulting in a pool of models.
5) *Model pool optimization:* In the model pool, directly fusing all models can lead to lower inference efficiency, and the presence of some low-quality models may degrade the overall performance. Therefore, before final model evaluation, we optimize the model pool. We first performed pre-optimization, discarding models that performed below average on the validation set. For the remaining models, we further optimized using the GWO algorithm to select the possible optimal combination of models.
6) *Evaluation and validation:* The performance of the EODE model is evaluated using appropriate metrics such as accuracy, average performance, and the size of the feature subset. The predictions of the selected models are combined using plurality voting. The combined predictions provide the final classification result. Moreover, cross-validation and independent validation datasets are used to assess the generalization ability of the model.

II. Methods

A. Methodology Overview of EODE

In this study, we present a novel nature-inspired method called EODE for rapid identification of biomarker genes for multiple cancer types in multiple cancer gene expression datasets. A schematic overview of the algorithm is provided in Fig. 1. The original input gene expression data $\mathcal{D}_{or} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is considered, where $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,dim})$ represents a sample with dim genes, y belongs to the set $\{1, 2, \ldots, c\}$ indicating the consensus molecular subtypes, and n is the total number of samples.

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.

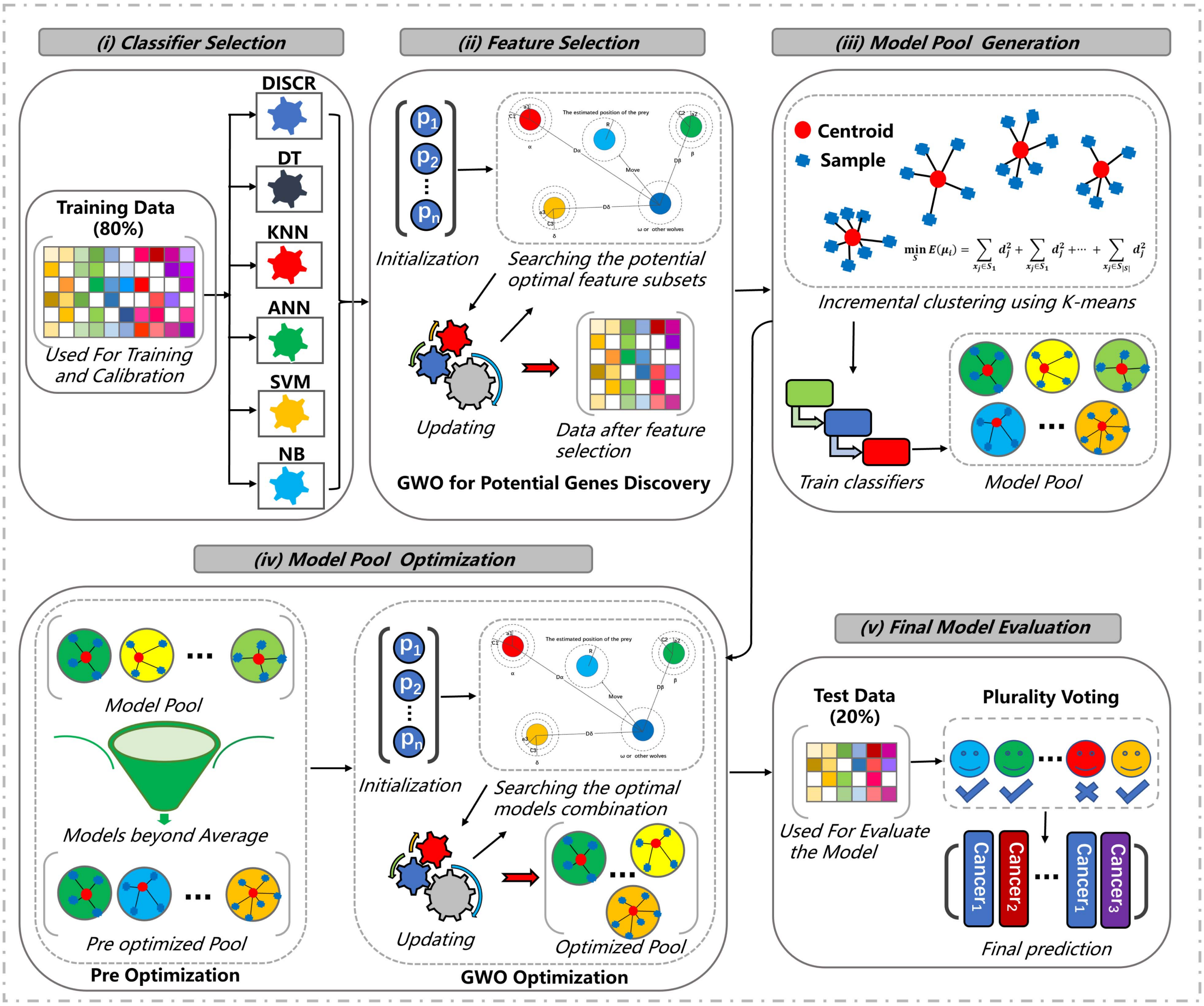

(i) Classifier Selection
Training Data (80%)
Used For Training and Calibration
DISCR
DT
KNN
ANN
SVM
NB
(ii) Feature Selection
p1
p2
pn
Initialization
Searching the potential optimal feature subsets
Updating
Data after feature selection
GWO for Potential Genes Discovery
(iii) Model Pool Generation
Centroid
Sample
$\min_S E(\mu_i) = \sum_{x_j \in S_1} d_j^2 + \sum_{x_j \in S_1} d_j^2 + \cdots + \sum_{x_j \in S_{|S|}} d_j^2$
Incremental clustering using K-means
Train classifiers
Model Pool
(iv) Model Pool Optimization
Model Pool
Models beyond Average
Pre optimized Pool
Pre Optimization
Initialization
Searching the optimal models combination
Updating
Optimized Pool
GWO Optimization
(v) Final Model Evaluation
Test Data (20%)
Used For Evaluate the Model
Plurality Voting
Cancer1
Cancer2
Cancer1
Cancer3
Final prediction

Fig. 1. Overview of the proposed EODE algorithm: In the GWO feature selection phase, the original cancer gene expression training data is utilized to train all base classifiers, and the classifier with the highest performance is selected as the evaluation classifier. The processed data is then optimized to construct an ensemble model. Specifically, the training data is incrementally clustered using the K-means method to form subspace clusters. These clusters are used to train individual base classifiers, which are then added to the model pool. Any classifiers in the pool with below-average performance are filtered out. Next, the GWO is applied to optimize the classifier pool and determine the best possible ensemble combination. Finally, the optimized ensemble model is evaluated on the independent test dataset using a plurality voting strategy to generate the final cancer type predictions.

In the feature selection step, we employ the GWO to extract relevant biomarker genes after training our model on the training gene expression matrix $\mathcal{D}_{tr}$. Each base classifier from the pool $\mathcal{B}$ (including Discriminant Analysis (DISCR), Decision Tree (DT), K-Nearest Neighbor (KNN), Artificial Neural Networks (ANN), Support Vector Machine (SVM), and Naive Bayes (NB)) is initially trained using the input data. The best-performing classifier is then chosen as the evaluation classifier for feature selection.

The processed data is subsequently utilized to train and optimize a diverse ensemble model. Specifically, the data undergoes five-fold cross-validation to construct the final model Ψ. Initially, the data is partitioned into progressive subspaces using the K-means method to form clusters. These clusters are then utilized to train base classifiers, which are subsequently incorporated into the model pool. Models in the pool with below-average performance are filtered out. After that, the GWO approach is applied to optimize the model pool and identify the best possible combination. Finally, the model Ψ is evaluated on the test data using a plurality voting strategy. The overall framework of EODE is summarized in Algorithm 1.

B. Nature-Inspired Feature Selection

Considering a training cancer gene expression data $\mathcal{D}_{tr} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,dim})$ represents the feature vector and dim denotes the number of features, y belongs to the set $\{1, 2, .., c\}$ representing the class, and n is the number of samples. It is important to note that the high-dimensional nature of the gene expression data may include many irrelevant genes, which can negatively impact identification accuracy while increasing computational time [7].

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Pseudo Code of EODE Algorithm.

```
Require: Training Data: $\mathcal{D}_{tr} = \{(x_{tr,1}, y_{tr,1}), \ldots, (x_{tr,n}, y_{tr,n})\}$, $x \in R^d$, $y \in \{1, 2, \ldots, c\}$, Test Data: $\mathcal{D}_{te} = \{(x_{te,1}, y_{te,1}), \ldots, (x_{te,n}, y_{te,n})\}$, a set of base classifiers $\mathcal{B}$, upper bounds of clustering $K$, population of GWO $\vec{X}$, the feature selection function $f_1$, the classifiers optimization function $f_2$
Use training data $\mathcal{D}_{tr}$ to train each base classifier in $\mathcal{B}$
The classifier $b$ with the best performance is selected for feature selection
Initialize a population of $|\vec{X}|$ individuals
while $t <$ max iterations $T$ do
    $\vec{X} \leftarrow$ Use Algorithm 2 to do $f_1(\vec{X})$
    $\vec{X}_i \leftarrow$ best individual
    $t$++;
end while
$bf \leftarrow$ best features selected by wolf $\vec{X}_i$
$fnum \leftarrow$ the number of features selected by wolf $\vec{X}_i$
$\mathcal{D}_{tr} \leftarrow \mathcal{D}_{tr}(bf)$
$\mathcal{D}_{te} \leftarrow \mathcal{D}_{te}(bf)$
$\mathcal{D}_{tr} = \mathcal{D}_{tr,1} \cup \ldots \cup \mathcal{D}_{tr,5}, \mathcal{D}_{tr,i} \cap \mathcal{D}_{tr,j} = \emptyset (i \neq j)$
for each $\mathcal{D}_{tr,i}$ in $\mathcal{D}_{tr}$ do
    $\mathcal{D}_{tr}^{-i} = \mathcal{D}_{tr} - \mathcal{D}_{tr,i}$
    for $k = 1 \to K$ do
        $C^S \leftarrow$ partition $\mathcal{D}_{tr}^{-i}$ into $k$ clusters
        $S = S + 1$
    end for
    for each $mp_i$ in $\mathcal{MP}$ do
        $Acc(i) \leftarrow$ calculate each $mp_i$'s validation accuracy on $D_{tr,i}$
    end for
    $\mathcal{MP} \leftarrow (mp_i$ if $Acc(mp_i) >$ mean$(Acc))$
    Initialize a population of $|\vec{X}|$ individuals
    while $t <$ max iterations $T$ do
        $\vec{X} \leftarrow$ Use Algorithm 2 to do $f_2(\vec{X})$
        $\vec{X}_i \leftarrow$ best individual
        $t$++;
    end while
    $\psi \leftarrow$ best models in $\mathcal{MP}$ selected by wolf $\vec{X}_i$
    Optimized classifier $\Psi \leftarrow \Psi + \psi$
end for
$testAcc \leftarrow$ classify samples of $\mathcal{D}_{te}$ by $\Psi$
Output: The optimized ensemble classifier $\Psi$, the number of selected features $fnum$ and the test accuracy $testAcc$
```

Therefore, performing feature selection is crucial to preprocess the data effectively.

The Grey Wolf Optimizer (GWO), initially proposed by Mirjalili [37], is a swarm intelligence algorithm inspired by the social hierarchy and hunting behavior of grey wolves in nature. GWO offers advantages such as good convergence, minimal parameter tuning, and ease of implementation [38]. The core concept of GWO revolves around three primary predation behaviors: encircling prey, hunting, and attacking prey, which are performed based on the social hierarchy among the wolves. The social hierarchy in GWO consists of four levels: α, β, δ, and ω, with α being the dominant wolf, followed by β and δ, while the remaining wolves are labeled as ω. Wolves at higher ranks exert dominance over those at lower ranks, and α, β, and δ play key roles in the algorithm, with α being the wolf king and β and δ serving as potential successors. The α wolf represents the fittest solution and guides the pack towards promising search areas. The second and third best fit solutions are modeled as β and δ wolves, respectively. The ω wolves represent the remaining weaker candidate solutions that follow the guidance of the α, β and δ wolves. During optimization, the candidate solutions iteratively update their positions towards the best three solutions until convergence upon the global optimal value. Specifically, a schematic representation of GWO is depicted in Fig. 2.

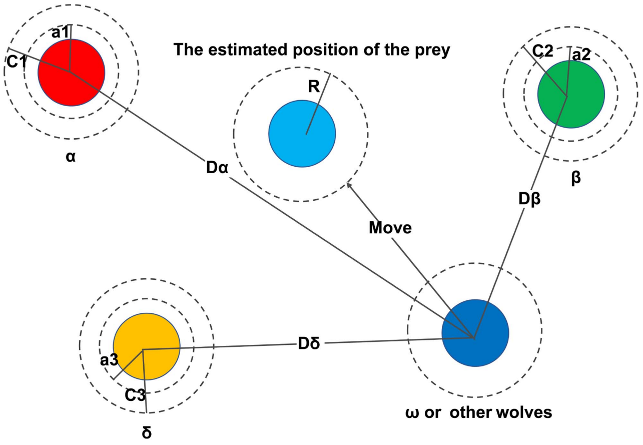


Fig. 2. GWO algorithm is illustrated in a schematic representation, highlighting the process of updating the positions of the wolves. Initially, the positions of the wolves are randomly initialized within the solution space. The fitness of each wolf is evaluated based on a fitness function. In each iteration, the positions of the wolves are updated using mathematical formulas that consider the social hierarchy, with the α wolf having the greatest influence. The update process involves attracting other wolves towards the positions of the α, β, and δ wolves. This iterative position updating continues until a termination condition is met. Ultimately, the position of the α wolf represents the best solution found by the GWO algorithm.

Building upon these foundations, we propose a nature-inspired feature selection method based on GWO, which comprises six essential components: classifier selection, population initialization, encircling prey phase, hunting phase, attacking phase, and feature selection objective function.

1) Classifier Selection: To evaluate the feature selection results, we consider six base classifiers in a classifier pool $\mathcal{B}$: Discriminant Analysis (DISCR), Decision Tree (DT), K-Nearest Neighbor (KNN), Artificial Neural Networks (ANNs), Support Vector Machine (SVM), and Naive Bayes (NB). However, incorporating all these classifiers into the ensemble method during the feature selection phase would be computationally expensive. Therefore, we adopt a pre-training approach to select the best-performing classifier from the pool $\mathcal{B}$. The cancer gene expression data $\mathcal{D}$ is subjected to five-fold cross-validation on each base classifier, and the classifier with the highest performance is chosen as the evaluation classifier for the feature selection phase.

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.

This approach allows us to efficiently select the most suitable classifier for the subsequent feature selection process.

2) Population Initialization: In the beginning, the population $\vec{X}$ is randomly created and represented as real numbers. Each individual, denoted as $\vec{X_i}$, is a set of genes: $\vec{X_i} = \{g_1, g_2, \ldots, g_{dim}\}$, where g_{dim} represents the dimth gene and dim is the total number of genes.

To convert these real numbers into a binary form, we use a threshold value θ. If a feature value (g_n) is greater than or equal to θ, it is set to 1, indicating that the corresponding feature is selected. On the other hand, if g_n is less than θ, it is set to 0, indicating that the feature is not selected. The process can be as follows:

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

After that, the position of each individual is represented by a binary (0/1) string.

3) Encircling Prey Phase: The “encircling prey” behavior is a strategy employed by the grey wolf pack to search for feature subsets. This behavior is mathematically modeled to simulate how the grey wolf gradually approaches its prey and surrounds it. The distance ($\vec{D}$) between the grey wolf and the prey is determined by the equation

$$\vec{D} = |2 \cdot r_2 \cdot \vec{X_p}(t) - \vec{X_i}(t)|, \tag{2}$$

where $\vec{D}$ represents the distance between them. During the search process, the current iteration is denoted by t, and $\vec{X_p}(t)$ and $\vec{X_i}(t)$ represent the position vectors of the prey and the grey wolf, respectively.

To update the position of the grey wolf, we utilize the formula $X_i(\vec{t+1}) = \vec{X_p}(t) - (2\vec{a} \cdot r_1 - \vec{a}) \cdot \vec{D}$. Here, $\vec{a}$ is the convergence factor that decreases linearly from 2 to 0 as the iterations progress. The convergence factor is calculated as $\vec{a} = 2 - 2t/max_t$, where t represents the current iteration, and max_t is the maximum number of iterations defined for the search process. Additionally, r_1 and r_2 are random numbers between 0 and 1.

By applying this position update formula, the grey wolf adjusts its position towards the prey. The term $(2\vec{a} \cdot r_1 - \vec{a})$ determines the magnitude and direction of the movement, while the distance $\vec{D}$ guides the grey wolf's movement in narrowing the gap with the prey. The process continues iteratively until the desired maximum number of iterations is reached (max_t). Ultimately, the grey wolf is expected to encircle the prey, indicating the discovery of a promising feature subset.

4) Hunting Phase: Grey wolves possess the ability to identify the general location of their prey and work together to surround it. However, in many unknown situations, they may not have precise knowledge of the exact location of the target. In our study, we simulate the behavior of grey wolves by introducing three key individuals: α, β, and δ. These individuals help guide the entire wolf pack in surrounding the prey and searching for the optimal solution.

Algorithm 2: Pseudo Code of Grey Wolf Optimizer (GWO).

```
Initialize a population $\vec{X}$ of wolves randomly within the solution space
Evaluate the fitness of each wolf $\vec{X_i}$ using a fitness function $f$
Set the initial values for $\alpha$, $\beta$, and $\delta$ as the wolves with the highest, second highest, and third highest fitness, respectively
**while** t $<$ max iterations $T$ **do**
  **for** each wolf $\vec{X_i}$ in the population **do**
    Update the position of the wolf based on the positions of $\alpha$, $\beta$, and $\delta$ using the following formulas:
    $\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}_i|$ // Distance from $\alpha$
    $\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}_i|$ // Distance from $\beta$
    $\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}_i|$ // Distance from $\delta$
    $\vec{X}' = \vec{X}_\alpha - A_1 \cdot \vec{D}_\alpha$ // Encircling $\alpha$
    $\vec{Y}' = \vec{X}_\beta - A_2 \cdot \vec{D}_\beta$ // Encircling $\beta$
    $\vec{Z}' = \vec{X}_\delta - A_3 \cdot \vec{D}_\delta$ // Encircling $\delta$
    Update the position of the wolf using:
    $X_i(\vec{t+1}) = (\vec{X}' + \vec{Y}' + \vec{Z}')/3$
    Apply boundary constraints to ensure the new position is within the solution space
  **end for**
  Update the fitness of each wolf $\vec{X_i}$ using a fitness function $f$
  Update $\alpha$, $\beta$, and $\delta$ based on the updated fitness values
  $t$++;
**end while**
*Output:* The position of the $\alpha$ wolf represents the best solution found by the GWO algorithm
```

To track the position of the prey, each individual grey wolf calculates its distance to the prey using the following equations:

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}_i|, \tag{3}$$

$$\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}_i|, \tag{4}$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}_i|. \tag{5}$$

Here, $\vec{D_\alpha}$, $\vec{D_\beta}$, and $\vec{D_\delta}$ represent the distances between the grey wolves α, β, δ and the prey, respectively. $\vec{X_\alpha}$, $\vec{X_\beta}$, and $\vec{X_\delta}$ denote the positions of α, β, and δ, while $\vec{X_i}$ represents the current position of the grey wolf. Additionally, $\vec{C_1}$, $\vec{C_2}$, and $\vec{C_3}$ are random vectors used to calculate these distances.

Each grey wolf updates its position based on these distance calculations

$$\vec{X}' = \vec{X}_\alpha - A_1 \cdot \vec{D}_\alpha, \tag{6}$$

$$\vec{Y}' = \vec{X}_\beta - A_2 \cdot \vec{D}_\beta, \tag{7}$$

$$\vec{Z}' = \vec{X}_\delta - A_3 \cdot \vec{D}_\delta. \tag{8}$$

Here, $\vec{X}'$, $\vec{Y}'$, and $\vec{Z}'$ represent the new positions of the grey wolves moving towards α, β, and δ, respectively. The constants

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.

A_1, A_2, and A_3 control the magnitude of the movement towards the prey.

Finally, the position of the grey wolf at the next time step $X_i(\vec{t+1})$ is determined as the average of the positions $\vec{X}'$, $\vec{Y}'$, and $\vec{Z}'$

$$X_i(\vec{t+1}) = \frac{\vec{X}' + \vec{Y}' + \vec{Z}'}{3}. \tag{9}$$

In this way, the entire wolf pack moves together towards the positions of α, β, and δ, and the new position of each individual is updated accordingly.

5) Attacking Phase: The final stage of the hunting process is the attack, during which the grey wolves aim to capture their prey and obtain the optimal solution. This phase involves adjusting certain parameters to strike a balance between global exploration and local exploitation.

To achieve this balance, two key parameters are considered: a and A. The value of a is progressively decreased from 2 to 0 in a linear manner. Simultaneously, the range of fluctuations in A is reduced. The parameter A takes on values within the range $[-a, a]$. The behavior of the grey wolves is influenced by the magnitude of A. When the absolute value of A is greater than 1, the grey wolves tend to spread out across different areas, enabling a global search for prey. Conversely, when the absolute value of A is less than 1, the grey wolves exhibit a more focused, local search.

In addition to these parameters, the influence of the grey wolves' positions on the prey is governed by a random weight, denoted as C. This weight, which ranges between 0 and 2, determines the random influence of the grey wolf's location on the prey. A value of C greater than 1 indicates a higher weight, emphasizing the significance of the grey wolf's position in guiding the search. Conversely, a value of C less than 1 assigns a lower weight, reducing the impact of the grey wolf's location. This random weight, C, helps prevent the algorithm from converging too early and becoming trapped in a local optimum.

By dynamically adjusting the values of a, A, and C during the attacking phase, the grey wolves strike a balance between exploration and exploitation, allowing them to efficiently search for and capture the optimal solution while avoiding premature convergence and local optima.

6) Feature Selection Objective Function: During each iteration of the GWO algorithm, the classification label for each candidate solution $\vec{X_i}$ is predicted using the evaluation classifier selected from the classifier selection phase. Specifically, the evaluation classifier is initially trained on the original training gene expression dataset $\mathcal{D}_{\vec{tr}}$ with all features using five-fold cross-validation. For each $\vec{X_i}$ containing a subset of selected features, the evaluation classifier generates predicted labels y_i' by classifying the corresponding data points from $\mathcal{D}_{tr}$ using only the selected features in $\vec{X_i}$. The performance of y_i' on $\mathcal{D}_{tr}$ determines the fitness value assigned to solution $\vec{X_i}$. This allows the GWO algorithm to determine the α, β, and δ solutions representing the current best feature subsets for classification.

In the feature selection stage, the primary objective is to identify and select relevant features while filtering out redundant ones for subsequent identification purposes in cancer gene expression data. Traditional studies often focus solely on classification accuracy, disregarding the resource costs associated with redundant features. In our study, we address this limitation by considering both classification accuracy and the size of the feature subsets as part of our feature selection objective function [39].

The objective function, denoted as f_1, is defined as follows:

$$f_1 = \alpha * \text{error} + \beta * \frac{f_{\text{num}}}{\text{dim}}. \tag{10}$$

Here, f_{num} represents the number of selected features during the evolutionary process, and dim represents the total number of features in the dataset. To strike a balance between the two objectives, we introduce weight coefficients to control their relative importance. In our study, we assign a weight of 0.9 to α to emphasize the significance of classification accuracy, while β is set to 0.1 to underscore the importance of the feature subset size. These weight coefficients were determined based on the findings in the reference [40], where classification accuracy was identified as the primary objective.

The classification error (error) is a key component of the objective function. It is calculated as the difference between 1 and the accuracy (acc), which is defined as

$$\text{error} = 1 - \text{acc}, \tag{11}$$

$$\text{acc} = \frac{\sum_{s=1}^{n} I(y_s', y_s)}{n}. \tag{12}$$

In the above equations, n represents the total number of instances, y_s' represents the predicted class label for instance s, and y_s represents the true class label for instance s. The function $I(y_s', y_s)$ evaluates to 1 if the predicted and true class labels match, and 0 otherwise.

C. Nature-Inspired Diverse Ensemble Learning

In this section, we propose a novel nature-inspired diverse ensemble learning method to improve the performance of cancer identification using selected features obtained through nature-inspired feature selection. Our method comprises diverse subspace generation, model pool generation, and model pool optimization.

1) Diverse Subspace Generation: Given the gene expression data after feature selection, denoted as $\mathcal{D}_{tr}' = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, where $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,dim})$ represents the feature vector with dim denoting the number of features, $y \in \{1, 2, \ldots, c\}$ represents the classification label, and m represents the number of input samples, we employ the K-means method [41] to cluster the input cancer gene expression data into multiple clusters. The clustering process is performed iteratively from 1 to t, generating K clusters in each iteration. Here, t denotes the total number of iterations. The clusters are

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.

obtained by minimizing the following function:

$$\underset{S}{argmin} \sum_{i=1}^{K} \sum_{x \in S_i} ||x - \mu_i||^2, \tag{13}$$

where x represents the feature vector and μ_i is the centroid of cluster S_i. This clustering process generates a set of diverse subspaces composed of all the obtained clusters.

2) Model Pool Generation: Each cluster in the diverse subspace is used to train six classifiers (DISCR, DT, KNN, ANN, SVM, and NB) to create a model pool. The base classifiers used in this step are independent. The resulting models are then added to the base model pool $\mathcal{MP}$. The base model pool $\mathcal{MP}$ consists of $l * |\mathcal{B}|$ models, where l represents the number of clusters and $|\mathcal{B}|$ represents the number of base classifiers. Finally, we employ nature-inspired optimization techniques to refine the base models in the ensemble. Here, any combination of classifiers can be utilized.

3) Model Pool Optimization: After obtaining the diverse base model pool $\mathcal{MP}$, we propose a pre-optimization step to refine $\mathcal{MP}$ by removing models with below-average performance. Subsequently, we incorporate a nature-inspired optimization method, namely GWO, to further optimize the pre-optimized base model pool $\mathcal{MP}$.

Population Initialization: The population is randomly initialized, and each individual is represented as follows:

$$\vec{X_i} = \{mp_1, mp_2, \ldots, mp_r\}. \tag{14}$$

Here, mp_r represents a classifier in the model pool $\mathcal{MP}$, and r is the total number of models in $\mathcal{MP}$. Similar to nature-inspired feature selection, the selection or non-selection of models is indicated by binary values. "1" indicates that a model is selected, while "0" indicates that the model is not selected. To convert the continuous search space of GWO into a binary search space, we introduce a threshold θ. The conversion from a continuous position to discrete binary values is defined as follows:

$$mp_r = \begin{cases} 1, & mp_r \geq \theta \\ 0, & mp_r < \theta \end{cases}. \tag{15}$$

Nature-inspired Optimization Process: In this phase, our aim is to discover optimal model subsets by optimizing the base model pool $\mathcal{MP}$. The population is used to explore optimal model subsets in the encircling phase, identify potential optimal solutions in the hunting phase, and ultimately obtain the optimal solution in the attacking phase.

Ensemble Optimizing Objective Function: Our objective is to achieve the highest identification performance with the smallest ensemble size. After clustering the data following feature selection and training the base classifiers to create a model pool, we aim to optimize the model pool to obtain the optimal ensemble model with the smallest size. The optimized model ensemble is then evaluated using the test data. The objective function in the model pool optimization stage, denoted as f_2, is defined as follows:

$$f_2 = \alpha * \text{error} + \beta * \frac{|\psi|}{r}. \tag{16}$$

Here, error represents the identification error rate described in (11), $|\psi|$ is the total number of selected models, and r is the number of models in $\mathcal{MP}$. The settings of α and β are identical to those in Section II-B6, with α accounting for 90% of the importance and β for 10%.

However, unlike in Section II-B6, where the predicted label y'_s is predicted by a single classifier, we consider the ensemble of multiple models. We employ a plurality voting method to combine the predictions of multiple models, which has been proven to be a simple and effective ensemble fusion technique in many studies [42], [43].

4) Ensemble Classifier Prediction: During the training process, we obtain multiple models ψ to represent the model Ψ. The model Ψ is used to generate an ensemble, and all models in Ψ are utilized to predict the test set. The predicted class labels y'_s from all the models in the model Ψ are fused using the plurality voting method. The identification accuracy can be calculated using (12).

D. Time Complexity Analysis

Here, we analyze the time complexity of our proposed EODE algorithm. The detailed analysis is outlined as follows:

- *Feature Selection:* The time complexity of the feature selection process depends on the algorithm used. Since we used GWO for feature selection, the time complexity is typically $O(T \times P \times F \times C)$, where T is the number of generations, P is the population size, F is the number of features, and C is the complexity of the fitness evaluation function. Generally, the feature selection process has a polynomial time complexity.
- *Diverse Subspace Generation:* The time complexity of the diverse subspace generation mainly depends on the clustering algorithm used. Here, we applied the K-means algorithm, the time complexity is usually $O(K \times N \times I \times d)$, where K is the number of clusters, N is the number of data points, I is the number of iterations, and d is the dimensionality of the data. The diverse subspace generation process has a polynomial time complexity.
- *Model Pool Generation:* The model pool generation involves training multiple base classifiers on each cluster. The time complexity depends on the complexity of the base classifiers and the number of clusters. Assuming the time complexity of training a base classifier on a single cluster is $O(N \times F \times C)$, where N is the number of data points, F is the number of selected features, and C is the complexity of the training algorithm, the overall time complexity of model pool generation is $O(L \times N \times F \times C)$, where L is the number of clusters. This process also has a polynomial time complexity.
- *Model Pool Optimization:* The time complexity of the model pool optimization stage depends on the optimization algorithm used. We employed a nature-inspired optimization algorithm named GWO, the time complexity is typically $O(T \times P \times C)$, where T is the number of generations, P is the population size, and C is the complexity of

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.

TABLE I
Thirty-five Different Gene Expression Datasets; Each Dataset Showing the Tissue Type, Number of Samples, Features, and Classes

Dataset	Tissue	Samples	Features	Classes	Dataset	Tissue	Samples	Features	Classes
Alizadeh-2000-v1	Blood	42	1095	2	Alizadeh-2000-v2	Blood	62	2093	3
Alizadeh-2000-v3	Blood	62	2093	4	Armstrong-2002-v1	Blood	72	1081	2
Armstrong-2002-v2	Blood	72	2194	3	Bhattacharjee-2001	Lung	203	1543	5
Bittner-2000	Skin	38	2201	2	Bredel-2005	Brain	50	1739	3
Chen-2002	Liver	179	85	2	Chowdary-2006	Breast, Colon	104	182	2
Dyrskjot-2003	Bladder	40	1203	3	Garber-2001	Lung	66	4553	4
Golub-1999-v1	Bone Marrow	72	1877	2	Golub-1999-v2	Bone Marrow	72	1877	3
Gordon-2002	Lung	181	1626	2	Khan-2001	Multi-tissue	83	1069	4
Laiho-2007	Colon	37	2202	2	Lapointe-2004-v1	Prostate	69	1625	3
Lapointe-2004-v2	Prostate	110	2496	4	Liang-2005	Brain	37	1411	3
Nutt-2003-v1	Brain	50	1377	4	Nutt-2003-v2	Brain	28	1070	2
Nutt-2003-v3	Brain	22	1152	2	Pomeroy-2002-v1	Brain	34	857	2
Pomeroy-2002-v2	Brain	42	1379	5	Ramaswamy-2001	Multi-tissue	190	1363	14
Risinger-2003	Endometrium	42	1771	4	Shipp-2002-v1	Blood	77	798	2
Singh-2002	Prostate	102	339	2	Su-2001	Multi-tissue	174	1571	10
Tomlins-2006-v1	Prostate	104	2315	5	Tomlins-2006-v2	Prostate	92	1288	4
West-2001	Breast	49	1198	2	Yeoh-2002-v1	Bone Marrow	248	2526	2
Yeoh-2002-v2	Bone Marrow	248	2526	6	-	-	-	-	-

the fitness evaluation function. Similar to the feature selection process, the model pool optimization stage generally has a polynomial time complexity.

- *Ensemble Classifier Prediction:* The time complexity of the ensemble classifier prediction is dependent on the number of models in the ensemble and the complexity of combining their predictions. Assuming we have M models in the ensemble and the complexity of combining predictions is $O(M)$, the overall time complexity is $O(M)$. This process has a linear time complexity.

In summary, the overall time complexity is: Overall Time Complexity = Feature Selection + Diverse Subspace Generation + Model Pool Generation + Model Pool Optimization + Ensemble Classifier Prediction = O(T × P × F × C) + O(K × N × I × d) + O(L × N × F × C) + O(T × P × C) + O(M)

Since all these time complexities are polynomial, we can express the overall time complexity as the highest-order term in the sum. Therefore, the overall time complexity of the EODE algorithm is

Overall Time Complexity = max{T × P × F × C, K × N × I × d, L × N × F × C, T × P × C, M}

III. Implementation

A. Datasets

The cancer gene expression datasets were collected from [44], and can be downloaded from the website https://schlieplab.org/Static/Supplements/CompCancer/datasets.htm. The 35 datasets contain multiple types of cancers with high-dimensional features, exceeding 1,000 dimensions, while having relatively small sample sizes (as shown in Table I). This poses the “Curse of Dimensionality” challenge, necessitating the development of a computational model with high robustness and good generalization capabilities to address the different cancers.

To enable rigorous evaluation, the collected raw datasets have been randomly split into disjoint training and testing sets in a 80:20 ratio prior to conducting experiments. The training sets, comprising 80% of the data, have been used for model training and hyperparameter tuning. The testing sets, comprising the held-out 20% of the data, have only been used for final evaluation of the fully trained model’s performance. This ensures an unbiased estimate of generalization capability. The precise training/testing splits has been done randomly while preserving class balance in each set. Specifically, the training and testing datasets can be downloaded from the following links: https://github.com/wangxb96/EODE/tree/master/TrainData and https://github.com/wangxb96/EODE/tree/master/TestData. For model selection and hyperparameter tuning, k-fold cross-validation (k=5) was utilized during model selection and hyperparameter optimization on the training data only. By segregating the training and testing data, we prevent information leakage and overfitting to the test set. This rigorous methodology allows us to evaluate true generalization error and robustness across multiple cancer types.

B. Baselines

To evaluate the effectiveness of our proposed method, we compared it against several existing classifiers and ensemble algorithms widely used in the literature. First, we compared our model with six base classifiers: DISCR (Discriminant Analysis) [45], DT (Decision Tree) [46], KNN (K-Nearest Neighbor) [47], ANN (Artificial Neural Networks) [48], SVM (Support Vector Machine) [49], and NB (Naive Bayes) [50]. These classifiers serve as the baseline for performance comparison.

Next, we compared our approach with seven evolutionary algorithms: ACO [51], CS [52], DE [53], GA [54], GWO [37], PSO [55], and ABC [56]. These algorithms are widely used for optimization problems. Furthermore, we evaluated our approach against four novel ensemble methods: PSOEL [27], EAEL [57], FESM [58], and GA-Bagging-SVM [59]. These methods were selected to demonstrate the effectiveness of our proposed approach in comparison to recent advancements in ensemble learning.

In addition, we compared our ensemble algorithm with six state-of-the-art ensemble classifiers: Random Forests (RF) [60], ADABOOST [22], RUSBOOST [61], SUBSPACE [62], TOTALBOOST [63], and LPBOOST [64]. Random Forests is

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.

TABLE II
PARAMETERS OF DIFFERENT MACHINE LEARNING METHODS

Methods	Parameters
DISCR	discrimtype = diaglinear
KNN	K = 3
SVM	'KernelFunction' = 'rbf', 'IterationLimit' = 50000, 'Standardize'= true
NB	distribution = kernel

TABLE III
PARAMETERS OF DIFFERENT EVOLUTIONARY ALGORITHMS

Methods	Parameters
ACO	tau = 1, eta = 1, alpha = 1, beta = 0.1, rho = 0.2, Pop = 100 , max_t = 50.
CS	lb = 0, ub = 1, $\theta = 0.5$, Pa = 0.25, alpha = 1, beta = 1.5, Pop = 100 , max_t = 50.
DE	lb = 0, ub = 1, $\theta = 0.5$, CR = 0.9, F = 0.5, Pop = 100 , max_t = 50.
GA	CR = 0.8, MR = 0.01, Pop = 100 , max_t = 50.
PSO	lb = 0, ub = 1, $\theta = 0.5$, c1 = 2, c2 = 2, w = 0.9, Vmax =(ub - lb)/2, Pop = 100 , max_t = 50.
ABC	lb = 0, ub = 1, $\theta = 0.5$, maxlimit = 5, Pop = 100 , max_t = 50.
GWO	lb = 0, ub = 1, $\theta = 0.5$, Pop = 100 , max_t = 50.

a well-known bagging method [60], while ADABOOST is a popular boosting method [22]. RUSBOOST is a random undersampling boosting method designed to address class imbalance [61]. SUBSPACE trains random feature subsets to reduce estimator correlation [62]. TOTALBOOST and LPBOOST aim to maximize the minimal margin of learned ensembles and have the ability to self-terminate [63], [64].

By comparing our method against these diverse algorithms, we aim to showcase its superiority and effectiveness in addressing the cancer gene expression data classification problem. Moreover, we have opened all computational model for public accessibility at “https://github.com/wangxb96/EODE/tree/master/ComparisonAlgorithms”.

C. Parameter Settings

Our experiments were conducted on a desktop computer with the following specifications: an Intel(R) Core(TM) i7-10700KF CPU @3.80 GHz, 32 GB of RAM, and a 64-bit Windows 10 operating system using Matlab 2021a. We utilized six base classifiers, namely DISCR, DT, KNN, ANN, SVM, and NB, to construct the ensemble. The parameters for DISCR, KNN, SVM, and NB are summarized in Table II, while the rest of the classifiers were used with their default settings. Additionally, Random Forest (RF) [60], ADABOOST [22], RUSBOOST [61], SUBSPACE [62], TOTALBOOST [63], and LPBOOST [64] were employed with their default parameter values. Furthermore, the parameters for four novel ensemble classifier methods, namely PSOEL [27], EAEL [57], FESM [58], and GA-Bagging-SVM [59], were set to be consistent with the original papers.

In our experiments, the original data was randomly divided into training and test datasets with an 8:2 ratio. The five-fold cross-validation method was used for training the data. For the GWO algorithm in feature selection and ensemble optimization, the population size (P) was set to 100, the number of iterations was set to 50, and the threshold (θ) was set to 0.5. Specifically, the threshold value θ in our study is utilized as a criterion within the Grey Wolf Optimizer algorithm to determine feature selection, and is not directly related to actual gene expression values themselves. In the clustering phase, the parameter t was set to $\sqrt[5]{m}$. The detailed parameters of seven classical evolutionary algorithms, including ACO [51], CS [52], DE [53], GA [54], GWO [37], PSO [55], and ABC [56], are summarized in Table III, where the population size (P) and the maximum iteration (max_t) are set to the same values.

For ACO, “tau” denotes the pheromone value, “eta” denotes the heuristic desirability, “alpha” denotes the control pheromone, “beta” denotes the control heuristic, and “rho” denotes the pheromone trail decay coefficient, which is set to 0.2. For CS, “Pa” denotes the discovery rate, “alpha” denotes the constant, and “beta” denotes the Levy component. For DE, “CR” denotes the crossover rate, and “F” denotes the scale factor. For GA, “CR” denotes the crossover rate, and “MR” denotes the mutation rate. For PSO, “c1” denotes the cognitive factor, “c2” denotes the social factor, “w” denotes the inertia weight, and “Vmax” denotes the maximum velocity.

IV. RESULTS AND ANALYSIS

A. Performance Comparisons With Other Nature-Inspired Ensemble Learning Algorithms

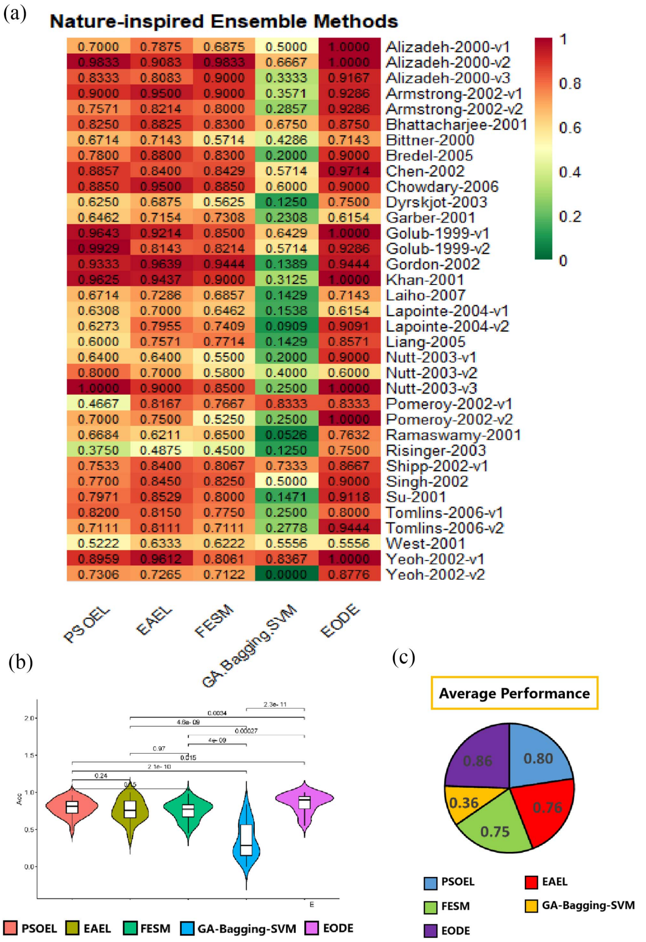
In our study, we conducted performance comparisons of EODE with several other nature-inspired ensemble learning algorithms, namely PSOEL, EAEL, FESM, and GA-Bagging-SVM. The experimental results are summarized in Fig. 3, where Fig. 3(A) presents detailed classification results, Fig. 3(B) illustrates the performance comparisons of EODE against the other ensemble methods, and Fig. 3(C) showcases the average performance values of these methods.

As shown in Fig. 3(A), EODE achieved the best results among all methods on 26 out of the 35 datasets. Specifically, EODE attained 100% classification accuracy on 7 datasets and achieved over 90% accuracy on more than half of the datasets. These results highlight the robustness of EODE in handling various types of cancers and its ability to provide highly accurate classifications. From Fig. 3(B), it is evident that EODE outperformed the other nature-inspired ensemble learning algorithms. The performance comparisons clearly demonstrate the superiority of EODE in terms of test accuracy. To provide a comprehensive performance overview, we present the average performance across all 35 cancer gene expression datasets in Fig. 3(C). The results indicate that EODE outperformed PSOEL by 6% and exhibited more than a 10% improvement compared to the other methods. These findings strongly support the conclusion that EODE performs better than other nature-inspired ensemble methods in the context of cancer gene expression classification.

B. Performance Comparisons of Different Machine Learning Algorithms

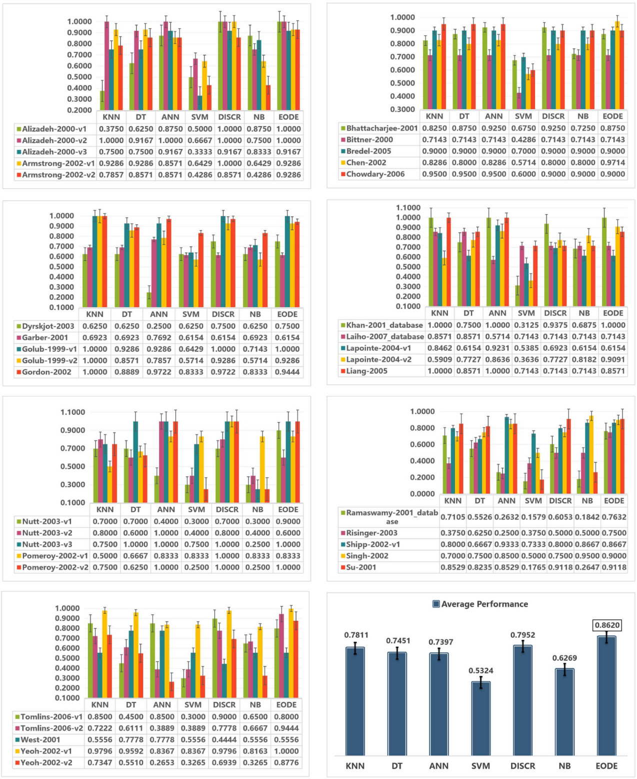
In our study, we conducted a comprehensive analysis and comparison of the performance between our proposed ensemble approach, EODE, and single classifier approaches. The experimental results, as shown in Fig. 4, clearly demonstrate the superiority of EODE in terms of classification accuracy for cancer gene expression datasets. EODE achieved the best classification accuracy for over 55% of the datasets, surpassing all single classifiers. This indicates the effectiveness and robustness of our ensemble approach in handling cancer gene expression classification tasks.

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.



PSOEL	EAEL	FESM	GA.Bagging.SVM	EODE	
0.7000	0.7875	0.6875	0.5000	1.0000	Alizadeh-2000-v1
0.9833	0.9083	0.9833	0.6667	1.0000	Alizadeh-2000-v2
0.8333	0.8083	0.9000	0.3333	0.9167	Alizadeh-2000-v3
0.9000	0.9500	0.9000	0.3571	0.9286	Armstrong-2002-v1
0.7571	0.8214	0.8000	0.2857	0.9286	Armstrong-2002-v2
0.8250	0.8825	0.8300	0.6750	0.8750	Bhattacharjee-2001
0.6714	0.7143	0.5714	0.4286	0.7143	Bittner-2000
0.7800	0.8800	0.8300	0.2000	0.9000	Bredel-2005
0.8857	0.8400	0.8429	0.5714	0.9714	Chen-2002
0.8850	0.9500	0.8850	0.6000	0.9000	Chowdary-2006
0.6250	0.6875	0.5625	0.1250	0.7500	Dyrskjot-2003
0.6462	0.7154	0.7308	0.2308	0.6154	Garber-2001
0.9643	0.9214	0.8500	0.6429	1.0000	Golub-1999-v1
0.9929	0.8143	0.8214	0.5714	0.9286	Golub-1999-v2
0.9333	0.9639	0.9444	0.1389	0.9444	Gordon-2002
0.9625	0.9437	0.9000	0.3125	1.0000	Khan-2001
0.6714	0.7286	0.6857	0.1429	0.7143	Laiho-2007
0.6308	0.7000	0.6462	0.1538	0.6154	Lapointe-2004-v1
0.6273	0.7955	0.7409	0.0909	0.9091	Lapointe-2004-v2
0.6000	0.7571	0.7714	0.1429	0.8571	Liang-2005
0.6400	0.6400	0.5500	0.2000	0.9000	Nutt-2003-v1
0.8000	0.7000	0.5800	0.4000	0.6000	Nutt-2003-v2
1.0000	0.9000	0.8500	0.2500	1.0000	Nutt-2003-v3
0.4667	0.8167	0.7667	0.8333	0.8333	Pomeroy-2002-v1
0.7000	0.7500	0.5250	0.2500	1.0000	Pomeroy-2002-v2
0.6684	0.6211	0.6500	0.0526	0.7632	Ramaswamy-2001
0.3750	0.4875	0.4500	0.1250	0.7500	Risinger-2003
0.7533	0.8400	0.8067	0.7333	0.8667	Shipp-2002-v1
0.7700	0.8450	0.8250	0.5000	0.9000	Singh-2002
0.7971	0.8529	0.8000	0.1471	0.9118	Su-2001
0.8200	0.8150	0.7750	0.2500	0.8000	Tomlins-2006-v1
0.7111	0.8111	0.7111	0.2778	0.9444	Tomlins-2006-v2
0.5222	0.6333	0.6222	0.5556	0.5556	West-2001
0.8959	0.9612	0.8061	0.8367	1.0000	Yeoh-2002-v1
0.7306	0.7265	0.7122	0.0000	0.8776	Yeoh-2002-v2

Fig. 3. Performance comparison to the other nature-inspired ensemble learning algorithms. (a) Test classification results of EODE and four other nature-inspired ensemble methods across the 35 cancer gene expression datasets. (b) Comparison graphs of EODE and the other four nature-inspired ensemble methods. (c) The average performance of EODE and the other four nature-inspired ensemble methods across the 35 cancer gene expression datasets.



	KNN	DT	ANN	SVM	DISCR	NB	EODE
🟩 Alizadeh-2000-v1	0.3750	0.6250	0.8750	0.5000	1.0000	0.8750	1.0000
🟥 Alizadeh-2000-v2	1.0000	0.9167	1.0000	0.6667	1.0000	0.7500	1.0000
■ Alizadeh-2000-v3	0.7500	0.7500	0.9167	0.3333	0.9167	0.8333	0.9167
🟨 Armstrong-2002-v1	0.9286	0.9286	0.8571	0.6429	1.0000	0.6429	0.9286
🟥 Armstrong-2002-v2	0.7857	0.8571	0.8571	0.4286	0.8571	0.4286	0.9286

	KNN	DT	ANN	SVM	DISCR	NB	EODE
🟩 Bhattacharjee-2001	0.8250	0.8750	0.9250	0.6750	0.9250	0.7250	0.8750
🟥 Bittner-2000	0.7143	0.7143	0.7143	0.4286	0.7143	0.7143	0.7143
■ Bredel-2005	0.9000	0.9000	0.9000	0.7000	0.9000	0.9000	0.9000
🟨 Chen-2002	0.8286	0.8000	0.8286	0.5714	0.8000	0.8000	0.9714
🟥 Chowdary-2006	0.9500	0.9500	0.9500	0.6000	0.9000	0.9000	0.9000

	KNN	DT	ANN	SVM	DISCR	NB	EODE
🟩 Dyrskjot-2003	0.6250	0.6250	0.2500	0.6250	0.7500	0.6250	0.7500
🟥 Garber-2001	0.6923	0.6923	0.7692	0.6154	0.6154	0.6923	0.6154
■ Golub-1999-v1	1.0000	0.9286	0.9286	0.6429	1.0000	0.7143	1.0000
🟨 Golub-1999-v2	1.0000	0.8571	0.7857	0.5714	0.9286	0.5714	0.9286
🟥 Gordon-2002	1.0000	0.8889	0.9722	0.8333	0.9722	0.8333	0.9444

	KNN	DT	ANN	SVM	DISCR	NB	EODE
🟩 Khan-2001_database	1.0000	0.7500	1.0000	0.3125	0.9375	0.6875	1.0000
🟥 Laiho-2007_database	0.8571	0.8571	0.5714	0.7143	0.7143	0.7143	0.7143
■ Lapointe-2004-v1	0.8462	0.6154	0.9231	0.5385	0.6923	0.6154	0.6154
🟨 Lapointe-2004-v2	0.5909	0.7727	0.8636	0.3636	0.7727	0.8182	0.9091
🟥 Liang-2005	1.0000	0.8571	1.0000	0.7143	0.7143	0.7143	0.8571

	KNN	DT	ANN	SVM	DISCR	NB	EODE
🟩 Nutt-2003-v1	0.7000	0.7000	0.4000	0.3000	0.7000	0.3000	0.9000
🟥 Nutt-2003-v2	0.8000	0.6000	1.0000	0.4000	0.8000	0.4000	0.6000
■ Nutt-2003-v3	0.7500	1.0000	1.0000	0.7500	1.0000	0.2500	1.0000
🟨 Pomeroy-2002-v1	0.5000	0.6667	0.8333	0.8333	1.0000	0.8333	0.8333
🟥 Pomeroy-2002-v2	0.7500	0.6250	1.0000	0.2500	1.0000	0.2500	1.0000

	KNN	DT	ANN	SVM	DISCR	NB	EODE
🟩 Ramaswamy-2001_database	0.7105	0.5526	0.2632	0.1579	0.6053	0.1842	0.7632
🟥 Risinger-2003	0.3750	0.6250	0.2500	0.3750	0.5000	0.5000	0.7500
■ Shipp-2002-v1	0.8000	0.6667	0.9333	0.7333	0.8000	0.8667	0.8667
🟨 Singh-2002	0.7000	0.7500	0.8500	0.5000	0.7500	0.9500	0.9000
🟥 Su-2001	0.8529	0.8235	0.8529	0.1765	0.9118	0.2647	0.9118

	KNN	DT	ANN	SVM	DISCR	NB	EODE
🟩 Tomlins-2006-v1	0.8500	0.4500	0.8500	0.3000	0.9000	0.6500	0.8000
🟥 Tomlins-2006-v2	0.7222	0.6111	0.3889	0.3889	0.7778	0.6667	0.9444
■ West-2001	0.5556	0.7778	0.7778	0.5556	0.4444	0.5556	0.5556
🟨 Yeoh-2002-v1	0.9796	0.9592	0.8367	0.8367	0.9796	0.8163	1.0000
🟥 Yeoh-2002-v2	0.7347	0.5510	0.2653	0.3265	0.6939	0.3265	0.8776

Fig. 4. Performance comparisons of the different machine learning algorithms. The first 7 graphs represent the test classification accuracy on the different cancer gene expression datasets, and the last graph indicates the average performance of the seven methods on the 35 datasets.

Moreover, when considering the average performance across all 35 cancer gene expression datasets, EODE consistently outperformed all single classifiers. Specifically, our ensemble approach exhibited remarkable improvements compared to the worst classifier, with an increase in performance of nearly 33%. Furthermore, EODE consistently achieved performance improvements of more than 10% compared to the majority of the base classifiers.

These findings clearly highlight the advantages of our ensemble approach over traditional single classifier methods. By leveraging the collective wisdom of multiple classifiers, EODE effectively addresses the challenges posed by cancer gene expression classification, resulting in superior classification accuracy and overall performance. Fig. 4 provides a visual representation of the experimental results, further supporting the conclusions drawn from our performance comparisons. The results validate the effectiveness of our proposed ensemble approach, highlighting its potential as a valuable tool in the field of cancer gene expression analysis.

C. *Performance Comparisons of the Different Evolutionary Algorithms*

To further evaluate the performance of the proposed EODE method, we compared it against other state-of-the-art evolutionary algorithms, including: Ant Colony Optimization (ACO), Cuckoo Search (CS), Differential Evolution (DE), Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), Particle Swarm Optimization (PSO), and Artificial Bee Colony (ABC).

The experimental results are summarized in supplementary Figs. 1 and 2. Supplementary Fig. 1 shows the classification accuracy of different methods on each of the 35 cancer gene expression datasets, with the first 7 sub-figures presenting results on individual datasets and the last sub-figure reporting the average performance across all datasets. As seen in supplementary Fig. 1, EODE obtains the best classification accuracy on over 60% of the datasets. Notably, there is an improvement of 5-8% in average classification accuracy achieved by EODE compared to other evolutionary algorithms. Supplementary Fig. 2 depicts the number of features (i.e., biomarker genes) selected by each method on each dataset. We can observe that EODE selects the smallest feature subset in nearly 60% of datasets, indicating its ability to identify the most informative genes.

Overall, from both supplementary Figs. 1 and 2, we can deduce that EODE consistently demonstrates the best average performance across all 35 cancer gene expression datasets, outperforming other state-of-the-art evolutionary methods. This

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.

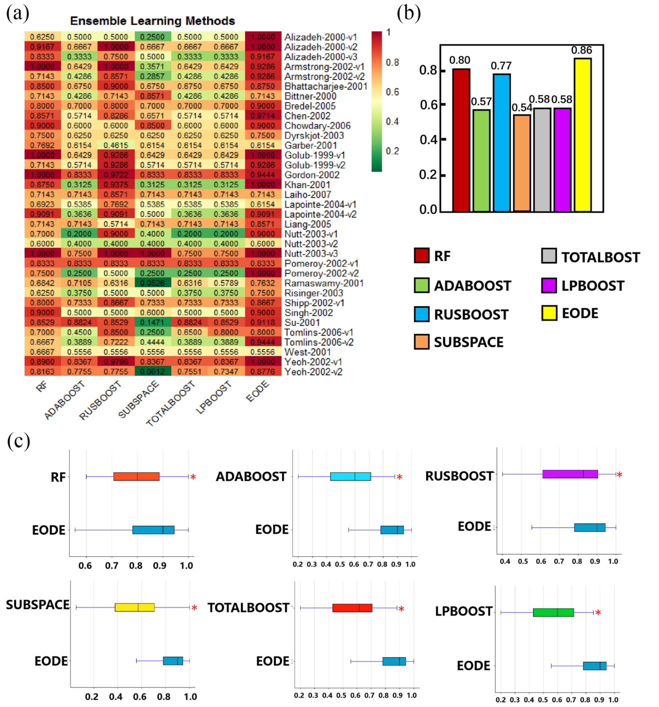


Fig. 5. Performance comparisons of the different ensemble learning algorithms. (a) Test classification results of EODE and six other ensemble methods across the 35 cancer gene expression datasets. (b) The average performance of EODE and the six other ensemble classifiers on the 35 datasets. (c) Graphs of EODE versus the other ensemble classifiers, where RF denotes Random Forest.

validates the effectiveness and robustness of the proposed EODE approach in discovering critical biomarker genes for cancer classification.

D. Performance Comparisons of the Different Ensemble Learning Algorithms

To further validate the effectiveness of the proposed EODE method, we conducted experiments comparing its performance to other state-of-the-art ensemble learning classifiers on the 35 cancer gene expression datasets. The methods considered for comparison include: Random Forest (RF) [60], ADABOOST [22], RUSBOOST [61], SUBSPACE [62], TOTALBOOST [63] and LPBOOST [64].

The results are shown in Fig. 5. Fig. 5(A) depicts a heat map of the classification accuracy of different methods on each of the 35 datasets, where darker colors indicate better performance. This heat map visualization allows us to qualitatively compare the performance of models across various cancer types. Fig. 5(B) summarizes the mean classification accuracy of each method averaged over the 35 datasets. The proposed EODE method achieves 6-32% better performance compared to other ensemble learning classifiers, demonstrating its superior predictive ability. Fig. 5(C) presents box plots to compare the distribution of classification accuracies obtained by each method on different datasets. We can observe that the median accuracy of EODE is higher than all other methods, indicating its stable and robust performance. Moreover, the box plot of EODE is more narrow compared to others, showing the consistency of results obtained.

Overall, these quantitative and qualitative comparisons presented in Fig. 5 validate that the proposed EODE method achieves the best classification accuracy on over 70% of the cancer gene expression datasets, outperforming other state-of-the-art ensemble classifiers. This clearly demonstrates the effectiveness and robustness of the EODE approach for cancer classification using gene expression data.

E. Ablation Study

1) Performance of EODE Without Ensemble Learning: Table IV presents a comprehensive evaluation of training and testing performance across 35 datasets using the proposed EODE approach against EODE without ensemble learning (WEL). Here, WEL means the nature-inspired diverse ensemble learning is not employed. On analysis, it is evident that the training accuracy of both EODE and WEL are comparable, with the EODE approach achieving a slightly higher average training accuracy of 0.9524 versus 0.9498 for WEL. This indicates that the model capacity for fitting the training data is similar between the two approaches. However, EODE demonstrates a significant test accuracy advantage over WEL, with average test accuracies of 0.8620 and 0.5456 respectively. This translates to an absolute improvement of over 30% in generalization performance by leveraging ensemble learning.

The key insight is that while ensemble learning does not markedly improve training fit, it provides superior generalization through effectively preventing overfitting. Single models are prone to overfitting the noise in small datasets. Ensemble learning creates multiple diverse models and aggregates their predictions, avoiding these spurious patterns. Across multiple datasets, EODE consistently exhibits stronger generalization, evidenced by the significantly higher test accuracies. This gap is particularly prominent in smaller datasets where individual models tend to overfit more. By reducing variance via ensembling, the proposed approach demonstrates more robust predictions on unseen test data. The results validate the effectiveness of ensemble learning in enhancing model generalization capability and tackling the overfitting challenge.

In conclusion, the ensemble framework shows considerable promise in boosting test performance over single model baseline across a wide range of conditions. This has important implications for real-world applications like speech emotion recognition where avoiding overfitting is critical. The analysis provides strong empirical evidence and rationale for adopting ensemble techniques.

2) Performance of EODE Ensemble versus Individual Classifiers: Unlike the analysis in Section IV-B, this study does not evaluate each base classifier model in isolation. Rather, this section investigates the impact of using a single base classifier within the nature-inspired diverse ensemble learning phase of the proposed approach, instead of aggregating multiple heterogeneous classifiers concurrently as intended in the ensemble methodology. By focusing on the ensemble learning stage, this analysis provides targeted insight into the benefits of leveraging diversity in the classifier combinations compared to relying on any individual modeling paradigm alone during this critical step.

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
PERFORMANCE ON TRAINING AND TESTING SETS WITH AND WITHOUT ENSEMBLE LEARNING (WEL) METHOD

Datasets	Training Accuracy		Test Accuracy		Datasets	Training Accuracy		Test Accuracy	
	WEL	EODE	WEL	EODE		WEL	EODE	WEL	EODE
Alizadeh-2000-v1	**1.0000**	**1.0000**	0.5114	**1.0000**	Lapointe-2004-v2	**0.8737**	0.8634	0.4339	**0.9091**
Alizadeh-2000-v2	**1.0000**	**1.0000**	0.6591	**1.0000**	Liang-2005	0.9515	**0.9667**	0.6494	**0.8571**
Alizadeh-2000-v3	**0.9745**	0.9600	0.4318	**0.9167**	Nutt-2003-v1	0.8614	**0.9250**	0.4273	**0.9000**
Armstrong-2002-v1	**1.0000**	0.9818	0.6558	**0.9286**	Nutt-2003-v2	**1.0000**	**1.0000**	0.5455	**0.6000**
Armstrong-2002-v2	**1.0000**	**1.0000**	0.4416	**0.9286**	Nutt-2003-v3	**1.0000**	**1.0000**	0.7955	**1.0000**
Bhattacharjee-2001	**0.9944**	0.9938	0.6750	**0.8750**	Pomeroy-2002-v1	**0.9697**	0.9667	**0.8333**	**0.8333**
Bittner-2000	0.9614	**0.9667**	0.5857	**0.7143**	Pomeroy-2002-v2	**0.9532**	0.9381	0.2841	**1.0000**
Bredel-2005	0.9114	**0.9250**	0.5727	**0.9000**	Ramaswamy-2001_database	0.7251	**0.7966**	0.1794	**0.7632**
Chen-2002	0.9918	**0.9931**	0.6649	**0.9714**	Risinger-2003	**0.8649**	0.8238	0.4545	**0.7500**
Chowdary-2006	**0.9904**	0.9875	0.8682	**0.9000**	Shipp-2002-v1	0.9800	**0.9833**	0.7212	**0.8667**
Dyrskjot-2003	**0.9948**	0.9381	0.5568	**0.7500**	Singh-2002	**0.9778**	0.9750	0.5955	**0.9000**
Garber-2001	**0.8883**	0.8873	0.5455	**0.6154**	Su-2001	0.9565	**0.9643**	0.1925	**0.9118**
Golub-1999-v1	0.9970	**1.0000**	0.6234	**1.0000**	Tomlins-2006-v1	0.8787	**0.8809**	0.3364	**0.8000**
Golub-1999-v2	0.9939	**1.0000**	0.5065	**0.9286**	Tomlins-2006-v2	**0.8721**	0.8648	0.3889	**0.9444**
Gordon-2002	**1.0000**	**1.0000**	0.8409	**0.9444**	West-2001	**1.0000**	**1.0000**	0.4646	**0.5556**
Khan-2001_database	**1.0000**	**1.0000**	0.3466	**1.0000**	Yeoh-2002-v1	**0.9950**	**0.9950**	0.8108	**1.0000**
Laiho-2007_database	**1.0000**	**1.0000**	**0.7143**	**0.7143**	Yeoh-2002-v2	0.8448	**0.8995**	0.2430	**0.8776**
Lapointe-2004-v1	0.8423	**0.8591**	0.5385	**0.6154**	**Average**	0.9498	**0.9524**	0.5456	**0.8620**

The bold values indicate that the method achieves the best performance on the corresponding dataset.

TABLE V
PERFORMANCE OF EODE ENSEMBLE VERSUS INDIVIDUAL BASE CLASSIFIERS

Dataset	**DISCR**	**DT**	**KNN**	**ANN**	**SVM**	**NB**	**EODE**
Alizadeh-2000-v1	0.8333	0.6875	0.7750	0.6875	0.5000	0.7500	**1.0000**
Alizadeh-2000-v2	**1.0000**	0.8452	0.9833	**1.0000**	0.6667	0.7333	**1.0000**
Alizadeh-2000-v3	**0.9306**	0.7381	0.8667	0.9271	0.3333	0.8000	0.9167
Armstrong-2002-v1	0.8929	**0.9490**	0.9143	0.8929	0.6429	0.8429	0.9286
Armstrong-2002-v2	0.8690	0.7959	0.8000	0.8482	0.4286	0.6714	**0.9286**
Bhattacharjee-2001	**0.9292**	0.8321	0.8350	0.9063	0.6750	0.7750	0.8750
Bittner-2000	0.6905	0.6531	**0.7143**	0.6786	0.4286	0.5714	**0.7143**
Bredel-2005	0.8667	0.6857	0.7800	0.8500	0.7000	0.8400	**0.9000**
Chen-2002	0.8619	0.7837	0.8171	0.8786	0.5829	0.8229	**0.9714**
Chowdary-2006	0.9000	0.9214	0.9300	**0.9438**	0.8900	0.8700	0.9000
Dyrskjot-2003	**0.7500**	0.6429	0.6750	0.6719	0.6250	0.6500	**0.7500**
Garber-2001	**0.7051**	0.6923	0.6769	0.6538	0.6154	0.6769	0.6154
Golub-1999-v1	0.9286	0.9184	0.9143	0.8482	0.6429	0.8000	**1.0000**
Golub-1999-v2	0.8810	0.8367	0.8857	0.7500	0.5714	0.6714	**0.9286**
Gordon-2002	**0.9861**	0.9643	0.9722	0.9757	0.8333	0.9500	0.9444
Khan-2001_database	0.8646	0.8571	0.9125	0.9297	0.3125	0.7000	**1.0000**
Laiho-2007_database	0.7381	**0.7551**	0.7143	0.6786	0.7143	0.7429	0.7143
Lapointe-2004-v1	0.7538	0.6264	**0.8154**	0.6923	0.5385	0.6000	0.6154
Lapointe-2004-v2	0.7386	0.5195	0.5909	0.7330	0.3636	0.7091	**0.9091**
Liang-2005	0.6786	0.7347	0.8000	**0.8750**	0.7143	0.7143	0.8571
Nutt-2003-v1	0.7000	0.3857	0.5800	0.5500	0.2800	0.3800	**0.9000**
Nutt-2003-v2	**0.8000**	0.4286	0.6800	0.6500	0.4400	0.4000	0.6000
Nutt-2003-v3	**1.0000**	0.8214	0.8500	**1.0000**	0.7500	0.8500	**1.0000**
Pomeroy-2002-v1	0.6250	0.6429	0.6333	0.6250	0.1667	0.6000	**0.8333**
Pomeroy-2002-v2	0.8750	0.4107	0.5500	0.5156	0.2500	0.2500	**1.0000**
Ramaswamy-2001	0.6250	0.5752	0.6263	0.5329	0.1579	0.2526	**0.7632**
Risinger-2003	0.5938	0.5000	0.3750	0.5625	0.3750	0.5000	**0.7500**
Shipp-2002-v1	0.7667	0.7619	0.8000	**0.8667**	0.7333	0.7733	**0.8667**
Singh-2002	0.8125	0.7833	0.7600	0.8188	0.5000	0.8500	**0.9000**
Su-2001	0.8676	0.7549	0.7588	0.8750	0.1588	0.4471	**0.9118**
Tomlins-2006-v1	0.7875	0.5250	**0.8200**	0.8000	0.3000	0.6800	0.8000
Tomlins-2006-v2	0.7778	0.6000	0.7000	0.7986	0.3889	0.6556	**0.9444**
West-2001	0.5833	**0.7778**	0.6444	0.5972	0.4667	0.6000	0.5556
Yeoh-2002-v1	0.9796	0.9714	0.9878	0.9566	0.6694	0.8327	**1.0000**
Yeoh-2002-v2	0.6735	0.6408	0.7673	0.5357	0.3265	0.3347	**0.8776**
Average	0.8076	0.7148	0.7687	0.7744	0.5069	0.6656	**0.8620**

The bold values indicate that the method achieves the best performance on the corresponding dataset.

As shown in Table V, across the 35 gene expression datasets analyzed, the best single classifier achieved an average accuracy of 0.8076 using a DISCR model. In contrast, the proposed EODE ensemble approach attained a significantly higher accuracy of 0.8620 by leveraging an integrated combination of diverse classifiers including DISCR, DT, KNN, ANN, SVM and NB. The results highlight that relying on any individual base classifier is suboptimal compared to the ensemble approach. No single modeling paradigm consistently dominates the performance across all datasets, due to the complexity of the classification problem. Different datasets exhibit variability in terms of which individual classifier achieves the best performance when used alone. However, EODE provides equal or higher accuracy relative to the top stand-alone model on 23 out of 35 datasets. The results empirically demonstrate that integrating multiple complementary base classifiers simultaneously is essential to maximize the potential of the ensemble framework and attain optimal classification performance on gene expression data. Reliance on any single constituent classifier within the ensemble learning process fails to harness the full synergistic advantages of the diverse ensemble.

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.

V. Conclusion

Cancer type identification is a critical aspect of cancer research, as it enables early diagnosis and tailored treatment for patients. One key challenge in this field is identifying the highly sensitive biomarker genes that are indicative of specific cancer types. In this study, we propose a novel approach called EODE to address the classification of cancer types, particularly in scenarios where the gene expression profile samples are high-dimensional and small in size. EODE leverages the grey wolf optimizer (GWO) to optimize feature subsets and collaboratively builds an optimized ensemble classifier. By combining nature-inspired feature selection and ensemble learning, EODE significantly improves the model's identification capability.

We conducted experiments on 35 datasets encompassing various cancer types, and the results demonstrate the effectiveness of our algorithm compared to four nature-inspired ensemble methods (PSOEL, EAEL, FESM, and GA-Bagging-SVM), six benchmark machine learning algorithms (KNN, DT, ANN, SVM, DISCR, and NB), six state-of-the-art ensemble algorithms (RF, ADABOOST, RUSBOOST, SUBSPACE, TOTALBOOST, and LPBOOST), and seven nature-inspired methods (ACO, CS, DE, GA, GWO, PSO, and ABC). Our algorithm outperformed these methods in terms of classification accuracy.

In future work, we aim to enhance the efficiency of the algorithm by improving the screening of redundant and invalid features. Additionally, as biomedical data often exhibit class imbalance, we plan to ensure robust results on class-imbalanced data. Beyond computational refinements, we intend to evaluate the proposed methodology on expanded gene expression datasets from diverse clinical cohorts. As cancer subtyping using gene expression data holds great promise for guiding individualized treatment decisions, we hope to transition this computational pipeline into real-world clinical settings.

References

[1] W. Cao, H.-D. Chen, Y.-W. Yu, N. Li, and W.-Q. Chen, "Changing profiles of cancer burden worldwide and in China: A secondary analysis of the global cancer statistics 2020," *Chin. Med. J.*, vol. 134, no. 07, pp. 783–791, 2021.

[2] K. Swanson, E. Wu, A. Zhang, A. A. Alizadeh, and J. Zou, "From patterns to patients: Advances in clinical machine learning for cancer diagnosis, prognosis, and treatment," *Cell*, vol. 186, pp. 1772–1791, 2023.

[3] W. L. Bi et al., "Artificial intelligence in cancer imaging: Clinical challenges and applications," *CA: A Cancer J. Clinicians*, vol. 69, no. 2, pp. 127–157, 2019.

[4] J. Mateo et al., "Delivering precision oncology to patients with cancer," *Nature Med.*, vol. 28, no. 4, pp. 658–665, 2022.

[5] D.-S. Huang and C.-H. Zheng, "Independent component analysis-based penalized discriminant method for tumor classification using gene expression data," *Bioinformatics*, vol. 22, no. 15, pp. 1855–1862, 2006.

[6] R. Su, J. Zhang, X. Liu, and L. Wei, "Identification of expression signatures for non-small-cell lung carcinoma subtype classification," *Bioinformatics*, vol. 36, no. 2, pp. 339–346, 2020.

[7] C. Qu et al., "Improving feature selection performance for classification of gene expression data using harris hawks optimizer with variable neighborhood learning," *Brief. Bioinf.*, vol. 22, 2021, Art. no. bbab097.

[8] H. S. Parker et al., "Preserving biological heterogeneity with a permuted surrogate variable analysis for genomics batch correction," *Bioinformatics*, vol. 30, no. 19, pp. 2757–2763, 2014.

[9] F. Schmidt, M. List, E. Cukuroglu, S. Köhler, J. Göke, and M. H. Schulz, "An ontology-based method for assessing batch effect adjustment approaches in heterogeneous datasets," *Bioinformatics*, vol. 34, no. 17, pp. i908–i916, 2018.

[10] T. Jin, N. D. Nguyen, F. Talos, and D. Wang, "ECMarker: Interpretable machine learning model identifies gene expression biomarkers predicting clinical outcomes and reveals molecular mechanisms of human disease in early stages," *Bioinformatics*, vol. 37, no. 8, pp. 1115–1124, 2021.

[11] B. He et al., "Integrating spatial gene expression and breast tumour morphology via deep learning," *Nature Biomed. Eng.*, vol. 4, no. 8, pp. 827–834, 2020.

[12] H. Gao, C. Bian, X. Wang, X. Li, and Y. Wang, "Exploring cancer biomarker genes from gene expression data via natureinspired multiobjective optimization," in *Proc. 34th Chin. Control Decis. Conf.*, 2022, pp. 5000–5007.

[13] X. Wang and W. Jia, "A feature weighting particle swarm optimization method to identify biomarker genes," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, 2022, pp. 830–834.

[14] S. Nemati, M. E. Basiri, N. Ghasem-Aghaee, and M. H. Aghdam, "A novel ACO–GA hybrid algorithm for feature selection in protein function prediction," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 12086–12094, 2009.

[15] N. Maleki, Y. Zeinali, and S. T. A. Niaki, "A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection," *Expert Syst. Appl.*, vol. 164, 2021, Art. no. 113981.

[16] R. C. T. de Souza, C. A. de Macedo, L. dos Santos Coelho, J. Pierezan, and V. C. Mariani, "Binary coyote optimization algorithm for feature selection," *Pattern Recognit.*, vol. 107, 2020, Art. no. 107470.

[17] G. Dhiman et al., "BEPO: A novel binary emperor penguin optimizer for automatic feature selection," *Knowl.-Based Syst.*, vol. 211, 2021, Art. no. 106560.

[18] A. I. Hammouri, M. Mafarja, M. A. Al-Betar, M. A. Awadallah, and I. Abu-Doush, "An improved dragonfly algorithm for feature selection," *Knowl.-Based Syst.*, vol. 203, 2020, Art. no. 106131.

[19] N. Neggaz, H. Essam Houssein, and K. Hussain, "An efficient henry gas solubility optimization for feature selection," *Expert Syst. Appl.*, vol. 152, 2020, Art. no. 113364.

[20] M. Pratama, W. Pedrycz, and E. Lughofer, "Evolving ensemble fuzzy classifier," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 2552–2567, Oct. 2018.

[21] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[22] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[23] R. Shen, A. B. Olshen, and M. Ladanyi, "Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis," *Bioinformatics*, vol. 25, no. 22, pp. 2906–2912, 2009.

[24] Z. Cao, X. Pan, Y. Yang, Y. Huang, and H.-B. Shen, "The lnclocator: A subcellular localization predictor for long non-coding RNAs based on a stacked ensemble classifier," *Bioinformatics*, vol. 34, no. 13, pp. 2185–2194, 2018.

[25] R. Su, X. Liu, G. Xiao, and L. Wei, "Meta-GDBP: A high-level stacked regression model to improve anticancer drug response prediction," *Brief. Bioinf.*, vol. 21, no. 3, pp. 996–1005, 2020.

[26] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: A survey and categorisation," *Inf. Fusion*, vol. 6, no. 1, pp. 5–20, 2005.

[27] M. Z. Jan, J. C. Munoz, and M. A. Ali, "A novel method for creating an optimized ensemble classifier by introducing cluster size reduction and diversity," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 7, pp. 3072–3081, Jul. 2022.

[28] T. T. Nguyen, A. V. Luong, M. T. Dang, A. W.-C. Liew, and J. McCall, "Ensemble selection based on classifier prediction confidence," *Pattern Recognit.*, vol. 100, 2020, Art. no. 107104.

[29] Y. Chen, M.-L. Wong, and H. Li, "Applying ant colony optimization to configuring stacking ensembles for data mining," *Expert Syst. Appl.*, vol. 41, no. 6, pp. 2688–2702, 2014.

[30] A. K. Das, S. K. Pati, and A. Ghosh, "Relevant feature selection and ensemble classifier design using bi-objective genetic algorithm," *Knowl. Inf. Syst.*, vol. 62, no. 2, pp. 423–455, 2020.

[31] X. Li, S. Zhang, and K.-C. Wong, "Single-cell RNA-seq interpretations using evolutionary multiobjective ensemble pruning," *Bioinformatics*, vol. 35, no. 16, pp. 2809–2817, 2019.

[32] S. Zhu et al., "The genetic algorithm-aided three-stage ensemble learning method identified a robust survival risk score in patients with glioma," *Brief. Bioinf.*, vol. 23, no. 5, 2022, Art. no. bbac344.

[33] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.

[34] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Mach. Learn.*, vol. 51, pp. 181–207, 2003.

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.

[35] Y. Zhang, S. Burer, W. N. Street, K. P. Bennett, and E. Parrado-Hern ández, "Ensemble pruning via semi-definite programming," *J. Mach. Learn. Res.*, vol. 7, no. 7, pp. 1315–1338, 2006.

[36] L. Rokach, "Ensemble-based classifiers," *Artif. Intell. Rev.*, vol. 33, pp. 1–39, 2010.

[37] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.

[38] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, 2018.

[39] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013.

[40] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognit. Lett.*, vol. 28, no. 4, pp. 459–471, 2007.

[41] D. J. C. MacKay and D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[42] R. Meir, M. Polukarov, J. Rosenschein, and N. Jennings, "Convergence to equilibria in plurality voting," in *Proc. AAAI Conf. Artif. Intell.*, 2010, pp. 823–828.

[43] R. Meir, "Plurality voting under uncertainty," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 2103–2109.

[44] M. Cp De Souto, I. G. Costa, D. Sa De Araujo, T. B. Ludermir, and A. Schliep, "Clustering cancer gene expression data: A comparative study," *BMC Bioinf.*, vol. 9, no. 1, pp. 1–14, 2008.

[45] P. A. Lachenbruch and M. Goldstein, "Discriminant analysis," *Biometrics*, vol. 35, no. 1, pp. 69–85, 1979, doi: 10.2307/2529937.

[46] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 3, pp. 660–674, May/Jun. 1991.

[47] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[48] B. Yegnanarayana, *Artificial Neural Networks*. India: PHI Learning Pvt. Ltd., 2009.

[49] W. S. Noble, "What is a support vector machine?," *Nature Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, 2006.

[50] K. P. Murphy et al., "Naive bayes classifiers," *Univ. Brit. Columbia*, vol. 18, no. 60, pp. 1–8, 2006.

[51] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.

[52] X.-S. Yang and S. Deb, "Cuckoo search via lévy flights," in *Proc. World Congr. Nature Biologically Inspired Comput.*, 2009, pp. 210–214.

[53] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[54] D. Whitley, "A genetic algorithm tutorial," *Statist. Comput.*, vol. 4, no. 2, pp. 65–85, 1994.

[55] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, 1995, pp. 1942–1948.

[56] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Glob. Optim.*, vol. 39, no. 3, pp. 459–471, 2007.

[57] Z. Md Jan and B. Verma, "Evolutionary classifier and cluster selection approach for ensemble classification," *ACM Trans. Knowl. Discov. Data*, vol. 14, no. 1, pp. 1–18, 2019.

[58] M. Zohaib Jan, "A novel framework for optimised ensemble classifiers," PhD dissertation, Sch. Eng. Technol., Central Queensland Univ., Rockhampton, Australia, 2020.

[59] J. Lin, H. Chen, S. Li, Y. Liu, X. Li, and B. Yu, "Accurate prediction of potential druggable proteins based on genetic algorithm and bagging-SVM ensemble classifier," *Artif. Intell. Med.*, vol. 98, pp. 35–47, 2019.

[60] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[61] C. Seiffert, M. TaghiJ. KhoshgoftaarV. Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," *IEEE Trans. Syst., Man, Cybern.-A, Syst. Humans*, vol. 40, no. 1, pp. 185–197, Jan. 2010.

[62] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[63] M. K. Warmuth, J. Liao, and G. Rätsch, "Totally corrective boosting algorithms that maximize the margin," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 1001–1008.

[64] A. J. Grove and D. Schuurmans, "Boosting in the limit: Maximizing the margin of learned ensembles," in *Proc. 15th Nat./10th Conf. Artif. Intell./Innovative Appl. Artif. Intell.*, 1998, pp. 692–699.

Xubin Wang is currently with the School of Artificial Intelligence, Jilin University, Changchun, China. His research interests span data mining, machine learning, and computational intelligence, with a focus on developing novel techniques for high-dimensional data analysis, feature selection, and knowledge discovery.

Yunhe Wang is an assistant professor with the School of Artificial Intelligence, Hebei University of Technology, Tianjin, China. Her current research interests include intelligent computation and machine learning.

Zhiqiang Ma received the PhD degree in computer science from Jilin University, in 2009. Currently, he is a professor with the Department of Information Science and Technology, Northeast Normal University. He is currently the vice president of the Research Association of Computer Education with Normal Universities of China, and the executive director of Jilin Computer Federation. His research interests include bioinformatics, software engineering, molecular biology, and data mining.

Ka-Chun Wong received the BEng degree in computer engineering from United College, Chinese University of Hong Kong, in 2008, and the MPhil degree from the Chinese University of Hong Kong, in 2010, and the PhD degree from the Department of Computer Science, University of Toronto, in 2014. He assumed his duty as an assistant professor with the City University of Hong Kong, in 2015. His research interests include bioinformatics, computational biology, evolutionary computation, data mining, machine learning, and interdisciplinary research. He is merited as the associate editor of *BioData Mining*, in 2016. In addition, he is on the editorial board of *Applied Soft Computing* since 2016. Remarkably, he has solely edited 2 books published by Springer and CRC Press, attracting 30 peer-reviewed book chapters around the world.

Xiangtao Li (Member, IEEE) received the BEng, MEng, and PhD degrees in computer science from North-east Normal University, Changchun, China, in 2009, 2012, and 2015, respectively. He is currently a professor with the School of Artificial Intelligence, Jilin University. He has published more than 80 research papers. His research interests include intelligent computation, evolutionary data mining, constrained optimization, multi objective optimization, and their applications.

Authorized licensed use limited to: Beijing Normal University. Downloaded on October 18,2024 at 09:16:20 UTC from IEEE Xplore. Restrictions apply.