Ant-TD with opposition-based learning for multi-label feature selection: When ant colony optimization and reinforcement learning meet opposition-based learning simultaneously

Sahar Shams Beiranvand [1], Vahid Mehrdad [2,*], Mohammad Bagher Dowlatshahi [3]

[1] *Department of Electrical Engineering, Faculty of Engineering, Lorestan University, Khorramabad, Iran.*
shams.s@fe.lu.ac.ir
Department of Electrical Engineering, Faculty of Engineering, Lorestan University, Khorramabad, Iran.
mehrdad.v@lu.ac.ir
[3] *Department of Computer Engineering, Faculty of Engineering, Lorestan University, Khorramabad, Iran.*
dowlatshahi.mb@lu.ac.ir

* *Corresponding author*

Abstract

This paper introduces an enhanced feature selection method for multi-label data using Ant Colony Optimization (ACO), augmented with Temporal Difference (TD) reinforcement learning and opposition-based learning (OBL). This novel integration of techniques aims to improve the efficiency and accuracy of feature selection. We propose three innovative algorithms that incorporate these methods, marking the first application of such a model to multi-label datasets. The proposed algorithms leverage ACO to navigate the feature space, while TD learning dynamically updates heuristic functions, specifically state value V, to better estimate future rewards for selected feature subsets. Additionally, OBL ensures comprehensive exploration by considering both original and opposite solutions. We evaluate our methods on nine diverse multi-label datasets using the Multi-Label K-Nearest Neighbors (MLKNN) classifier, with performance measured by accuracy and hamming loss. Experimental results demonstrate that our hybrid approach significantly outperforms traditional feature selection techniques, offering improved classification performance and more relevant feature subsets. This work establishes a new paradigm in multi-label feature selection, combining ACO, TD learning, and OBL for the first time.

1. Introduction

In data analysis and machine learning, the principle "less is more" frequently applies, particularly regarding the features or variables used in modeling. Feature selection, an essential step in data preprocessing, focuses on identifying and retaining the most relevant and informative features while eliminating those that are redundant or less useful [1]. This process significantly enhances the performance, interpretability, and efficiency of predictive models [2].

Feature selection acts as a strategic tool for data scientists, allowing them to simplify complex datasets into concise and effective models [3]. Its influence spans various fields, from predictive analytics to pattern recognition, enabling practitioners to extract actionable insights and make informed decisions from data [4]. Feature selection involves selecting a subset of relevant features or variables from a larger set available in a dataset [5]. The goal is to improve model performance, reduce computational complexity, enhance interpretability, and mitigate the risk of overfitting by focusing on the most informative features [6]. There are several approaches to feature selection, broadly categorized into three main types: Filter Methods, Wrapper Methods, and Embedded Methods. The choice of feature selection method depends on various factors, including dataset size, dimensionality, nature of features, computational resources, and desired model performance [7]. Experimentation and validation are essential to determine the most suitable feature selection approach for a specific task and dataset [8].

Multi-label data refers to datasets where each instance can be associated with multiple labels or categories simultaneously. In contrast to traditional single-label datasets, where each instance is assigned only one label, multi-label datasets allow instances to have multiple labels assigned to them [9]. Not all problems fit neatly into predefined categories. Some scenarios require a more nuanced approach, where each data point may belong to multiple categories or classes simultaneously [10]. This is where multilabel data comes into play, offering a flexible and powerful framework for addressing complex classification tasks. The need for multilabel data arises from the inherent complexity and ambiguity of real-world phenomena [11]. Many real-world scenarios exhibit overlapping or intersecting categories, making it unrealistic to force data points into mutually exclusive classes [12]. By embracing the multilabel paradigm, analysts and researchers can better capture the richness and diversity of the underlying data, leading to more nuanced and accurate models [13]. Handling multi-label data requires specialized techniques for data preprocessing, feature engineering, model training, and evaluation [14]. Algorithms designed for multi-label classification need to account for label dependencies, imbalanced label distributions, and the complexity of the label space [15].

ACO is a metaheuristic optimization algorithm inspired by the foraging behavior of ants. It was introduced by Marco Dorigo in the early 1990s as a method for solving combinatorial optimization problems [16]. ACO mimics the collective behavior of ants searching for food and effectively finds solutions to complex optimization problems through iterative exploration of solution space. The key idea behind ACO is the use of artificial ants to construct solutions to optimization problems [17]. These artificial ants traverse the solution space, gradually building solutions by probabilistically selecting components based on pheromone trails, which represent a form of indirect communication among ants [18]. Ants deposit pheromones on paths they traverse, and the strength of the pheromone trail influences the likelihood of other ants choosing the same path in subsequent iterations [19]. ACO has been successfully applied to various combinatorial optimization problems, including the traveling salesman problem, the quadratic assignment problem, job scheduling, and routing problems in telecommunication networks [20]. Its effectiveness lies in its ability to effectively explore the solution space, adaptively focus search efforts on promising regions, and exploit valuable information shared among artificial ants through pheromone communication [21].

OBL, also known as Opposition-based Optimization (OBO) or Opposition-based Computing (OBC), is a computational intelligence technique inspired by the concept of opposition in nature [22]. OBL aims to enhance optimization algorithms by incorporating the idea of opposition, which involves considering both the original and opposite states of solutions during the search process [23-24]. The fundamental principle of OBL is based on the belief that exploring both the original and opposite states of solutions can lead to better convergence and more effective exploration of the search space [25]. In optimization problems, the opposite of a solution is defined as the reflection or negation of its components. For instance, in continuous optimization, the opposite of a real-valued solution vector can be obtained by negating each component or by reflecting it around a predefined reference point [26].

Reinforcement learning (RL) has made significant strides and gained widespread adoption in various domains in recent years. RL is a machine learning paradigm where an agent learns to make sequential decisions by interacting with an environment to maximize cumulative rewards [27]. Unlike supervised learning, which involves training on labeled data, or unsupervised learning, which identifies patterns in unlabeled data, RL focuses on learning from feedback received from the environment [28]. In RL, the agent is the learner or decision-maker that interacts with the environment. The environment provides feedback to the agent based on its actions. Actions affect the state of the environment, and the state offers context for the agent's decisions. Rewards are feedback from the environment that indicate the desirability of the agent's actions. The goal of the agent is to maximize cumulative rewards over time [29-31]. In this article we explore various approaches and strategies for applying ACO to feature selection in multi-label data tasks. These include using OBL and temporal different in traditional ACO algorithms adapted for multi-label data feature selection, as well as hybrid approaches that combine ACO with other optimization techniques or feature selection methods. The article introduces OBL, a computational intelligence technique inspired by the concept of opposition in nature. OBL involves considering both the original and opposite states of solutions during the optimization process to enhance exploration and exploitation of the search space.

The article discusses the evaluation metrics commonly used to assess the performance of ACO-based feature selection methods, such as classification accuracy, computational efficiency, and stability. It also highlights the advantages and limitations of ACO in comparison to other feature selection techniques.

The proposed method includes the following important points:

- As far as we know, it is the first time that both OBL and TD methods are simultaneously used in ACO.
- For the first time, OBL is used to help heuristic learning in ACO.
- Innovation in the concurrent use of OBL in both pheromone and heuristic function
- The opposition method is used to select optimal features in multi-label data.

We have compared our proposed method with various criteria in 9 different datasets with other proposed methods in this field and have shown that using OBL as a heuristic function in the ACO algorithm has resulted in better performance.

This article is written in the following order. In Section 2, we examine related works in the field of multi-labeled data. In Section 3, we introduce the basic concepts used in the algorithm. In Section 4, we introduce the proposed method. In Section 5, algorithm settings and experiments are presented. Section 6 discusses the results, and finally, Section 7 concludes with a discussion and conclusion regarding the proposed method.

2. Related work

The field of multi-label feature selection has seen significant advancements with the application of ACO and related techniques. Several researchers have proposed innovative methods to enhance the performance of feature selection algorithms, leveraging the collective intelligence of artificial ants and other computational techniques.

MLACO, introduced by [32], represents a novel multi-label feature selection algorithm based on ACO. This algorithm is designed to improve multi-label classification tasks by identifying and selecting the most informative feature subsets from datasets. By efficiently exploring the feature space, MLACO enhances the performance of classification models through optimal feature subset selection. The proposed method, MFS-MCDM, integrates multi-criteria decision making (MCDM) techniques to identify and select the most relevant features from datasets with multiple labels. This approach aims to enhance the performance of classification models by effectively balancing various criteria involved in feature selection [33]. In a different approach, [35] proposed a method that combines multiple feature selection algorithms within a multi-criteria decision-making framework. This ensemble method aims to enhance the feature selection process by leveraging the strengths of various algorithms, resulting in a more robust and effective selection process. In the realm of feature selection, [39] proposed SemiACO, a semi-supervised feature selection method using ACO. This method addresses high-dimensional data challenges by selecting features with minimal redundancy and maximal relevance to the class label. SemiACO employs a nonlinear heuristic function and integrates TD reinforcement learning for heuristic learning, modeling the feature selection problem as an MDP.

Building on traditional ACO approaches, [34] presented Ant-TD, a multi-label feature selection method that integrates ACO with Temporal Difference (TD) reinforcement learning. Unlike static heuristic functions used in traditional ACO-based methods, Ant-TD employs a dynamic heuristic learning approach. By modeling the feature selection problem as a Markovian Decision Process (MDP), where features are treated as states and selection actions as actions, Ant-TD adapts the heuristic function based on past experiences, improving the selection process through reinforced learning signals. Several challenges and open research questions in ACO-based feature selection were highlighted by [36]. Issues such as scalability, parameter tuning, and algorithmic complexity were discussed, along with potential future research directions like exploring hybrid approaches and integrating ACO with deep learning techniques. Further extending the ensemble approach, [37] introduced a method for ensemble feature selection using ACO. This approach integrates multiple heuristics through a multi-criteria decision-making framework, enhancing the effectiveness and robustness of the feature selection process by utilizing the collective wisdom of diverse heuristics.

The key modifications proposed in the paper [38] focus on the solution construction phase of ACS. Three direct methods involve pairing ants and synchronizing their path selection, while two other methods adjust ants' decisions using opposite-pheromone content. Additionally, an extension to the update phase was introduced, which involves additional pheromone updates based on opposite decisions. Other advancements in computational intelligence techniques have also been explored. For instance, [40] introduced MOBL, a multi-operator and neighborhood structure-based approach for numerical optimization, demonstrating enhanced exploration and exploitation capabilities. Similarly, [41] proposed Orthogonal OBL (OOBL), inspired by the concept of opposition in nature, to improve the optimization process by considering both original and orthogonal states of solutions. Incorporating OBL and K-means clustering techniques, [42] presented an enhanced version of the Elephant Herding Optimization algorithm. These enhancements aim to improve exploration and exploitation capabilities, leading to better convergence and solution quality in solving numerical optimization problems. Additionally, [43] introduced Opposed Pheromone ACO (OPACO), which applies ACO principles to identify properties of nonlinear structures accurately. This algorithm offers potential applications in various engineering domains, contributing significantly to optimization for structural analysis and design.

Despite these advancements, there has been no combination of ACO, TD, and OBL observed in existing literature. In this article, we introduce an innovative approach that concurrently utilizes multi-stage and heuristic opposition along with reinforcement learning, aiming to advance the state-of-the-art in multi-label feature selection.

3. Basic concepts

3.1 Ant Colony Optimization

ACO is a metaheuristic algorithm inspired by the foraging behavior of ants, commonly used to solve combinatorial optimization problems like the Traveling Salesman Problem (TSP), the Vehicle Routing Problem (VRP), and the Quadratic Assignment Problem (QAP) [44]. The process begins by initializing a population of artificial ants, which are placed either randomly or at predefined locations in the search space. Each ant constructs a solution by iteratively selecting components based on a probabilistic rule [45]. For example, in the TSP, each ant constructs a tour by probabilistically selecting the next city to visit, influenced by pheromone trails and heuristic information [46]. After all ants have constructed their solutions, pheromone trails are updated based on the quality of these solutions. Typically, more pheromone is deposited on edges that are part of better solutions [47]. Ants that discover higher-quality solutions deposit more pheromone on the paths they take. To prevent stagnation and encourage exploration, pheromone trails undergo evaporation, controlled by an evaporation rate parameter that determines how quickly the pheromone dissipates [48]. Steps 2-4 are repeated for multiple iterations or until a termination criterion is met. When the predetermined number of iterations is reached, or the termination criteria are satisfied, the best solution found by the ants is selected as the final solution. This solution is then evaluated for quality and feasibility [49]. The algorithm concludes when a predefined stopping criterion, such as a maximum number of iterations without improvement or convergence to a satisfactory solution, is met [50].

In ACO, updating pheromone trails is a critical step that steers the search process towards promising solutions. The pheromone update rule ensures that better solutions contribute more to the pheromone trails, thus reinforcing the exploration of promising regions within the search space [51]. There are typically two main components in updating pheromone trails: global update and local update [52].

- **Global Pheromone Update**:

After all ants have constructed solutions in an iteration, the global update is performed to update the pheromone trails based on the quality of the solutions found. The amount of pheromone deposited on an edge is typically proportional to the quality of the solution that includes that edge. Better solutions contribute more pheromone to the edges they traverse [53]. The global pheromone update can be calculated as follows [54]:

$$\boldsymbol{\tau_{ij}^{(t+1)} = (1-\rho).\tau_{ij}^{(t)} + \sum_{k=1}^{N} \Delta\tau_{ij}^{k}} \tag{1}$$

where:

  - $\boldsymbol{\tau_{ij}^{(t)}}$ is the pheromone level on edge (*i*, *j*) at iteration *t*,
  - ρ is the evaporation rate, representing the amount of pheromone that evaporates from the trails,
  - $\boldsymbol{\Delta\tau_{ij}^{k}}$ is the amount of pheromone deposited on edge (*i*, *j*) by ant *k*, and
  - *N* is the total number of ants.
- **Local Pheromone Update**:

In addition to the global update, a local pheromone update is often applied to intensify the search near the edges that have been recently traversed by ants. The local update involves depositing a small amount of pheromone on the edges immediately after an ant traverses them, regardless of the solution's quality [55]. The local pheromone update can be calculated as follows [56]:

$$\boldsymbol{\tau_{ij}^{(t+1)} = (1-\alpha).\tau_{ij}^{(t)} + \alpha \cdot \tau_0} \tag{2}$$

where:

- α is the local update factor, determining the amount of pheromone deposited locally,
- $\boldsymbol{\tau_0}$ is a constant representing the initial pheromone level on edges, and
- $\boldsymbol{\tau_{ij}^{(t)}}$ is the pheromone level on edge (i, j) before the local update.

The balance between global and local updates, as well as the choice of parameters such as the evaporation rate and local update factor, play crucial roles in the convergence and performance of the ACO algorithm. These parameters are often tuned empirically to achieve optimal results for specific problem instances [57].

3.2 Opposition-based learning mechanism

OBL is a versatile concept that can be applied in various ways across different optimization algorithms and problem domains. While there isn't an exhaustive list of opposition learning operators, several common techniques have been developed [58]. Absolute Value Opposition Operator**,** Negation Opposition Operator**,** Reflection Opposition Operator**,** Inversion Opposition Operator**,** Centered Opposition Operator**,** Random Opposition Operator**,** Weighted Opposition Operator**,** Variable-Based Opposition Operator and Crossover-Based Opposition Operator are different kind of opposition [59]**.** OBL involves generating opposite solutions based on the current solution and then selecting the best among them. While there isn't a single formula that encapsulates the entire mechanism, this paper can provide a basic mathematical representation of how the opposite solution is generated [60].

The absolute value opposition (ABS) operator is a concept used in OBL to generate an opposite solution by taking the absolute value of each component of the current solution [61]. Mathematically, if we have a current solution represented as $\boldsymbol{X} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n})$ the ABS operator can be defined as follows [62]:

$$\boldsymbol{X'} = (|\boldsymbol{x_1}|, |\boldsymbol{x_2}|, \dots, |\boldsymbol{x_n}|) \tag{3}$$

where $\boldsymbol{X'}$ represents the opposite solution generated using the absolute value opposition operator.

Here's how it works:

1. For each component $\boldsymbol{x_i}$ of the current solution X, take the absolute value $|\boldsymbol{x_i}|$.
2. Assign the absolute value $|\boldsymbol{x_i}|$ to the corresponding component $\boldsymbol{x'_i}$ of the opposite solution $\boldsymbol{X'}$.

This operation essentially mirrors the values of the components across the origin, resulting in an opposite solution that explores the opposite direction in the search space [63].

3.3 Temporal Difference

TD learning is a key concept in RL that combines elements of dynamic programming and Monte Carlo methods. It is used for estimating the value function of a policy based on experience gathered through interactions with the environment [64]. In TD learning, an agent learns by updating its value estimates based on the temporal difference between successive estimates. This difference, often denoted as δ, represents the discrepancy between the predicted value and the actual observed value obtained from the environment [65]. The TD error δ is typically defined as the difference between the sum of the immediate reward and the estimated value of the next state subtracted by the current state's estimated value. Mathematically, it can be expressed as [66]:

$$\boldsymbol{\delta} = \boldsymbol{r} + \boldsymbol{\gamma V}(\boldsymbol{s'}) - \boldsymbol{V}(\boldsymbol{s}) \tag{4}$$

where:

- δ is the temporal difference error.
- r is the immediate reward obtained after transitioning from state s to state s'.
- γ is the discount factor, which determines the importance of future rewards.
- $\boldsymbol{V}(s)$ and $\boldsymbol{V}(s')$ are the estimated values of states s and s', respectively.

TD learning algorithms, such as TD(0) and Q-learning, utilize this TD error to update the value estimates iteratively. By updating value estimates based on the observed rewards and predictions, TD learning allows the agent to learn efficiently from its experiences without requiring complete knowledge of the environment's dynamics [67].

3.4 Mutual Information (MI)

MI is a measure of the amount of information that can be obtained about one random variable through another random variable. In the context of probability theory and statistics, it quantifies the amount of dependence between two variables [68]. The MI between two discrete random variables X and Y, denoted by $I(X; Y)$, can be expressed mathematically as:

$$\boldsymbol{I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) log(\frac{p(x,y)}{p(x).p(y)})} \tag{5}$$

where:

- $p(x, y)$ is the joint probability mass function of X and Y, representing the probability that $X= x$ and $Y=y$.
- $p(x)$ and $p(y)$ are the marginal probability mass functions of X and Y, respectively.
- The summations are taken over all possible values of X and Y [69].

3.5 Cosine similarity

The cosine similarity is a mathematical concept often used to determine the similarity between two vectors in high-dimensional space. Cosine similarity, especially in the field of data analysis, text mining, and recommendation systems, is popular. Given two vectors A and B in an n-dimensional space, their cosine similarity is calculated as the cosine of the angle between the two vectors [70]. Mathematically, it is defined as follows [71]:

$$\boldsymbol{Cosine(A,B) = \frac{A.B}{\| A \|.\| B \|}} \tag{6}$$

Where in this formula:

- $\cdot$ represents the dot product between two vectors.
- $\|\|$ represents the Euclidean norm (magnitude) of a vector. $A\cdot B$ measures the alignment between two vectors in space, while the denominator $\|A\|.\|B\|$ normalizes the result, turning cosine similarity into a metric.

3.6 Ant-TD

The proposed algorithm combines elements of ACO and TD reinforcement learning to improve the process of feature selection in multi-label datasets. The key innovation is using TD learning to dynamically update the heuristic function, specifically the state value V, which guides the ants in selecting the most informative features [72].

1. **Initialization of Pheromone Trails and State Values:**
    - Initialize pheromone trails for each feature to a small positive value.
    - Initialize state values $V(s)$ for each state (feature subset) to zero.
2. **Ant Movement and Solution Construction:**
    - Each ant starts with an empty feature subset.
    - Ants iteratively add features to their subset based on a probabilistic rule influenced by pheromone levels and the heuristic information provided by $V(s)$.
3. **Heuristic Information Using TD Learning:**
    - Define a state ss as the current subset of selected features.
    - Define actions as the selection of a feature to add to the subset.
    - Calculate the value $V(s)$ using TD learning, which estimates the expected future reward (performance improvement) of the current feature subset.
4. **Transition and Reward:** After selecting a feature and transitioning to a new state s', update the state value $V(s)$ using the TD update rule:

$$\boldsymbol{V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]} \tag{7}$$

    where α is the learning rate, γ is the discount factor, and r is the reward signal (e.g., improvement in classification performance) [73].

5. **Pheromone Update:** Update the pheromone levels based on the quality of the solutions constructed by the ants. This can involve:
    - Evaporation: Reducing pheromone levels to avoid premature convergence.
    - Deposition: Increasing pheromone levels on the paths (feature subsets) that lead to high-quality solutions.
6. **Termination**
    - **Termination Condition:** The algorithm iterates until a stopping criterion is met (e.g., a fixed number of iterations, convergence of pheromone levels, or no improvement in performance).
    - **Solution Extraction:** The best feature subset is selected based on the highest pheromone levels and the best $V(s)$ values, representing the most informative features for multi-label classification.

Using TD learning, the heuristic function $V(s)$ adapts based on the exploration process, providing a more accurate estimate of future rewards for feature subsets. OBL ensures a comprehensive exploration of the feature space by considering both original and opposite solutions. The combination of ACO, TD learning, and OBL leads to better feature selection, as evidenced by improved metrics in multi-label classification tasks.

4. Proposed method

This section provides a detailed analysis of the OBL and TD algorithms within ACO for feature selection in multi-label data. As discussed earlier, one effective method for multi-label feature selection involves employing ACO with a dynamic heuristic function. In this algorithm, optimal features are chosen by updating pheromone values, state values, and solutions. Various techniques, such as mutual correlation and cosine correlation, are utilized to calculate and update these values. The effectiveness of the Improved ACO algorithm heavily relies on accurately computing the correlation and relationship between the data matrix's pheromone values and state values. TD learning offers significant advantages, particularly its capability to update value estimates online after each time step, making it suitable for real-time applications and environments with continuous interaction. Moreover, TD learning typically achieves faster convergence compared to Monte Carlo methods by updating value estimates based on incomplete episodes, enabling more frequent learning updates.

In this article, three methods of the OBL Algorithm are integrated into the ACO framework. Initially, local updates incorporate opposite pheromones. The second method involves utilizing the opposite vector **V** during updates. Finally, both the opposite pheromone matrix and vector **V** are simultaneously employed in updates, facilitating a comprehensive investigation. In ACO, ants navigate a fully connected, undirected graph where nodes represent data features and edges denote relationships between these features. Notably, this approach does not involve learning algorithms, rendering it a filter approach. Ants are placed randomly on features, constructing solutions by visiting different features in the search space using a combination of greedy and probabilistic state transition rules.

4.1 Using opposition in pheromones

The process of updating pheromones is a foundational aspect of the ACO algorithm, crucial for guiding the accuracy of paths constructed based on these values. In this approach, each ant begins at a randomly chosen point on a graph and starts its journey by selecting a node. After completing its tour or path, each ant undergoes a local pheromone update. The amount of pheromone deposited on the edges of the ant's path is determined by the quality of the solution found, typically inversely proportional to the length or cost of the path, shorter paths receive more pheromone. This local update encourages subsequent ants to favor edges from shorter paths. To enhance the convergence speed and ensure that ant movement isn't purely random from the outset, the algorithm computes the (ph0) matrix. This *d*-dimensional matrix is based on MI between the dataset and class labels.

Opposite Pheromone per Node (OPN) introduces a method of utilizing opposition within ACO. This involves each ant's decision-making process when choosing a node to continue its path. If the comparison with a λ_0 indicates a smaller value, the ant follows the opposition path rather than the pheromone-calculated path. This opposition approach diverges from the anticipated paths typically chosen by ants based on equations and predicted paths. Both local and global updates of pheromones are influenced by this method, directing ant exploration towards paths less likely to be chosen.

Algorithm 1: OBL for pheromone update

```
function opposition-based initialization(data):
    Compute MI matrix (Eq 5)
    Find min and max values for central point calculation
    Compute opposition matrix (Eq 3)
    Calculate column-wise mean of opposition matrix
    Initialize pheromone vector (ph0)
  return ph0
```

4.2 Using opposition in state value V

Another influential factor in the ACO algorithm is the state value vector, denoted as V. This vector must be updated in parallel of pheromone for each ant to find the best solution. The **V** vector represents the value of each point in the graph for ant movement. By giving the initial value of TD(0) to the **V** vector, the ACO algorithm can be transformed into a RL state, for which the formulas (3) and (6) are used to obtain this initial state. Using opposition in this case means that the values of the points in the graph change for the opposite points, and thus, for local and final updates, opposition is used based on the comparison value of λ_0.

Algorithm 2: OBL for update **V**

```
function opposition-based initialization(data):
    Compute cosine similarity matrix (Eq 6)
    Find min and max values for central point calculation
    Compute opposition matrix (Eq 3)
    Calculate column-wise mean of opposition matrix
    Initialize state value vector (V0)
return V0
```

4.3 Using opposition in both pheromones and state value V

In this section, we address the simultaneous use of the oppositions introduced in Sections 1 and 2. In previous sections, when ants were moving in the presence of opposition, only one of the search factors was altered. For example, when using opposition pheromones, the state value vector still had the highest value for initial movement on the graph, and in the reverse part of the graph, it had a lower value. However, with simultaneous opposition use, if the value of its pheromone changes, the vector **V** also adjusts its sequence towards the other side of the graph, giving higher value. In this case, better solutions are likely to be explored, leading to more thorough examination of the graph points to reach the optimal solution.

Algorithm 3: Local Pheromone Update

```
Initialize pheromone trails and heuristic information
While termination condition is not met:
    For each ant:
        Initialize ant's path
        While ant does not complete its tour:
            If λ> λ0:
                Algorithm 1
                Algorithm 2
            Else
                Regular selection rule (Eq 2)
          Select the next city to visit based on pheromone levels and heuristic information
          Update ant's path and pheromone trail
        End While
        Update global best solution if necessary (Eq 1)
    End For
    Update pheromone trails based on ant paths and global best solution
End While
```

The Algorithm 1 and Algorithm 2 demonstrate the construction of opposition-based functions for pheromone and **V**, respectively. The Algorithm 3 presents the local update mechanism that employs both opposition-based pheromone and **V** concurrently. The Algorithm 4 illustrates the search procedure of the final proposed algorithm.

Algorithm 4: Ant-OBL-TD

Initialize pheromone trails τ and state values V
For each iteration do:
    For each ant do:
        Initialize empty feature subset S
        While not termination condition for ant do:
            Select next feature f based on τ and **V** using probabilistic rule (Algorithm 3)
            Add feature f to subset S
            Transition to new state S'
            Compute reward r for new state S'
            Update V(S) using TD learning: (Eq 7, 4)
        End While
        Evaluate the quality of subset S
        Evaluate V(s) and update V(0) and τ
    End For
    Update pheromone trails τ based on ant solutions
End For
Select the best feature subset based on τ and V

---

5. Experiments setup

5.1 Datasets

We have tested our hypothesis on 9 multi-label datasets. These datasets, which include birds, scene, art, science, chess, coffee, cooking, image, and language, cover various types of multi-label data. The language dataset is text-based, the image dataset consists of images, and the other datasets belong to different categories. All these multi-label datasets have been sourced from reference [74-75].

Table 1 Datasets and their characteristics.

Name	Features	Labels	Instances
Birds	258	21	645
Scene	294	6	2407
Arts	262	26	5000
Science	243	40	5000
Chess	812	227	1675
Coffee	1763	123	225
Cooking	977	400	10491
Image	294	5	2000
Language	1004	75	1459

5.2 Parameter settings

To predict the relevant labels using the implicit pattern among the data, we employed the well-known and widely used multi-label classifier, Multi-Label *K*-Nearest Neighbors (MLKNN) [76]. This classifier uses the *K*-nearest method for multi-label data, predicting the label set of an unseen instance based on information obtained from the neighborhood of the instance. Among the elements used in this algorithm are membership count statistics and the use of the Maximum A Posteriori (MAP) principle. In our case, we used a neighborhood size of K=10.

RL and ACO algorithms typically include parameters that need to be tuned. The values of these parameters usually converge to appropriate values based on various experiments, which are listed in Table (2) for the datasets used in this experiment. For the ACO algorithm, the number of ants is denoted by *nAnt*, the number of graph traversal cycles for each ant by *nCycle*, the number of features each ant can observe in each cycle by *NF*, and the pheromone evaporation rate per move by $\boldsymbol{\rho}$. In the heuristic function used and reinforcement learning, the learning rate is denoted by $\boldsymbol{\alpha}$, the discount rate by $\boldsymbol{\gamma}$, and the parameter $\boldsymbol{\beta}$. The value of final top features is indicated by m in the table. Additionally, for the OBL function, we used the value of $\boldsymbol{\lambda_0}$.

Table 2 Parameter settings

Parameter	Explanation	Value
α	The Learning rate of TD(0)	0.5
ρ	Pheromone decay rate	0.2
γ	The Discount rate of TD(0)	0.8
nCycle	Number of iteration that the algorithm should repeat	25
nAnt	Number of ants that search the features space	5
NF	Number of features each ant should traverse	$\frac{1}{8}d \leq NF \leq \frac{1}{6}m$
q	exploration-exploitation coefficient	0.7
β	The trade-off between heuristic information and pheromone	1
m	Number of top features that should be selected	$10 \leq m \leq 100$
λ_0	The OBL using rate	0.001

To compare the proposed algorithm with other feature selection methods in multi-label data, we utilized several evaluation metrics, including accuracy and hamming loss. These metrics are example-based and evaluate the performance of each test instance individually, then calculate the average value for the test set. Accuracy is calculated based on the set of correctly predicted labels among all predicted and actual label sets for each instance, according to the following relationship.

Accuracy is a measure of how close a value is to its true value. In various contexts, accuracy can be defined slightly differently, but it generally relates to the correctness of a result [77]. Example-based accuracy measures the average accuracy across all instances. For each instance, it is calculated as the proportion of correctly predicted labels to the total number of labels (both true and predicted) [78].

$$Accuracy = \frac{1}{n}\sum_{i=1}^{N} \frac{|\hat{Y}_i \cap Y_i|}{|\hat{Y}_i \cup Y_i|} \tag{9}$$

where:

- $|\hat{\boldsymbol{Y}}_{\boldsymbol{i}} \cap \boldsymbol{Y}_{\boldsymbol{i}}|$ is the number of correctly predicted labels for the i-th instance.

- $|\hat{Y}_i \cup \boldsymbol{Y_i}|$ is the total number of unique labels in both the predicted and true sets for the *i*-th instance.

Hamming loss computes the number of label mismatches (where the predicted label does not match the true label) for each label across all instances. It then normalizes this count by the total number of labels across all instances [79].

$$\boldsymbol{Hamming\ Loss} = \frac{\mathbf{1}}{\boldsymbol{N-L}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}\sum_{\boldsymbol{j=1}}^{\boldsymbol{L}} \mathbf{1}(\boldsymbol{x_{ij}} \neq \boldsymbol{y_{ij}}) \qquad (10)$$

where:

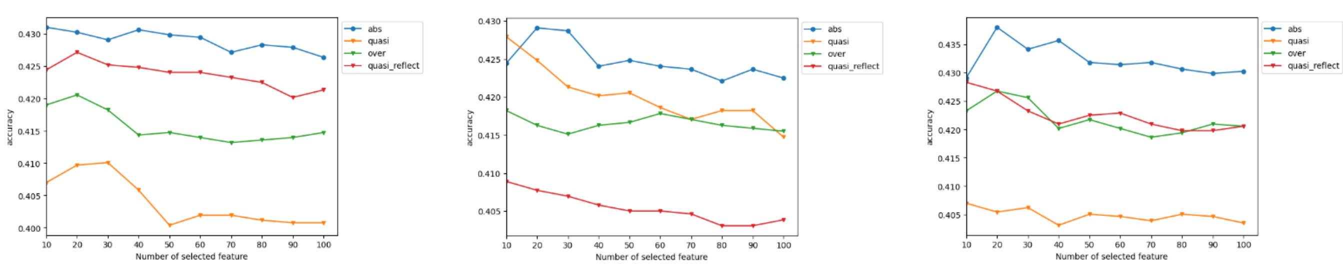
- *N* is the number of instances.
- *L* is the total number of labels.
- $\boldsymbol{x_{ij}}$ is the predicted label for the *j*-th label of the *i*-th instance.
- $\boldsymbol{y_{ij}}$ is the true label for the *j*-th label of the *i*-th instance.
- 1 is the indicator function, which equals 1 if $\boldsymbol{x_{ij}} \neq \boldsymbol{y_{ij}}$ and 0 otherwise.

6. Result

In this section, we present a comparative analysis of our proposed method against several established algorithms for feature selection, including Cluster, PPT-MI, LRFS, MCLS, MDFS, MGFS, MSSL, and Ant-TD. For each method, the top-100 feature subsets were sequentially selected and evaluated to determine their classification performance.

6.1 Opposition Model Selection on Birds Dataset

As mentioned in Section 2, there are several methods available for calculating opposition. In this study, we compare four methods: quasi-reflect, quasi, over, and abs, and identify the optimal option for the "Birds" dataset based on the accuracy metric. Figure (1) depicts the accuracy of the proposed algorithms using these four different opposition models on sample data.



Opposition V   Opposition pheromone   Opposition **V** and pheromone

Figure 1 Performance of Different Opposition Methods on the Birds Dataset

Figure (1) demonstrates that calculating opposition using the absolute value method (abs) yields superior results in terms of the accuracy metric compared to other methods for all three proposed algorithms. Henceforth, whenever "opposition" is mentioned, it specifically refers to opposition calculated using the abs method.

6.2 Accuracy Comparison

Figure (2) presents a comparison of accuracy between our proposed algorithms and other established methods. The results clearly demonstrate the superior performance of our algorithms. This improvement can be attributed to the effective selection of ants, which leverage the opposition path for each movement. Points in the graph deemed less important due to either the ant's movement choices or the evaporation process were included in the feature selection cycle by the algorithm. If this new path results in the selection of relevant features, it is reinforced by pheromones; otherwise, it is excluded from the selection cycle by other ants. This leads to a more accurate feature selection process, thereby reducing Hamming loss.

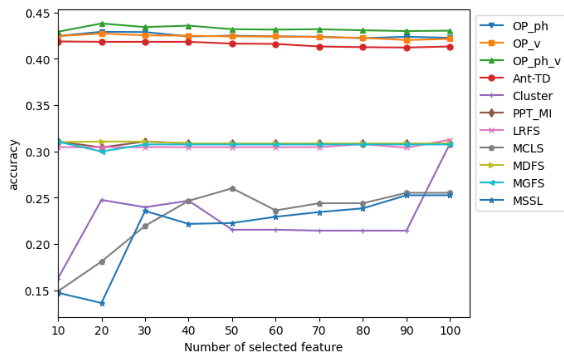
From now on, the following abbreviations will be used for chart and table explanations:

- **op-ph**: The improved ACO algorithm based on Reinforcement Learning and using opposition in pheromone update.
- **op-v**: The improved ACO algorithm based on Reinforcement Learning and using opposition in **V** update.
- **op-ph-v**: The improved ACO algorithm based on Reinforcement Learning and using opposition in both pheromone and **V** updates simultaneously.

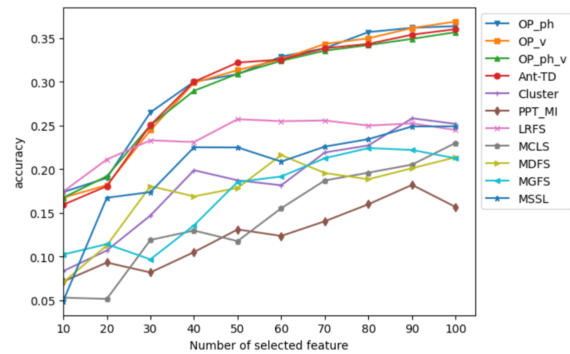
Table 3 Average classification performance in terms of Accuracy on 10 feature subsets (the higher the better).

	Op-ph-v	Op-ph	Op-v	Ant-TD	PPT-MI	MSSL	Cluster	MDFS	MGFS	LRFS	MCLS
Coffee	0.0477	**0.0514**	0.0485	0.0463	0.0033	0.0064	0.0057	0.0162	0.0001	0.0042	0.0207
Cooking	0.1395	**0.1500**	0.1399	0.1401	0.1155	0.1540	0.1186	0.1257	0.1380	0.0218	0.0389
Chess	0.1585	**0.1740**	0.1557	0.1496	0.0782	0.1068	0.0700	0.1064	0.0710	0.0701	0.0593
Arts	0.1592	0.1533	**0.1731**	0.1557	0.0415	0.1109	0.0244	0.1272	0.1534	0.0139	0.0520
Language	0.1707	0.1742	0.1598	0.1651	**0.2454**	0.0018	0.0243	0.0011	0.0411	0.0057	0.0182
Science	**0.0438**	0.0384	0.0414	0.0339	0.0113	0.0229	0.0111	0.0415	0.0433	0.0026	0.0110
Scene	0.4098	0.4200	**0.4413**	0.4019	0.0952	0.2863	0.1821	0.1842	0.2259	0.1984	0.1277
Birds	**0.4322**	0.4170	0.4236	0.4156	0.3085	0.2171	0.2279	0.3091	0.3069	0.3055	0.2290
Image	0.2915	**0.2986**	0.2954	0.2933	0.1246	0.2007	0.1861	0.1726	0.1698	0.2364	0.1445

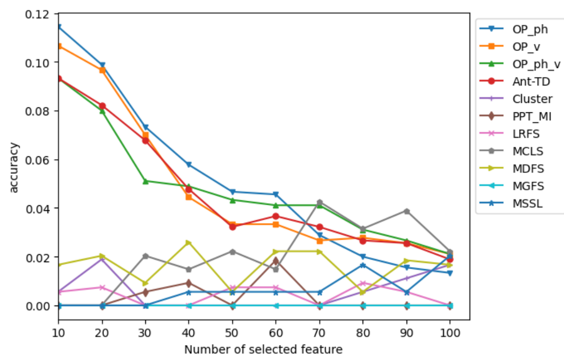
Table (3) displays the average accuracy of 10 groupings of selected features. The table clearly shows that the three proposed methods outperform the other methods presented. In some cases, using opposition in the pheromone update leads to optimal accuracy, while in other cases, using opposition in the vector **V** is more effective. There are also instances where the simultaneous use of both opposition methods yields the best results. Another notable observation from the table is that the three proposed methods exhibit similar accuracy, but they are consistently better than the other methods. Overall, it can be concluded that the combined use of both opposition techniques in the algorithm provides higher accuracy.



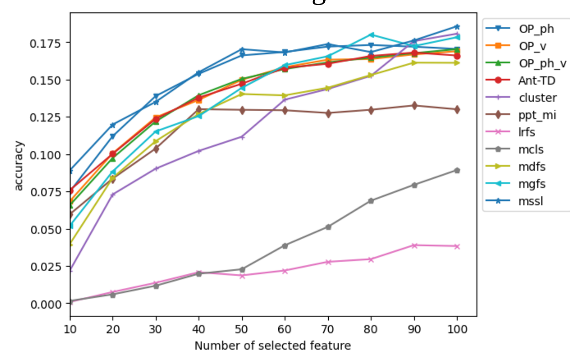
Birds



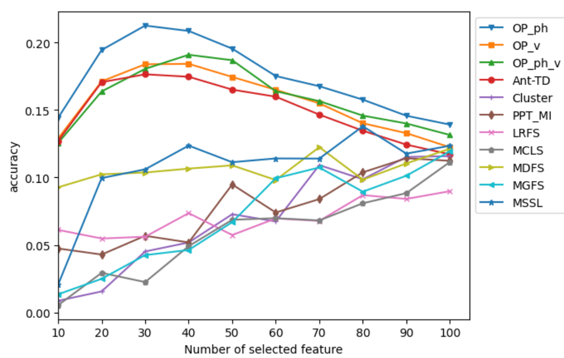
Image



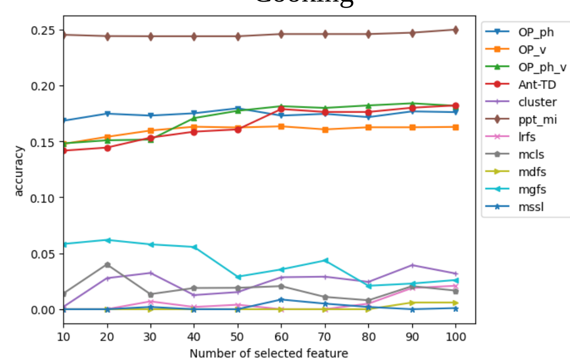
Coffee



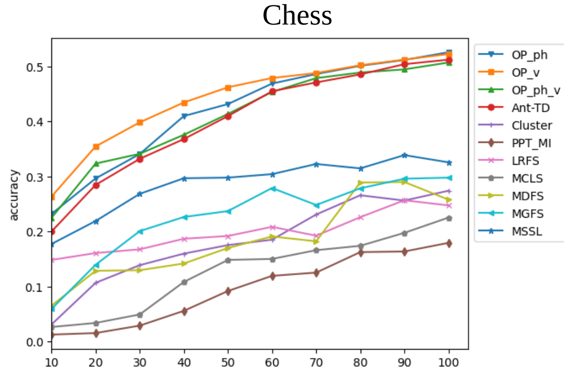
Cooking



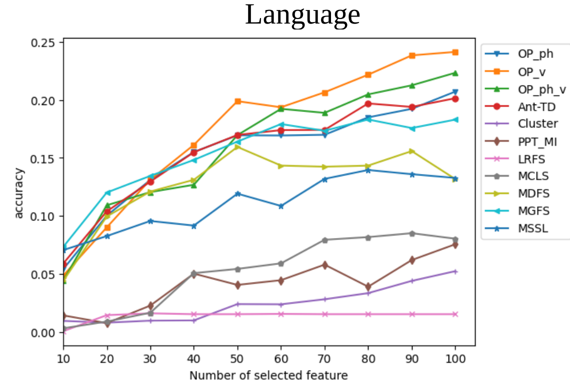
Chess



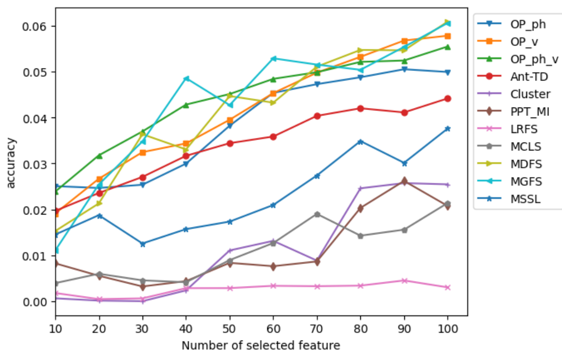
Language



Scene



Arts



Science

Figure 2 Average performance in terms of Accuracy.

Figure (2) illustrates the accuracy for selected features in groups of 10, 20, ..., up to 100. The figure compares nine different methods. As shown, the proposed algorithms demonstrate higher accuracy than the other methods. The charts reveal that the three proposed methods yield very similar results and consistently outperform the other nine methods in terms of accuracy. It can be concluded that the combined use of both opposition methods in the proposed algorithm delivers superior performance.

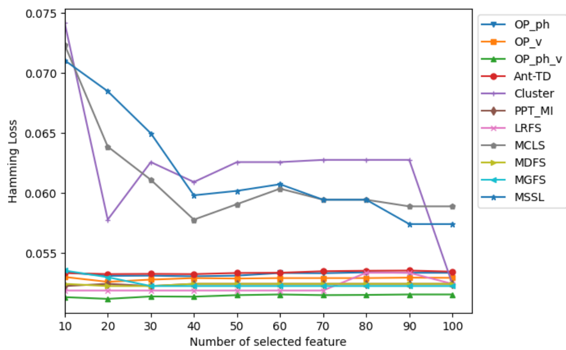
6.3 Hamming Loss Comparison

The chart in Figure (3) shows a comparison of Hamming loss across nine multi-label datasets using the methods introduced in this paper against other common feature selection methods. As illustrated, our proposed method consistently achieves lower Hamming loss compared to other methods, further confirming its superior accuracy.

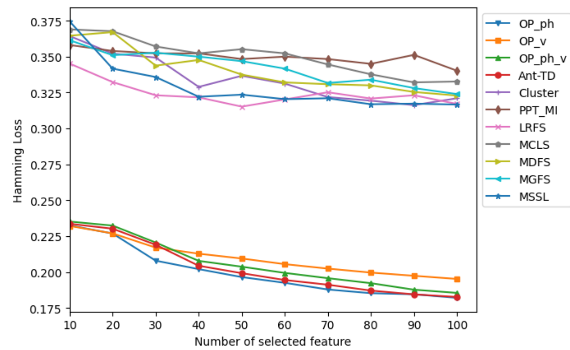
Table 4 Average classification performance in terms of hamming-loss 10 feature subsets (the smaller the better).

	Op-ph-v	Op-ph	Op-v	Ant-TD	PPT-MI	MSSL	Cluster	MDFS	MGFS	LRFS	MCLS
Coffee	0.0139	0.0146	**0.0138**	0.0142	0.0161	0.0163	0.0163	0.0163	0.0162	0.0163	0.0162
Cooking	0.0038	**0.0037**	0.0038	0.0038	0.0051	0.0051	0.0052	0.0051	0.0051	0.0055	0.0054
Chess	0.0070	**0.0069**	0.0071	0.0071	0.0101	0.0100	0.0103	0.0100	0.01026	0.0102	0.0103
Arts	0.0619	0.0624	0.0611	0.0621	0.0619	0.0594	0.0626	0.0589	**0.0577**	0.0622	0.0617
Language	0.0170	0.0174	**0.0154**	0.0155	0.0158	0.0158	0.0158	0.0158	0.0157	0.0158	0.0157
Science	**0.0355**	0.0361	0.0356	0.0361	0.0430	0.0429	0.0429	0.0424	0.0426	0.0431	0.0430
Scene	0.1300	0.1284	**0.1252**	0.1300	0.2845	0.2393	0.2645	0.2527	0.2527	0.2590	0.2803
Birds	**0.0514**	0.0532	0.0528	0.0533	0.0523	0.0618	0.0621	0.0523	0.0524	0.0522	0.0611
Image	0.2060	**0.1998**	0.2098	0.2026	0.3499	0.3290	0.3341	0.3401	0.3421	0.3243	0.3500

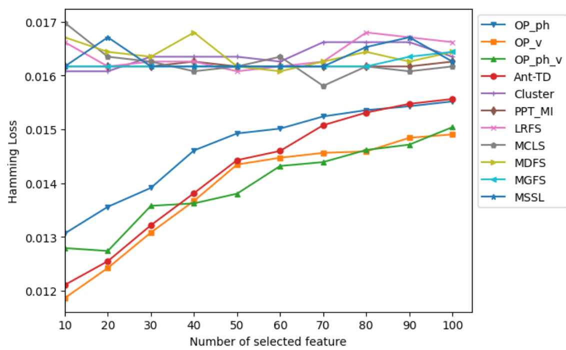
In Table (4), we evaluate the Hamming Loss criterion for nine datasets, averaging the results over ten groups. The table indicates that the Hamming Loss values for the three proposed algorithms are very close to each other. Furthermore, the proposed method consistently demonstrates the lowest Hamming Loss values in most cases compared to other methods introduced, highlighting its superior performance.



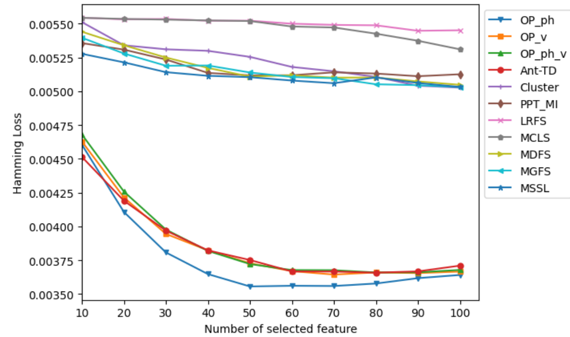
Birds



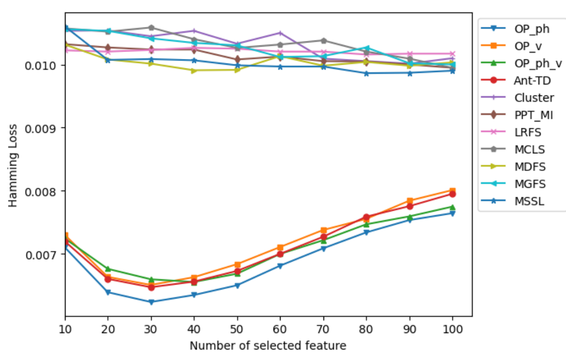
Image



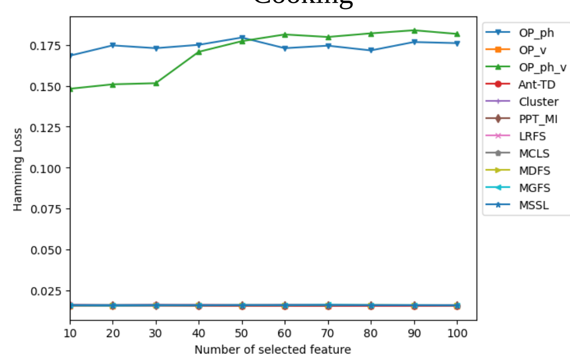
Coffee



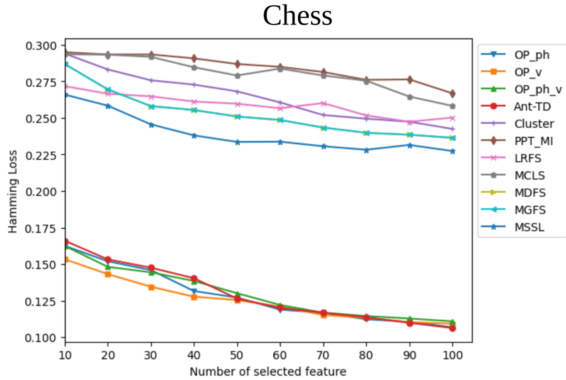
Cooking



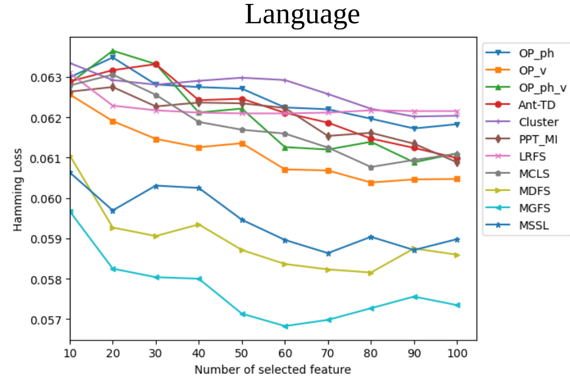
Chess



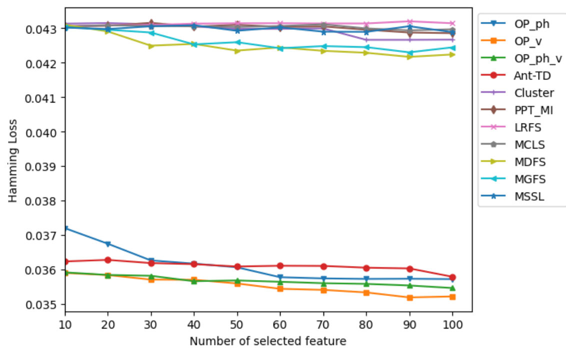
Language



Scene



Arts



Science

Figure 3 Average performance in terms of Hamming-loss

The chart in Figure (3) illustrates the comparison of groups of 10 features among the top 100 features of each dataset based on the Hamming Loss criterion. From the nine charts presented in the figure, it can be inferred that the three proposed methods in this paper demonstrate superior performance compared to other feature selection methods for multi-label data. Additionally, due to the closeness of the values in the three proposed methods, it can be concluded that the combined use of opposition in pheromone updating and the vector v, on average, serves as the optimal approach.

In the datasets **Chess** and **Coffee**, which have the highest number of labels among the introduced datasets, it is observed that as the number of samples analyzed increases, the algorithm becomes slower. This indicates the high complexity of these datasets due to the large number of labels, which require more time for processing and analysis. On the other hand, in the datasets **Arts**, **Scene**, **Birds**, **Science**, and **Image**, which have fewer labels, an increase in the number of selected features leads to an ascending trend in accuracy and a descending trend in Hamming loss. This demonstrates that the algorithm achieves better performance in terms of accuracy by selecting more appropriate features in these datasets.

For the **Language** dataset, where the number of labels falls between the two previous groups, the accuracy curve shows an upward trend, and the Hamming loss curve shows a downward trend. However, the rate of improvement in accuracy is slower compared to datasets with fewer labels. This indicates that as the number of labels increases, the rate of accuracy improvement decreases. Nevertheless, the algorithm still improves its performance by selecting the correct features, albeit with a lower slope compared to datasets with fewer labels.

This study developed an innovative algorithm integrating OBL and reinforcement learning into the ACO framework to enhance feature selection in multi-label data. The algorithm was tested through three proposed methods: using opposition to update pheromone levels, using opposition to update the heuristic value **V**, and applying opposition in both updating pheromones and values. Our experiments were conducted on nine diverse multi-label datasets, comparing the performance of our proposed methods against nine different established models for feature selection in multi-label data. The results, illustrated in two figures and two tables, clearly demonstrate that our proposed methods consistently outperform the existing models across all datasets. This consistent performance underscores the robustness and effectiveness of incorporating OBL and RL into the ACO framework.

The success of our methods can be attributed to the synergistic effect of opposition-based learning and reinforcement learning. Opposition-based learning provides a more comprehensive search space exploration, while reinforcement learning enhances the algorithm's ability to adapt and improve through iterative feedback. This combination results in a more efficient and accurate feature selection process, as evidenced by our experimental results. Furthermore, we analyzed the impact of different forms of opposition on the algorithm's performance. One of our figures presents this comparison, highlighting the optimal form of opposition for enhancing the algorithm's efficiency. This analysis emphasizes the importance of selecting the appropriate form of opposition to maximize the performance gains of the algorithm.

Based on the charts and tables related to Hamming loss and accuracy, all three proposed methods are the top performers in the comparison of feature selection in multi-label data. In other words, the ranks 1 to 3 are attributed to the methods proposed in the article. For some datasets, using pheromone opposition has improved performance, indicating that the reinforcement learning-based ACO algorithm had been stuck at an optimal point during pheromone updates, and incorporating pheromone opposition in the updates enhanced the algorithm's search capability. Similarly, for some datasets, using opposition vector **V** improved performance. Although reinforcement learning was employed for updating the vector **V** values, it had been trapped at an optimal point. By utilizing opposition during updates, the algorithm's exploration capability increased, resulting in better performance.

As observed in some datasets, simultaneous use of opposition in updates improved performance in feature selection for multi-label data. It can be concluded that if only one optimal algorithm is to be recommended for all datasets, the third proposed method—simultaneous use of opposition in updating both pheromones and vector **V**—would be the best choice.

Finally, the consistent success of our proposed methods across various datasets highlights their potential for broader application in multi-label data analysis. By leveraging the strengths of opposition-based learning and reinforcement learning, our algorithm not only improves feature selection accuracy but also introduces a novel approach to optimizing multi-label classification tasks. This work sets a new benchmark for feature selection in multi-label data, offering a promising direction for future research and application.

7. Conclusion

Integrating opposition learning into ACO offers several advantages. Firstly, it enhances solution diversity by considering opposite directions in the search space, potentially leading to a broader exploration. Secondly, opposition learning provides a mechanism for efficient exploitation of promising regions while simultaneously exploring new areas of the solution space. This can help prevent premature convergence and encourage the discovery of high-quality solutions. Additionally, opposition-based techniques can aid in escaping local optima by introducing perturbations to the current solutions. By leveraging both exploration and exploitation, opposition learning enriches the search process, promoting robustness and adaptability. Furthermore, opposition-based initialization offers a promising starting point for optimization algorithms, improving convergence speed and solution quality. Overall, incorporating opposition learning in ACO enhances its effectiveness, making it a powerful tool for solving complex optimization problems in various domains.

References

[1] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature Selection: A Data Perspective," *ACM Comput. Surv.*, vol. 50, no. 6, Article 94, pp. 1-45, Nov. 2018, doi: 10.1145/3136625.

[2] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16-28, 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[3] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 1–4, pp. 131-156, 1997, doi: 10.1016/S1088-467X(97)00008-5.

[4] A. Hashemi, M. B. Dowlatshahi, and H. Nezamabadi-pour, "An efficient Pareto-based feature selection algorithm for multi-label classification," *Information Sciences*, vol. 581, pp. 428-447, 2021, doi: 10.1016/j.ins.2021.09.052.

[5] A. Hashemi, M. B. Dowlatshahi, and H. Nezamabadi-pour, "VMFS: A VIKOR-based multi-target feature selection," *Expert Systems with Applications*, vol. 182, p. 115224, 2021, doi: 10.1016/j.eswa.2021.115224.

[6] M. Miri, M. B. Dowlatshahi, and A. Hashemi, "Feature selection for multi-label text data: An ensemble approach using geometric mean aggregation," in *Proc. 2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, Bam, Iran, 2022, pp. 1-6, doi: 10.1109/CFIS54774.2022.9756484.

[7] F. Beiranvand, V. Mehrdad, and M. B. Dowlatshahi, "Unsupervised feature selection for image classification: A bipartite matching-based principal component analysis approach," *Knowledge-Based Systems*, vol. 250, p. 109085, 2022.

[8] A. Hashemi, M. B. Dowlatshahi, and H. Nezamabadi-Pour, "A bipartite matching-based feature selection for multi-label learning," *Int. J. Mach. Learn. & Cyber.*, vol. 12, pp. 459–475, 2021, doi: 10.1007/s13042-020-01180-w.

[9] H. Noormohammadi and M. B. Dowlatshahi, "Feature Selection in Multi-label Classification based on Binary Quantum Gravitational Search Algorithm," in *Proc. 2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, Tehran, Iran, 2021, pp. 1-6, doi: 10.1109/CSICC52343.2021.9420617.

[10] N. Ghamrawi and A. McCallum, "Collective multi-label classification," in *Proc. 14th ACM international conference on Information and knowledge management (CIKM '05)*, New York, NY, USA, 2005, pp. 195–200, doi: 10.1145/1099554.1099591.

[11] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier Chains for Multi-label Classification," in *Machine Learning and Knowledge Discovery in Databases*, W. Buntine, M. Grobelnik, D. Mladenić, and J. Shawe-Taylor, Eds. Berlin, Heidelberg: Springer, 2009, pp. 254-269, doi: 10.1007/978-3-642-04174-7_17.

[12] J. Read, "Scalable Multi-label Classification," Ph.D. dissertation, Univ. Waikato, Hamilton, New Zealand, 2010. [Online]. Available: https://hdl.handle.net/10289/4645.

[13] S. M. Liu and J.-H. Chen, "A multi-label classification based approach for sentiment classification," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1083-1093, 2015, doi: 10.1016/j.eswa.2014.08.036.

[14] M. -L. Zhang and Z. -H. Zhou, "A Review on Multi-Label Learning Algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819-1837, Aug. 2014, doi: 10.1109/TKDE.2013.39.

[15] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining Multi-label Data," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Boston, MA: Springer, 2009, pp. 667-685, doi: 10.1007/978-0-387-09823-4_34.

[16] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[17] M. Dorigo and T. Stützle, "Ant Colony Optimization: Overview and Recent Advances," in *Handbook of Metaheuristics*, M. Gendreau and J.-Y. Potvin, Eds. Cham: Springer, 2019, pp. 311-351, doi: 10.1007/978-3-319-91086-4_10.

[18] D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, "Classification With Ant Colony Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 651-665, Oct. 2007, doi: 10.1109/TEVC.2006.890229.

[19] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, "Data mining with an ant colony optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 321-332, Aug. 2002, doi: 10.1109/TEVC.2002.802452.

[20] Y. Pei, W. Wang, and S. Zhang, "Basic Ant Colony Optimization," in *Proc. 2012 International Conference on Computer Science and Electronics Engineering (ICCSEE)*, Hangzhou, China, 2012, pp. 665-667, doi: 10.1109/ICCSEE.2012.178.

[21] K. M. Sim and W. H. Sun, "Ant colony optimization for routing and load-balancing: survey and new directions," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 33, no. 5, pp. 560-572, Sept. 2003, doi: 10.1109/TSMCA.2003.817391.

[22] H. R. Tizhoosh, "Opposition-Based Learning: A New Scheme for Machine Intelligence," in *Proc. International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, Vienna, Austria, 2005, pp. 695-701, doi: 10.1109/CIMCA.2005.1631345.

[23] A. B. Nasser, K. Z. Zamli, F. Hujainah, W. A. H. M. Ghanem, A.-M. H. Y. Saad, and N. A. M. Alduais, "An Adaptive Opposition-Based Learning Selection: The Case for Jaya Algorithm," *IEEE Access*, vol. 9, pp. 55581-55594, 2021, doi: 10.1109/ACCESS.2021.3055367.

[24] W. Wang, H. Wang, H. Sun, and S. Rahnamayan, "Using opposition-based learning to enhance differential evolution: A comparative study," in *Proc. 2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, Canada, 2016, pp. 71-77, doi: 10.1109/CEC.2016.7743780.

[25] A. R. Malisia and H. R. Tizhoosh, "Applying Opposition-Based Ideas to the Ant Colony System," in *Proc. 2007 IEEE Swarm Intelligence Symposium*, Honolulu, HI, USA, 2007, pp. 182-189, doi: 10.1109/SIS.2007.368044.

[26] R. Sihwail, K. Omar, K. A. Z. Ariffin, and M. Tubishat, "Improved Harris Hawks Optimization Using Elite Opposition-Based Learning and Novel Search Mechanism for Feature Selection," *IEEE Access*, vol. 8, pp. 121127-121145, 2020, doi: 10.1109/ACCESS.2020.3006473.

[27] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An Introduction to Deep Reinforcement Learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219-354, 2018, doi: 10.1561/2200000071.

[28] A. G. Barto, "Reinforcement Learning," in *Neural Systems for Control*, O. Omidvar and D. L. Elliott, Eds. Academic Press, 1997, pp. 7-30, doi: 10.1016/B978-012526430-3/50003-9.

[29] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: A Bradford Book, 2018.

[30] C. Szepesvári, *Algorithms for Reinforcement Learning*, 2nd ed. Springer Nature, 2022.

[31] E. Ernst, D. Damien, and A. Louette, "Introduction to reinforcement learning," in *Generative AI*, S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, Eds. *Business & Information Systems Engineering*, vol. 66, no. 1, pp. 111-126, 2024.

[32] M. Paniri, M.B. Dowlatshahi, and H. Nezamabadi-pour, "MLACO: A multi-label feature selection algorithm based on ant colony optimization," *Knowledge-Based Systems*, vol. 192, p. 105285, 2020, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2019.105285.

[33] A. Hashemi, M. B. Dowlatshahi, and H. Nezamabadi-pour, "MFS-MCDM: Multilabel feature selection using multi-criteria decision making," *Knowledge-Based Systems*, vol. 206, p. 106365, 2020.

[34] M. Paniri, M.B. Dowlatshahi, and H. Nezamabadi-pour, "Ant-TD: Ant colony optimization plus temporal difference reinforcement learning for multi-label feature selection," *Swarm and Evolutionary Computation*, vol. 64, p. 100892, 2021, ISSN 2210-6502, https://doi.org/10.1016/j.swevo.2021.100892.

[35] A. Hashemi, M.B. Dowlatshahi, and H. Nezamabadi-pour, "Ensemble of feature selection algorithms: a multi-criteria decision-making approach," *International Journal of Machine Learning and Cybernetics*, vol. 13, pp. 49–69, 2022, https://doi.org/10.1007/s13042-021-01347-z.

[36] A. Hashemi and M.B. Dowlatshahi, "Exploring Ant Colony Optimization for Feature Selection: A Comprehensive Review," in N. Dey (Ed.), *Applications of Ant Colony Optimization and its Variants*, Springer Tracts in Nature-Inspired Computing, Springer, Singapore, 2024, https://doi.org/10.1007/978-981-99-7227-2_3.

[37] A. Hashemi, M. Joodaki, N.Z. Joodaki, and M.B. Dowlatshahi, "Ant colony optimization equipped with an ensemble of heuristics through multi-criteria decision making: A case study in ensemble feature selection," *Applied Soft Computing*, vol. 124, p. 109046, 2022, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2022.109046.

[38] J. Li, L. Gao, and X. Li, "Multi-operator opposition-based learning with the neighborhood structure for numerical optimization problems and its applications," *Swarm and Evolutionary Computation*, vol. 84, p. 101457, 2024. [Online]. Available: https://doi.org/10.1016/j.swevo.2023.101457

[39] F. Karimi, M.B. Dowlatshahi, and A. Hashemi, "SemiACO: A semi-supervised feature selection based on ant colony optimization," *Expert Systems with Applications*, vol. 214, p. 119130, 2023, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2022.119130.

[40] P. Huang, Y. Zhou, W. Deng, H. Zhao, Q. Luo, and Y. Wei, "Orthogonal opposition-based learning honey badger algorithm with differential evolution for global optimization and engineering design problems," *Alexandria Engineering Journal*, vol. 91, pp. 348-367, 2024. [Online]. Available: https://doi.org/10.1016/j.aej.2024.02.024

[41] W. Li and G. G. Wang, "Improved elephant herding optimization using opposition-based learning and K-means clustering to solve numerical optimization problems," *J Ambient Intell Human Comput*, vol. 14, pp. 1753–1784, 2023. [Online]. Available: https://doi.org/10.1007/s12652-021-03391-7

[42] S. Farahmand-Tabar and S. Shirgir, "Opposed Pheromone Ant Colony Optimization for Property Identification of Nonlinear Structures," in *Applications of Ant Colony Optimization and its Variants*, Springer Tracts in Nature-Inspired Computing. Springer, Singapore, 2024. [Online]. Available: https://doi.org/10.1007/978-981-99-7227-2_5

[43] Alice R. Malisia and Hamid R. Tizhoosh, "Applying Opposition-Based Ideas to the Ant Colony System," presented at the 2007 IEEE Swarm Intelligence Symposium, Honolulu, HI, USA, Apr. 2007, pp. 182-189. http://doi.org/10.1109/SIS.2007.368044. Available: IEEE Xplore.

[44] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Proc. 1999 Congress on Evolutionary Computation-CEC99*, Washington, DC, USA, 1999, pp. 1470-1477 Vol. 2, doi: 10.1109/CEC.1999.782657.

[45] C. Blum, "Ant colony optimization: Introduction and recent trends," *Physics of Life Reviews*, vol. 2, no. 4, pp. 353-373, Dec. 2005, doi: 10.1016/j.plrev.2005.10.001.

[46] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," *Theoretical Computer Science*, vol. 344, no. 2–3, pp. 243-278, Nov. 2005, doi: 10.1016/j.tcs.2005.05.020.

[47] B. Chandra Mohan and R. Baskaran, "A survey: Ant Colony Optimization based recent research and implementation on several engineering domain," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4618-4627, 2012, doi: 10.1016/j.eswa.2011.09.076.

[48] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1155-1173, Mar. 2008, doi: 10.1016/j.ejor.2006.06.046.

[49] D. Merkle and M. Middendorf, "Modeling the Dynamics of Ant Colony Optimization," in *Evolutionary Computation*, vol. 10, no. 3, pp. 235-262, Sept. 2002, doi: 10.1162/106365602760234090.

[50] C. Blum and M. Dorigo, "The hyper-cube framework for ant colony optimization," in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 2, pp. 1161-1172, April 2004, doi: 10.1109/TSMCB.2003.821450.

[51] T. Stutzle and M. Dorigo, "A short convergence proof for a class of ant colony optimization algorithms," in *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 358-365, Aug. 2002, doi: 10.1109/TEVC.2002.802444.

[52] M. den Besten, T. Stützle, and M. Dorigo, "Ant Colony Optimization for the Total Weighted Tardiness Problem," in *Parallel Problem Solving from Nature PPSN VI. PPSN 2000. Lecture Notes in Computer Science*, vol. 1917. Springer, Berlin, Heidelberg, 2000, doi: 10.1007/3-540-45356-3_60.

[53] T. A. Runkler, "Ant colony optimization of clustering models," *Int. J. Intell. Syst.*, vol. 20, pp. 1233-1251, 2005, doi: 10.1002/int.20111.

[54] Q. Yang et al., "Adaptive Multimodal Continuous Ant Colony Optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 191-205, April 2017, doi: 10.1109/TEVC.2016.2591064.

[55] X. Liu, H. Yi, and Z. Ni, "Application of ant colony optimization algorithm in process planning optimization," *J. Intell. Manuf.*, vol. 24, pp. 1–13, 2013, doi: 10.1007/s10845-010-0407-2.

[56] C. Liu, L. Wu, W. Xiao, G. Li, D. Xu, J. Guo, and W. Li, "An improved heuristic mechanism ant colony optimization algorithm for solving path planning," *Knowledge-Based Systems*, vol. 271, 2023, 110540, ISSN 0950-7051, doi: 10.1016/j.knosys.2023.110540.

[57] A. Onan, "SRL-ACO: A text augmentation framework based on semantic role labeling and ant colony optimization," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 7, 2023, 101611, ISSN 1319-1578, doi: 10.1016/j.jksuci.2023.101611.

[58] M. Kelidari and J. Hamidzadeh, "Feature selection by using chaotic cuckoo optimization algorithm with levy flight, opposition-based learning and disruption operator," *Soft Comput.*, vol. 25, pp. 2911–2933, 2021, doi: 10.1007/s00500-020-05349-x.

[59] H. Liu, Z. Wu, H. Li, H. Wang, S. Rahnamayan, and C. Deng, "Rotation-Based Learning: A Novel Extension of Opposition-Based Learning," in *PRICAI 2014: Trends in Artificial Intelligence*, D. N. Pham and S. B. Park, Eds. Cham: Springer, 2014, pp. 365-376, doi: 10.1007/978-3-319-13560-1_41.

[60] M. Tubishat, N. Idris, L. Shuib, M. A. M. Abushariah, and S. Mirjalili, "Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection," *Expert Systems with Applications*, vol. 145, p. 113122, 2020, doi: 10.1016/j.eswa.2019.113122.

[61] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, "Enhancing particle swarm optimization using generalized opposition-based learning," *Information Sciences*, vol. 181, no. 20, pp. 4699-4714, Oct. 2011, doi: 10.1016/j.ins.2011.03.016.

[62] Q. Xu, L. Wang, N. Wang, X. Hei, and L. Zhao, "A review of opposition-based learning from 2005 to 2012," *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 1-12, Mar. 2014, doi: 10.1016/j.engappai.2013.12.004.

[63] S. Mohapatra and P. Mohapatra, "Fast random opposition-based learning Golden Jackal Optimization algorithm," *Knowledge-Based Systems*, vol. 275, 2023, 110679, ISSN 0950-7051, doi: 10.1016/j.knosys.2023.110679.

[64] M. A. Wiering and H. Van Hasselt, "Two novel on-policy reinforcement learning algorithms based on TD (λ)-methods," in *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, 2007, doi: 10.1109/ADPRL.2007.368199.

[65] P. Cichosz, "Truncating temporal differences: On the efficient implementation of TD (lambda) for reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 2, pp. 287-318, 1994.

[66] X. Xu, H.-g. He, and D. Hu, "Efficient reinforcement learning using recursive least-squares methods," *Journal of Artificial Intelligence Research*, vol. 16, pp. 259-292, 2002.

[67] K. Doya, "Reinforcement learning in continuous time and space," *Neural Computation*, vol. 12, no. 1, pp. 219-245, 2000, doi: 10.1162/089976600300015961.

[68] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E*, vol. 69, no. 6, 2004, 066138, doi: 10.1103/PhysRevE.69.066138.

[69] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: a survey," *IEEE Transactions on Medical Imaging*, vol. 22, no. 8, pp. 986-1004, 2003, doi: 10.1109/TMI.2003.815867.

[70] F. Rahutomo, T. Kitasuka, and M. Aritsugi, "Semantic cosine similarity," in *The 7th International Student Conference on Advanced Science and Technology (ICAST)*, South Korea: University of Seoul, 2012.

[71] P. Xia, L. Zhang, and F. Li, "Learning similarity with cosine similarity ensemble," *Information Sciences*, vol. 307, pp. 39-52, 2015, doi: 10.1016/j.ins.2015.02.016.

[72] L. M. Gambardella and M. Dorigo, "Ant-Q: A reinforcement learning approach to the traveling salesman problem," in *Machine Learning Proceedings 1995*, Morgan Kaufmann, 1995, pp. 252-260, doi: 10.1016/B978-1-55860-377-6.50039-3.

[73] Y. Liu and B. Cao, "A novel ant colony optimization algorithm with Levy flight," *IEEE Access*, vol. 8, pp. 67205-67213, 2020, doi: 10.1109/ACCESS.2020.2986932.

[74] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "Mulan: A Java Library for Multi-Label Learning," *Journal of Machine Learning Research*, vol. 12, pp. 2411-2414, 2011.G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," Data Min. Knowl. Discov. Handb., 2009, doi: 10.1007/978-0-387

[75] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," Data Min. Knowl. Discov. Handb., 2009, doi: 10.1007/978-0-387

[76] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038-2048, 2007, doi: 10.1016/j.patcog.2006.12.019.

[77] J. A. Swets, "Measuring the accuracy of diagnostic systems," *Science*, vol. 240, no. 4857, pp. 1285-1293, 1988, doi: 10.1126/science.3287615.

[78] P. Eusebi, "Diagnostic accuracy measures," *Cerebrovascular Diseases*, vol. 36, no. 4, pp. 267-272, 2013, doi: 10.1159/000353863.

[79] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov, "Hamming distance metric learning," in *Advances in Neural Information Processing Systems 25*, 2012.