

Research Article

Improved Double-Layer Structure Multilabel Classification Model via Optimal Sequence and Attention Mechanism

Geqiao Liu ^{1,2} and Mingjie Tan^{1,2}

¹College of Engineering and Technology, The Open University of Sichuan, Chengdu 610073, China

²Research Center for Educational Information Management and Information Systems, The Open University of China, Beijing 100039, China

Correspondence should be addressed to Geqiao Liu; liuqg@sartvu.net

Received 1 April 2022; Revised 29 July 2022; Accepted 14 September 2022; Published 22 November 2022

Academic Editor: Xiao Ling Wang

Copyright © 2022 Geqiao Liu and Mingjie Tan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multilabel classification is a key research topic in the machine learning field. In this study, the author put forward a two/two-layer chain classification algorithm with optimal sequence based on the attention mechanism. This algorithm is a classification model with a two-layer structure. By introducing an attention mechanism, this study analyzes the key attributes to achieve the goal of classification. To solve the problem of algorithm accuracy degradation caused by the order of classifiers, we adopt the OSS (optimal sequence selection) algorithm to find the optimal sequence of tags. The test results based on the actual dataset show that the ATDCC-OS algorithm has good performance on all performance evaluation metrics. The average accuracy of this algorithm is over 80%. The microaverage AUC performance reaches 0.96. In terms of coverage performance, its coverage performance is below 10%. The comprehensive result of single error performance is the best. The loss performance is about 0.03. The purpose of the ATDCC-OS algorithm proposed in the study is to help improve the accuracy of multilabel classification so as to obtain more effective data information.

1. Introduction

Multilabel classification, a commonly used method in big data analysis, aims to associate multiple labels to a sample at the same time. The ubiquity of multilabel data in real-life scenarios makes multilabel classification methods a popular research topic. However, in real-life applications, the integrity of the labels is usually not guaranteed. Due to poor data collection and the high cost of labeling and other reasons, only part of the labels in those samples is marked. There are many ambiguous examples in the real world. Sample instances are of a certain probability to be calibrated to different attributes. Many multilabel classification algorithms come into being. Usually, it is very challenging to extend the theory of single-label classification to multilabel classification. With the development of machine learning, multilabel classification algorithms can be applied to imaging, recommendation systems, medical diagnosis,

information retrieval, and many other fields [1–8]. In recent years, an ocean of research works accepted by top conferences (e.g., ACL, AACL, COLING, KDD, NIPS, ICDM, CIKM, INTERSPEECH, ICML, and IJCAI) proposed technologies and solutions for multilabel classification. The multilabel classification theory is a heated topic in data mining, which has attracted wide attention in the machine learning community.

There are two commonly used methods to construct a multilabel classification model: algorithm adaptation and problem transformation. The algorithm adaptation method is to adjust the existing algorithms (e.g., AdaBoost and decision trees) to solve multilabel classification issues. The performance of the algorithm adaptation method often remains poor. The problem transformation method splits a multilabel classification task into several single-label classification tasks. Then, the classical single-label classification theory is utilized to solve the problem, which brings the

trained single-label classifiers together as a super-classifier through linear combination. In this study, we investigate the multilabel classification theory algorithm based on problem transformation.

There are many existing problem transformation methods, such as the BR method [9], the CC theory [10], the MBR model [11], and the DLMC-OS algorithm [12]. However, these methods usually ignore the correlation among labels, the randomness of label sequences, and the redundant interactive label information, which reduces the accuracy of classification. The problem transformation method uses extended attributes to dig out the correlation between labels, but for different classification tasks, the importance of feature attributes is usually ignored during the process, which decreases the sensitivity of the classifiers. Therefore, we try to introduce an attention mechanism into the methods. Such attention mechanism method [13] is a bionic process based on how the human brain works. It is widely used in machine learning in areas such as speech recognition, image recognition, natural language processing, and so on. The attention mechanism usually calculates the probability mapping from an input to different outputs. The result with the largest probability will be chosen as the output, which has a great impact on considering the correlation between multiple attributes and labels. Then, we propose an attention mechanism-based multilabel classification algorithm, based on the double-layer chain structure.

In the proposed algorithm (algorithm of two/double-layer chain classification with optimal sequence based on attention mechanism, ATDCC-OS), we integrate three multilabel classification frameworks (including BR, MBR, and CC) and an attention mechanism into a chain structure with two layers. This structure exploits a binary association classification framework. In layer one, it carries out the initial classification. In layer two, the chain-based classifier utilizes an updating process to complete the final classification, which interacts with the label information coming from the output of layer one. In particular, we put an attention mechanism in layer two and use the output of layer one to calculate the probability of final classification results. Thus, this can find important information between different attributes and can improve the final classifier accuracy for different tasks. However, there is a random chain order problem in ATDCC-OS. We leverage the optimal sequence selection (OSS) algorithm to solve this issue. OSS integrates several variables and methods (including the hierarchical traversal algorithm, PageRank, Kruskal's algorithm, and mutual information) to decide labels' priority. Then, the priority rank is used to help ATDCC-OS to assign classifiers and construct the chain classification model.

In this study, the main contributions are as follows: (1) A double-layer structure multilabel classification model is constructed to fully integrate the advantages of three classical classification models. At the same time, an attention mechanism is introduced to further analyze the influence of key attributes on classification results to optimize traditional classification. (2) The OSS algorithm is proposed to solve the problem of low classification accuracy due to the existence of random chain order in the chain classification model. It is

applied to improve the second layer of the chained classification model. This classification model does not depend on any classification algorithm separately. Experiments on benchmark datasets validate the effectiveness of the proposed approach by comparing it with the state-of-the-art methods in terms of predictive performance.

The rest of this study includes the following: Section 2 deals with related work. Section 3 displays the proposed ATDCC-OS method. Then, we introduce the datasets used in the experiments and perform some simulations to verify the proposed method and discuss the experimental results in Section 4. We conclude our work in Section 5.

2. Related Work

2.1. Multilabel Classification Method. The multilabel classification approach has received much attention and is widely used in various fields, including text classification, scene and video classification, and bioinformatics. The multilabel classification includes two common methods: problem transformation process and algorithm adaptation process. The former changes a multilabel problem into one or several single-label issues [11] and uses basic classification algorithms, such as Naive Bayesian, supporting vector machine [14], k-nearest neighbor algorithm, and so on to solve them. The latter transforms the existing algorithms so that they can solve the multiclassification problem, e.g., ML-RBF method [15, 16], ML-kNN approach [17, 18], rank-SVM classification [9], and associated classification algorithm [19, 20].

BR (binary relevance) [9] is a common method of problem transformation, which transforms the multilabel classification issue into several binary relevance problems where it trains a binary classification model one by one for all labels. However, BR is often overlooked because it cannot effectively use the correlation between labels. The MBR based on BR was proposed [11], which was constructed as a two-layer model. Layer one in MBR is taken as the input of layer two as a sample attribute to consider label correlation. However, the problem of the label value redundancy is ignored in the training process of layer two.

The CC method was proposed by the authors in [10], where the chain is exploited to build the correlation among all labels. It converts all classifiers into the linear stochastic data chain and adds previous classifiers' output to the data sample attribute set and takes it as the input to the next classifier. However, there are many disadvantages to the random chain. First, in the CC training process, the classifier output is input as a new attribute together with the original attribute into the next classifier. So, the former classifiers in the chain have a greater impact on classification than the latter classifiers. The order of classifiers in the chain affects the classification result. Second, CC considers the correlation of attributes, but two linked classifiers can use the correlation between adjacent attributes, and the other correlation between attributes cannot be used. Finally, the order of classifiers in the chain is randomly assigned, so the CC model is not unique, which makes the model have strong randomness and ruins the stability of the algorithm [21, 22].

The two-layer classification model, DLMC-OS, is proposed to solve the classification problem [12]. In this model, the output of the first-level classifier is forwarded to the second-level classifier as an extended feature. Each classifier in the second layer of the model passes the latest classification results backward through a chain to consider the correlation between labels. This approach suppresses the classifier chain randomness, but it cannot obtain the unique classifier sequence in the chain.

2.2. Attention Mechanism. The attention mechanism method [23–25] is derived from the study of human vision, which simulates the perspective interest of human vision when observing. When the human eye scans the global image, the part of the information that assists the judgment is tracked dynamically in the image, and irrelevant information is ignored. This process can effectively decrease or reduce the amount of information processing when the eyes recognize images by paying more attention to part information. The modern attention mechanism is adopted for machine translation, and it greatly improves the performance of the model [26]. In 2014, Google Brain published an article on the attention mechanism [27]. The article pointed out that when viewing an image, people do not first look at the image pixels but pay more attention to the image’s specific parts based on their requirements. In addition, as humans, we will focus on the required attention locations in the future based on previous observations of images. The authors designed a new architecture named transformer. In a transformer, the self-attention mechanisms are extensively utilized to perform text representations [28], which break away from the traditional RNN/CNN. In recent years, transformer-style models achieved many good results in various tasks. Subsequently, attention mechanisms have become more common and are widely used in classification tasks, such as sentiment classification [29], musical instrument recognition [30], visual recommender systems [31], multilabel text classification [32], and multiple protein subcellular location prediction [33].

3. ATDCC-OS

3.1. Preliminaries. We set $\chi \in \mathfrak{R}^d$ and $Y \in R^L$ as the input domain and output domain, respectively. There are d -dimensional attributes in the input domain and L -dimensional labels in the output domain. Instance x belongs to a subset of attributes. We use the set L vector $x_i \in \chi$ to represent that x is the input and y is the output. If the label j is related to x , then $y_j = 1$, or $y_j = 0$. The set $D = \{(x_i, Y_i) | 1 \leq i \leq m\}$ represents the trained multilabel classification model, where $x_i \in \chi$ is an attribute vector $(x_{i1}, x_{i2}, \dots, x_{id})^T$ with d dimensions and $Y_i \subset \gamma$ indicates a label set corresponding to x_i . To construct a multilabel classifier, we let $H: \chi \rightarrow 2^\gamma$. $H^f = (H_1^f, H_2^f, \dots, H_L^f)$ and $H^s = (H_1^s, H_2^s, \dots, H_L^s)$ as the first and second layers of the multilabel classifier, respectively. $c^f = (y_1^f, y_2^f, \dots, y_L^f)$ and $c^s = (y_1^s, y_2^s, \dots, y_L^s)$ are the outputs of the first and second layers.

3.2. The ATDCC Framework. By referring to algorithm DLMC-OS, we construct the double-layer chain classification based on the attention mechanism (ATDCC). ATDCC converts the multilabel classification issue into a series of binary classification issues, each one of which is independent of others. In layer one, ATDCC performs binary transformation on labels and constructs some classifiers between attributes and labels. After training, the classifiers of each binary classification model can be obtained [12]. ATDCC completes the binary classification of instances in layer one and then makes the classification results as the extended attributes transfer to layer two. In layer two, ATDCC constructs a classification method with a chain structure by realizing the updating process of dynamic feedback. It exploits the classifier chain to transfer and change the labels. It realizes the interaction among labels and optimizes the classification result. ATDCC utilizes correlation among all labels for multilabel classification through label information interaction within layers and labels information transfer between layers.

3.2.1. ATDCC^{First-Layer}. ATDCC^{First-layer} follows the idea of a binary correlation classification model. It constructs a classifier with a binary structure for all labels. These binary classifiers are combined as classification one, as shown in Figure 1.

In step one, assume there is an annotated dataset with a size being L . ATDCC^{First-layer} constructs an attribute set for all labels by using the following equation:

$$D_{y_k}^f = \{(x_i, y_k) | 1 \leq i \leq m\},$$

$$\text{where } y_k = \begin{cases} 1, & \text{if } y_k \in Y_k, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

In step two, some binary algorithms B (such as SMO) are utilized to create the binary classifier of the training instance: $H_{y_k}^f \leftarrow B(D_{y_k}^f)$.

In step three, we use the obtained binary classifier to classify and predict the unseen instance X .

$$H_{y_k}^f: X \times \{0, 1\}^{L-1} \rightarrow \{0, 1\},$$

$$y_k^f = \{H_{y_k}^f(X) | X \in X \times (0, 1)^{L-1}, \quad 1 \leq k \leq L\}. \quad (2)$$

Finally, the prediction result of each classifier (i.e., $c^f = (y_1^f, y_2^f, \dots, y_L^f)$, as shown in Figure 1(b)) is the output of the unseen instance in the first layer of ATDCC, integrating these output c^f with the attribute set of samples to build a new attribute set $x' = \{(x_i, c^f) | 1 \leq i \leq m\}$. Let x' be the input of layer two in ATDCC.

3.2.2. ATDCC^{AT-Layer}. The attention mechanism is usually exploited in sequence-to-sequence learning paradigms. For different multilabel classification tasks, the attribute mapping weights between the two layers of ATDCC are different. The attention mechanism method can capture the weight value of all attributes in samples according to requirements. It can improve the final accuracy of classification results.

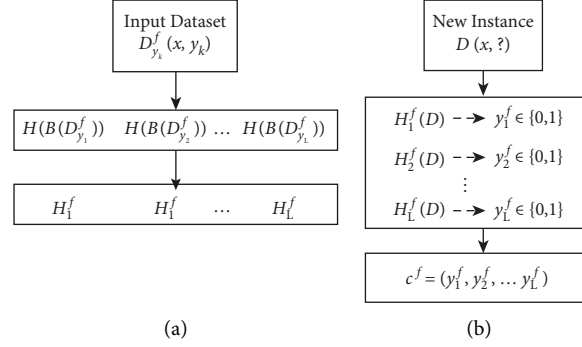


FIGURE 1: The first layer of ATDCC: (a) the training procedure of ATDCC^{First-layer} and (b) the test procedure of ATDCC^{First-layer}.

ATDCC^{AT-layer} uses the attention mechanism mentioned above to dynamically compute the extended attributes' weights. The layer two model can adapt to the requirement of the current classification task by adjusting the weight value of the transfer attributes between the two layers in ATDCC.

In step one, according to the original sample attributes' dimension of the first layer to define weight matrix W , the tanh function is exploited to train ATDCC^{AT-layer} to capture correlations between input attributes and label i . The trained model can be expressed as

$$e_{ij} = \tanh(W_{ij}x_{ij}' + b), \quad (1 \leq i \leq m), \quad (3)$$

where W and b , respectively, denote the weight matrix and the model's bias.

In step two, ATDCC^{AT-layer} uses a softmax function to transform the output of equation (3) to a probability value and then obtains the weight value of the attention scores.

$$W_{ij}' = \text{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{m=1}^m \exp(e_{ij})}. \quad (4)$$

Finally, the extended attribute set is weighted based on the attention weights obtained from equation (4):

$$x'' = \sum_{m=1}^m x_{ij}' w_{ij}'. \quad (5)$$

The parameters in our model are optimized by carrying out the minimization of the feedback result of the loss function. The cross-entropy loss in equation (6) is used as the loss function. The following equation calculates the accumulated loss derived from actual and predicted labels for each instance:

$$J(\theta) = -\frac{1}{l} \sum_t^{k=1} \log p(y_k | y_k'). \quad (6)$$

3.2.3. *ATDCC^{Second-Layer}*. ATDCC^{Second-layer} is the second layer of the ATDCC model (Figure 2), which uses the classification structure with a chain and exploits an updating process to classify instances in the second time. The attribute set of each binary model expands the correlation of the classification labels before the instance to create the chain structure of classifiers. The attribute set of all binary models is augmented via the 0/1 label estimation value obtained in layer one as well as the whole prior binary correlation estimations from layer two. In the second layer, the correlations between each label are fully applied. Given the attribute set, each classifier in the chain will learn and predict the binary association of labels.

In step one, ATDCC^{Second-layer} creates the extended attribute vector $D_{y_k}^s$ ($1 \leq k \leq L$) for each class label as shown in the following equation.

$$D_{y_k}^s = \{([w_i x_i, w_{i+1} y_1^f, \dots, w_{i+k-1} y_{k-1}^f, w_{i+k+1} y_{k+1}^f, \dots, w_{i+L} y_L^f], y_k^f) | 1 \leq i \leq m\}, \quad (7)$$

where W represents the set of attributes' weight value.

In step two, we use binary approach B (such as SM) to learn the constructed extended attribute vector (O) to create the binary classifier, $H_{y_k}^s \leftarrow B(D_{y_k}^s)$.

In the third step, use the constructed binary classifier to classify and predict the unseen instance X .

$$D_{y_k}^s = \{([w_i x_i, w_{i+1} y_1^f, \dots, w_{i+k-1} y_{k-1}^f, w_{i+k+1} y_{k+1}^f, \dots, w_{i+L} y_L^f], y_k^f) | 1 \leq i \leq m\}. \quad (8)$$

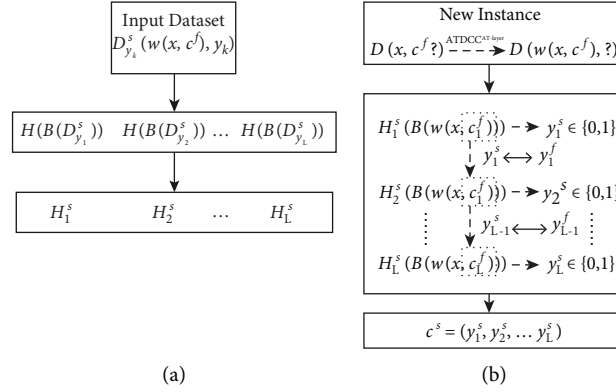


FIGURE 2: The second layer of ATDCC: (a) the training procedure of ATDCC^{Second-layer} and (b) the testing procedure of ATDCC^{Second-layer}.

In the model training process, we use the latest predicted label value to change each sample attribute set's label value. For example, for the third classifier H_3^s in a chain, the next sample's attribute variable is $[x, y_1^s, y_2^s, y_3^s, y_4^s, \dots, y_L^s]$ instead of $[x, y_1^f, y_2^f, y_3^f, y_4^f, \dots, y_L^f]$.

Finally, ATDCC evaluates the classification prediction result $c^s = (y_1^s, y_2^s, \dots, y_L^s)$ of each classifier as the final classification for the unseen instance.

3.3. OSS Method. In the MBR model, the sequence of the classifiers in the chain is randomly arranged. If the classification accuracy of the classifier at the core of this chain is very low, an error will be propagated via a backward way along this classifier chain, decreasing the classifier's accuracy. This further leads to lower classification correctness and accuracy for the whole chain. As the number of labels increases, the randomness of the OSS classifier chain also increases rapidly. The algorithm DLMC-OS can reduce the classifier chain's randomness, but the optimal label recognition sequence cannot be determined due to the non-uniqueness of the root node. The most effective method is to sort the sequence of the chain. The sequence of the classifier needs to be ranked according to attributes and the characteristics of the chain classification model. For this reason, the following constraints are proposed to search for the optimal chain sequence:

- (A) The label list is ordered according to a sequence which contains all label information
- (B) The label sequence satisfies the greatest correlation of labels
- (C) The label list sequence is optimal under current conditions

Under these design rules, we propose OSS in the model, which integrates mutual information and PageRank with the Kruskal algorithm and the hierarchical traversal method to find an optimal label sequence. The chain classification model uses sequences as the rules to assign the order of each classifier, and the second layer will optimize the ATDCC with the OSS algorithm.

3.3.1. Subalgorithm Related with OSS

- (1) *Mutual Information (MI) Theory.* In the information theory and probability theory, mutual information (MI) is used to evaluate the interdependence between two random variables, so we can obtain the "information amount" of a stochastic variable by observing the other random variables. Equation (9) shows the MI of the two variables. In current information technologies, the probability theory and information theory have been widely used. The MI theory is widely exploited in research works. In the machine learning field, MI can be utilized to select the features [34, 35]. The search engine often uses MI among phrases and contexts to find discover semantic clusters [36]. In statistical mechanics, MI is usually used to solve mechanical problems together with Loschmidt's paradox [37, 38].

Based on the MI application, we evaluate the correlation between labels by capturing MI among labels. Then, we exploit it as edges' weight in the fully connected graph.

- (2) *PageRank.* PageRank (PR) is used to overcome the page ranking issue in the detailed link analysis process, which was proposed in reference [39]. The key idea of this algorithm includes that the page's importance is related to the number as well as the detailed quality of another page that points to this page. This algorithm is applied in Google's search engine [40]. The importance of a Webpage can be quantified by the number of links in the link structure, rather than relying on specific search requests. Twitter uses a personalized PageRank to show users' another account [41]. In this study, we use PageRank and priority search to build the customized PageRank algorithm to decide a very important label to act as the chain's first node. This can overcome the issue of nonuniqueness of the chain head.
- (3) *Edge Weight-Based Graph Algorithm.* Usually, the connection between different entities can be

formulated as a graph with edge weights [42]. The weight of an edge may represent cost, length, or capacity, depending on the current problem to be solved [43–46]. In the model, we exploit this weighted graph method to create the graph with a fully connected relationship related to the labels in Algorithm 1.

- (4) *Detailed Kruskal’s Algorithm Idea.* In this study, the referred Kruskal’s algorithm is utilized to seek a tree with minimum spanning [47]. We use Kruskal’s method to seek a tree with the largest label spanning. This can provide a basis to create a sequence in which the association with labels is the largest. The designed algorithm is shown in Algorithm 2.
- (5) *Breadth-First Based Search Method.* In this study, the breadth-first based search (BFS) is an algorithm used for seeking the available paths of the graph, which traverses or searches the tree or graph data structures. Then, we use PageRank to find the starting point and use BFS to traverse the spanning tree with the maximum label to construct the resulting label order, as shown in Algorithm 3.

3.3.2. *The OSS’s Detailed Design Framework.* The detailed design steps for the OSS algorithm in this study are shown in Figure 3.

Step 1. Calculate the MI of the correlation between labels. Assuming that there are N labels y_1, y_2, \dots, y_n , we use formula (9) to calculate the MI on any two labels y_i and y_j , and the MI must be nonnegative.

Definition 1. The formula of MI calculation is

$$I(y_i, y_j) = \sum_{y_i, y_j} p(y_i, y_j) \log \left(\frac{p(y_i, y_j)}{p(y_i)p(y_j)} \right). \quad (9)$$

Step 2. Construct a fully connected graph G via labels, where the labels are the graph’s vertices, and MI volume among labels acts as edges’ weights. Utilize the Kruskal algorithm to build the label tree with the maximum weight. Then, invert the mutual information value to obtain the maximum weight spanning tree.

Step 3. Use the PageRank algorithm to sort each label in the dataset by “voting” and decide on the label node whose PR value is the highest. This node acts as the root node that belongs to a tree with the maximum weight. It is also selected to act as the first node of the traversal algorithm that is hierarchical. This can overcome the issue of not unique head label in the chain.

Step 4. Use Kruskal’s algorithm to generate a minimum weight label tree (MWT) used for the fully connected graph G . The label tree includes the whole labels and the entire

edges. These edges connect the label nodes. The weighted sum is the largest.

Step 5. Traverse the MWT with the label nodes obtained by BFS and PageRank to obtain the label sequence. Use this sequence as a guide for constructing the sequence of each classifier in a chain to overcome the uncertainty issue for the classification, as shown in Algorithm 4.

3.4. *The ATDCC-OS Framework.* The ATDCC-OS design framework is plotted in Figure 4. Figure 4 shows that the $ATDCC^{First-layer}$ and $ATDCC^{Second-layer}$ are the first and second layers. We utilize the OSS approach to optimize the chain structure in the ATDCC-OS framework. Then, we can seek an optimal sequence of labels. According to the best and optimal serials, we train each classifier in our model. We utilize this attention mechanism layer between layer one and layer two to find important attributes and features from the current task. In such a case, we can build a better classifier in layer two, as shown in Algorithms 5 and 6.

4. Experiments

To validate the method, we perform some simulations and use the experimental results to analyze the performance of the proposed algorithm. In the simulation, we analyze the algorithm (ATDCC-OS) presented in this study with other algorithms of multilabel classification (including DLMC-OS and BR and CC and MBR) via five metrics. We then take seven datasets as the multilabel benchmark.

4.1. *Test Datasets.* We utilize the standard datasets provided on the Mulan [48] platform as the multilabel benchmark. Table 1 describes each dataset and related statistical data in the simulation. N , F , and L represent instances’ numbers, attributes’ numbers contained in each instance, and labels’ numbers in the dataset, respectively. The notation label cardinality (LCard) represents the normal measure as shown in [49]. LCard denotes the average label number associated with an instance.

4.2. *Evaluation Methods.* The evaluation indicator is a measure that directs the indication of the algorithm’s performance. To better evaluate the method, we used mean accuracy, coverage rate, single error, ranking loss rate, and microaverage AUC to analyze the performance of ATDCC-OS.

- (1) Average precision: average precision [12] is an accurate metric, which associates recall with precision to sort search results. It reviews a mean score of labels with a higher rank than a specific tag. The larger the value of the average precision is, the better the classifier will be. The average precision can be expressed as

Input label values to construct a label map with weights:
 Input $Y = \{y_1, y_2, \dots, y_L\}$
 Output: G

- (1) $G \leftarrow \{\}$
- (2) $G.V \leftarrow Y$
- (3) For each (u, v) in $G.V$
- (4) Calculate the mutual information of $MI(u, v)$ according to Definition 1.
- (5) $G.E \leftarrow MI(u, v)$
- (6) $G \leftarrow G(V, E)$
- (7) Return G

ALGORITHM 1

Constructing the minimum spanning tree of labels based on Algorithm 1:
 Input: $G(V, E)$
 Output: MWT

- (1) $MWT \leftarrow \{\}$
- (2) For $v \in G$, then V is:
- (3) Make the set (v)
- (4) For (u, v) in $G.E$ is ordered according to weight (u, v) via an increasing way:
- (5) If $\text{set}(u) \neq \text{set}(v)$:
- (6) $MST = MST \cup \{(u, v)\}$
- (7) Let it Union (u, v)
- (8) Return $MWT \leftarrow MST$

ALGORITHM 2

Hierarchical traversal to get the label sequence:
 Input: MWT (V, E)
 Output: OS

- (1) Queue $Q \leftarrow \{\}$
- (2) For each $v \in MWT.V$:
- (3) $Q \leftarrow Q \cup (v)$
- (4) while $(Q! = \emptyset)$
- (5) $v \leftarrow Q.\text{head}, w \leftarrow Q.\text{next}$
- (6) while $(w! = \emptyset)$
- (7) $Q \leftarrow Q \cup (w)$
- (8) end while
- (9) end while
- (10) end for
- (11) $OS \leftarrow Q$
- (12) Return OS

ALGORITHM 3

$$\text{avgprec}_D(H) = \frac{1}{P} \sum_{i=1}^P \frac{1}{|Y_i|} \sum_{y \in Y_i} \frac{|\{y' | \text{rank}_C(x, y') \leq \text{rank}_C(x_i, y), y' \in Y_i\}|}{\text{rank}_C(x_i, y)}, \quad (10)$$

Where $\text{rank}(\cdot)$ is a sort function.

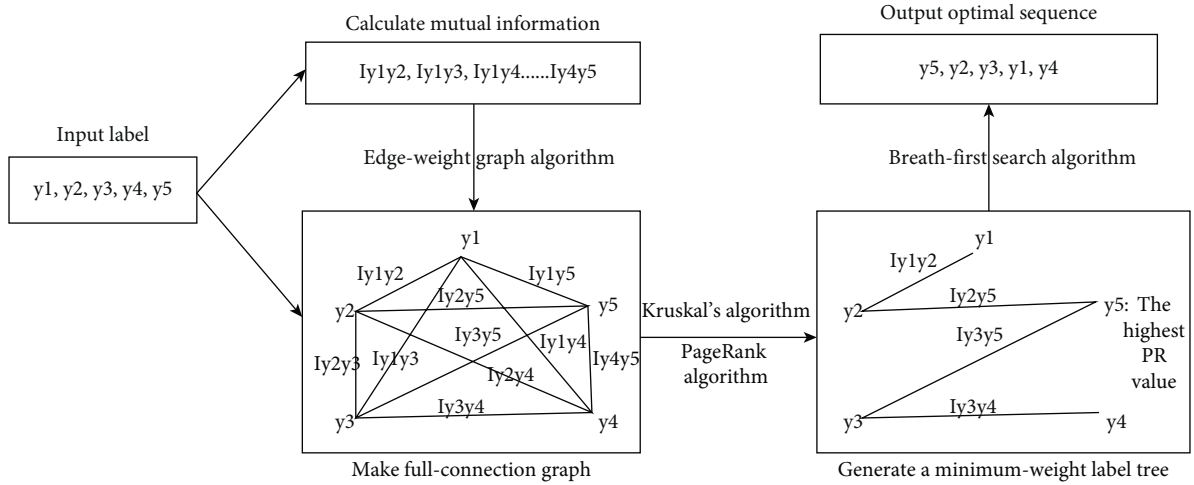
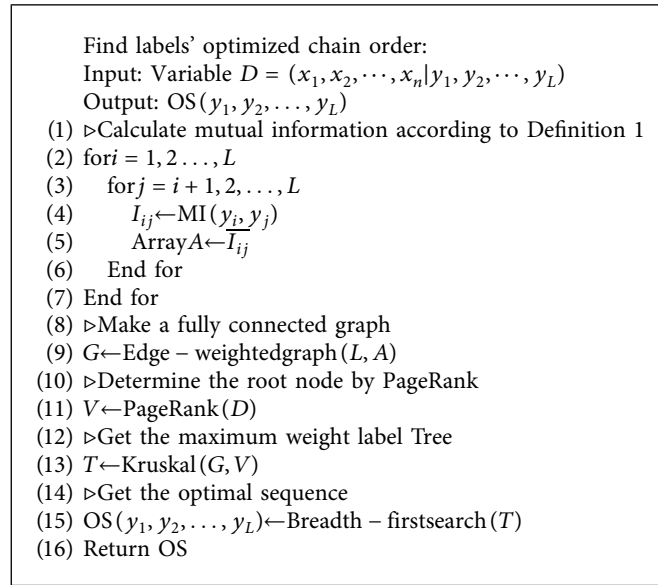


FIGURE 3: The OS computation process.



ALGORITHM 4

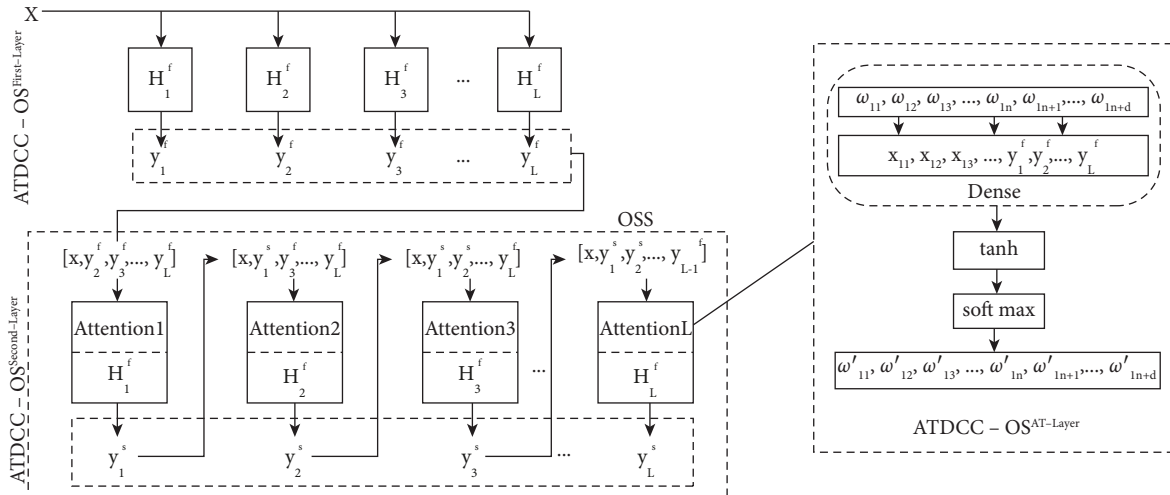


FIGURE 4: The DCC-OS model's design framework.


```

     $D$  is the training set,  $L$  is the labels' number
    TRAINING  $D = \{(x_i, Y_i) | i = 1, 2, \dots, m\}$ 
    (1) ▷Train the first-layer classifier
    (2) for  $j = 1, 2, \dots, L$ 
    (3)    $D_{y_j}^f \leftarrow$ 
    (4)   do  $x \leftarrow [x_{i1}, x_{i2}, \dots, x_{im}]$ 
    (5)    $D_{y_j}^f \leftarrow x \cup y_j$ 
    (6)    $H_{y_j}^f \leftarrow B(D_{y_j}^f)$ 
    (7) end for
    (8) ▷Use the OSS algorithm to obtain the label priority order to guide the training of the second-layer classifier
    (9) for  $j = \text{sort}1, 2, \dots, L$  by OSS ( $D$ )
    (10)   $D_{y_j}^s \leftarrow$ 
    (11)  do  $x \leftarrow [x_{i1}, x_{i2}, \dots, x_{im}, y_1, \dots, y_{j-1}, y_{j+1}, \dots, y_L]$ 
    (12)  ▷Compute attribute value weights using the attention mechanism
    (13)    $W \leftarrow \{W_{i1}, W_{i2}, \dots, W_{im+L}\}$ 
    (14)    $W \leftarrow \text{attention}(x)$ 
    (15)    $x' \leftarrow x \times W$ 
    (16)    $D_{y_j}^s \leftarrow x' \cup y_j$ 
    (17)    $H_{y_j}^s \leftarrow B(D_{y_j}^s)$ 
    (18)    $y_j = H_{y_j}^s(x')$ 
    (19) End for
  
```

ALGORITHM 5

```

    Classify( $x$ ): classify new instance  $X$ 
    (1) Global  $c^f = (y_1^f, y_2^f, \dots, y_L^f)$ ,  $c^s = (y_1^s, y_2^s, \dots, y_L^s)$ 
    (2) ▷Classify  $x$  for the first time using the first-layer classifier
    (3) for  $j = 1, 2, \dots, L$ 
    (4)   do  $x \leftarrow [x_{i1}, x_{i2}, \dots, x_{im}]$ 
    (5)    $y_j^f \leftarrow H_{y_j}^f(x)$ 
    (6) End for
    (7) ▷Classify  $x$  for the first time using the second-layer classifier
    (8)  $x' \leftarrow [x, y_1^f, y_2^f, \dots, y_L^f]$ 
    (9) for  $j = \text{sort}1, 2, \dots, L$  by OSS ( $D$ )
    (10)  do  $x'' \leftarrow x' \times W_j'$ 
    (11)   $y_j^s \leftarrow H_{y_j}^s(x'')$ 
    (12) End for
    (13) ▷Get the final classification result
    (14)  $c^s \leftarrow (y_1^s, y_2^s, \dots, y_L^s)$ 
    (15) Return  $c^s$ 
  
```

ALGORITHM 6

(2) Coverage [12]: coverage indicates that the algorithm can cover all possible labels. This metric describes how far or how deep we are to go in the tag list on average to include possible labels related to the document. At the perfect recall level, coverage is loosely related to accuracy. The smaller the value of coverage is, the better the algorithm will be. The coverage can be calculated as

$$\text{coverage}_D(H) = \frac{1}{P} \sum_{i=1}^P \max \text{rank}_f(x_i, y) - 1, \quad (11)$$

Where notation $\text{rank}(\cdot)$ denotes a sort and ranking function related to the classifier $H(\cdot)$.

(3) One-error metric [50]: one-error metric is used to indicate the proportion of examples where the top label does not fall into the selected label set. The bigger this metric is, the worse the algorithm will be. The one-error metric can be expressed as

$$\text{one-error}_D(H) = \frac{1}{P} \sum_{i=1}^P \left\| \left[\arg \max_{y \in \gamma} f(x_i, y) \notin Y_i \right] \right\|, \quad (12)$$

TABLE 1: Datasets with multiple labels.

Dataset	N	F	L	LCard	Type
Flags	194	19	7	3.392	Images
Emotion	593	72	6	1.87	Music
Birds	654	300	21	1.104	Audio
Medical	978	1449	45	1.245	Text
Enron	1702	1001	53	3.38	Text
Yeast	2417	103	14	4.24	Biology
BibTeX	7395	1836	159	2.40	Text

Where $f(\cdot)$ stands for a function associated with a classifier $H(\cdot)$ with multiple labels.

- (4) Ranking loss metric [12]: the ranking loss metric is related to those situations in which the classed labels of samples are not sorted in order, that is, in the label

serials, the classified labels (that are not related to the researched instance) fall into the previous related labels. The bigger this indicator is, the better the algorithm performance will be. The ranking loss metric can be expressed as

$$\text{rloss}_D(H) = \frac{1}{P} \sum_{i=1}^P \frac{1}{|Y_i| |\bar{Y}_i| \times |\{(y', y'') | f(x_i, y') \leq f(x_i, y''), (y', y'') \in Y_i \times \bar{Y}_i\}|} \quad (13)$$

- (5) Microaveraged AUC [50]: this metric shows the area covered by a ROC curve graph. Its value is from 0.1 to 1. This metric is directly exploited to review the classifier's performance. The smaller this metric is, the worse this algorithm will be. The microaveraged AUC metric is

$\text{AUC}_{\text{micro}} = |\{(x', x'', y', y'') | f(x', y') \geq f(x'', y''), (x', y') \in S^+, (x'', y'') \in S^-\}| / |S^+| |S^-|$, where $f(\cdot)$ is a real-valued function [51] and the following equations can be obtained:

$$S^+ = \{(x_i, y) | y \in Y_i, 1 \leq i \leq p\}. \quad (14)$$

$$S^- = \{(x_i, y) | y \notin Y_i, 1 \leq i \leq p\}, \quad (15)$$

where they denote label pairs' sets which are related or unrelated.

4.3. Experimental Setting. We use the dataset provided by the Mulan platform to evaluate all algorithms. The Mulan [48] is an open-source dataset used for classification with multiple labels, which is based on Weka. In this study, we use SMO as a basis for classification algorithms. Four different classifiers are utilized to carry out comparisons, including the DLMO-OS algorithm, the MBR algorithm, the CC algorithm, and the BR algorithm. We select 80% of instances from every dataset to act as training datasets, while we choose the rest to act as testing datasets. We adopt Adam [52] as the optimizer during the training process. We list the default parameters of Adam's hyperparameters as follows: let alpha be 0.001, set beta1 to be 0.9, let beta2 be 0.999, and set epsilon to be 10^{-8} . Our simulation platform includes the Intel(R) Xeon(R) E5-2630 CPU, 128 GB RAM, as well as the

operating system Centos 7.6. We design and implement the algorithms in the Java (JDK 1.8) running environment.

4.4. Results and Discussion. Figures 5–10 show the performance comparison among ATDCC-OS, DLMO, MBR, CC, and BR algorithms, using mean accuracy, coverage metric, single error metric, ranking loss metric, and microaverage AUC metric. We use the metric of the mean ranking (Ave. rank) parameter to review different classification results of the algorithms [53]. In these figures, each color represents an algorithm and the name of the algorithm has been listed in the upper left corner of the graph. The number on the top of each bar is the performance rank of the algorithms in the dataset. In Figures 5–9, the ordinate y -axis denotes the results of the evaluation, while the abscissa x -axis stands for the names of the dataset. In Figure 10, x -axis denotes the name of the algorithm, while y -axis shows the average rank of algorithms in all datasets.

Figure 5 shows the accuracy of each algorithm in each dataset. The ATDCC-OS method proposed in the study has the best performance in the dataset. Compared with other methods, except for the lowest accuracy in the yeast dataset, the accuracy in other datasets is the highest. Among them, the accuracy in the datasets of flags, emotions, and the medical dataset is over 80%.

In Figure 6, we can see the comparison of the microaverage AUC performance of the algorithms. The ATDCC-OS algorithm is also the most excellent and stable in terms of microaverage AUC performance. The performance of this algorithm is the best except for that in the birds dataset, and the performance in the medical dataset is 0.96.

Figure 7 shows the comparison of the coverage performance of each algorithm. The lower the coverage, the

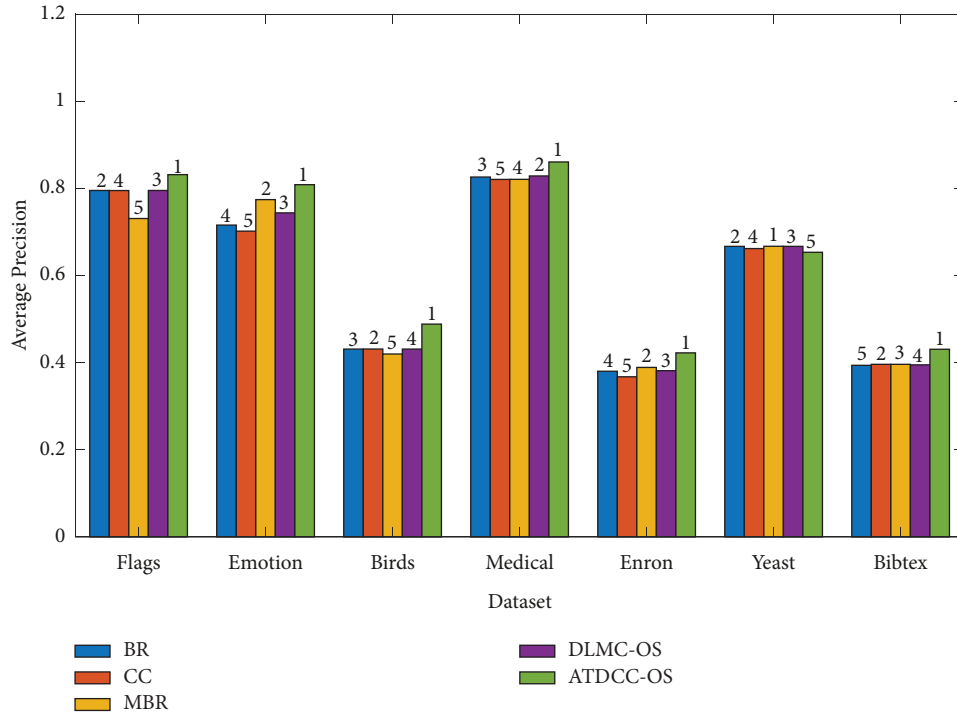


FIGURE 5: The average precision of algorithms.

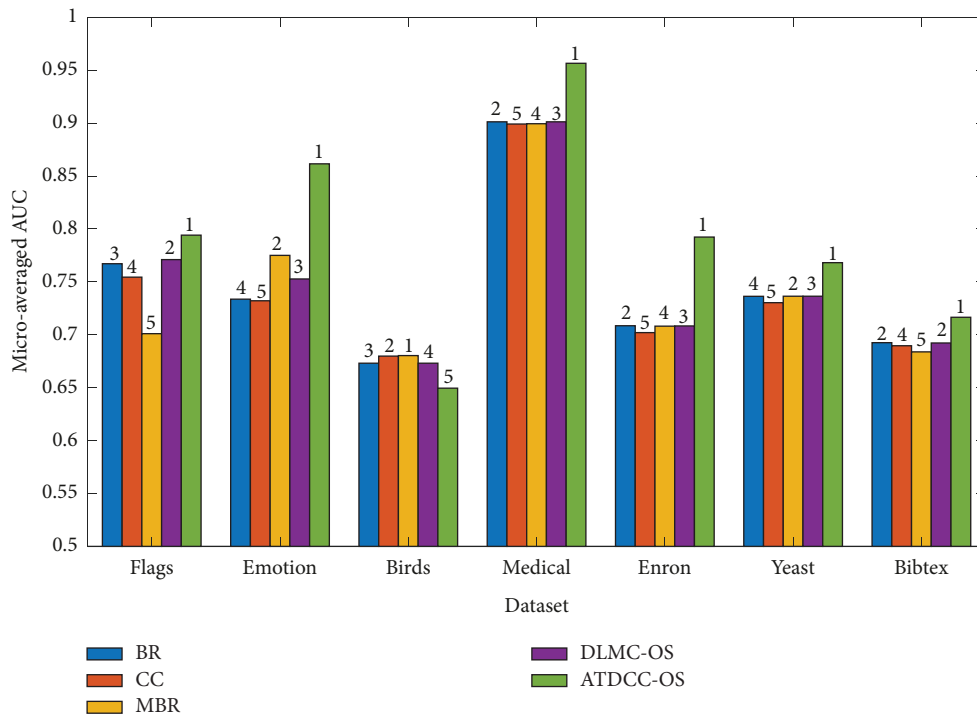


FIGURE 6: The microaveraged AUC performance of algorithms.

better the performance of the algorithm. The coverage performance of the ATDCC-OS algorithm proposed in the study is optimal in all datasets, and its coverage performance is less than 10% in flags, emotions, birds, medical datasets, and yeast datasets.

The single error performance of each algorithm is shown in Figure 8. The performance of the proposed ATDCC-OS algorithm in this graph is relatively unstable compared with other algorithms. However, from a comprehensive perspective, the performance of this algorithm is still good, and

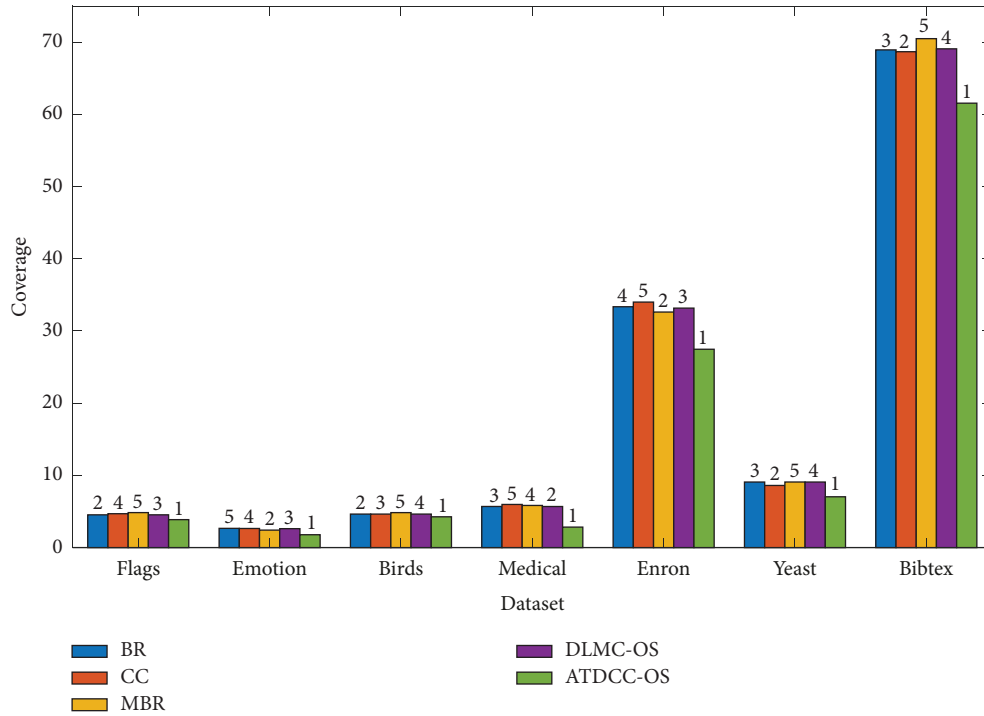


FIGURE 7: The coverage performance of algorithms.

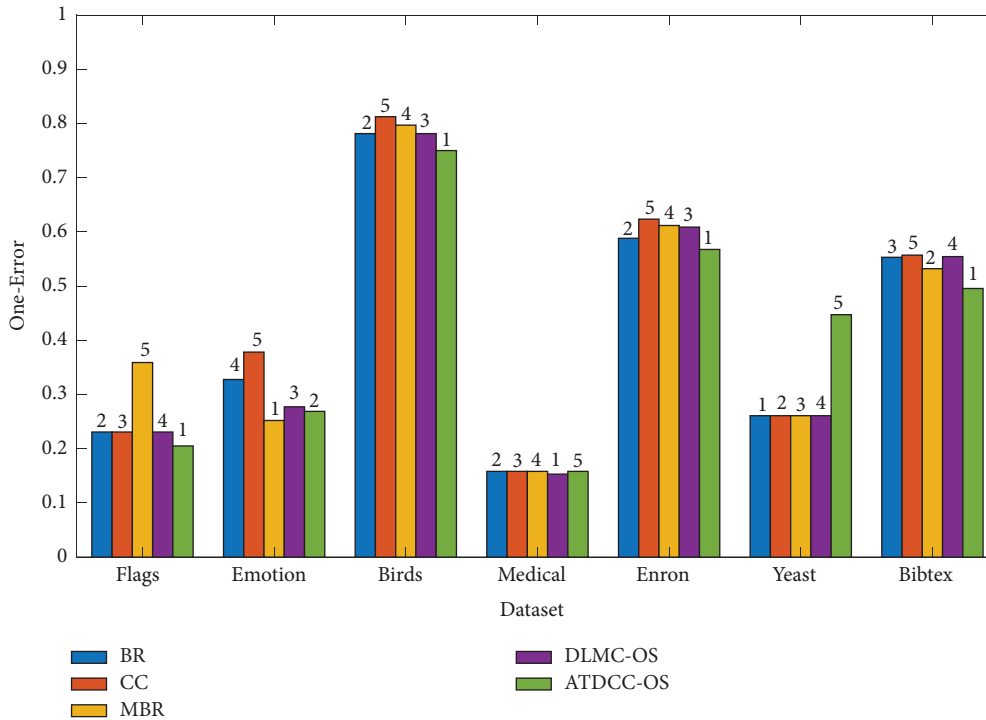


FIGURE 8: The one-error performance of algorithms.

the performance in the flags, birds, Enron, and BibTeX datasets is the best. In the emotion dataset, the performance of this algorithm is second only to that of the MBR algorithm.

From Figures 5–9, we can see that ATDCC-OS shows the optimal classification performance, while algorithm DLMC-OS presents better performance. However, other methods indicate worse performance. For all reviewing metrics, the

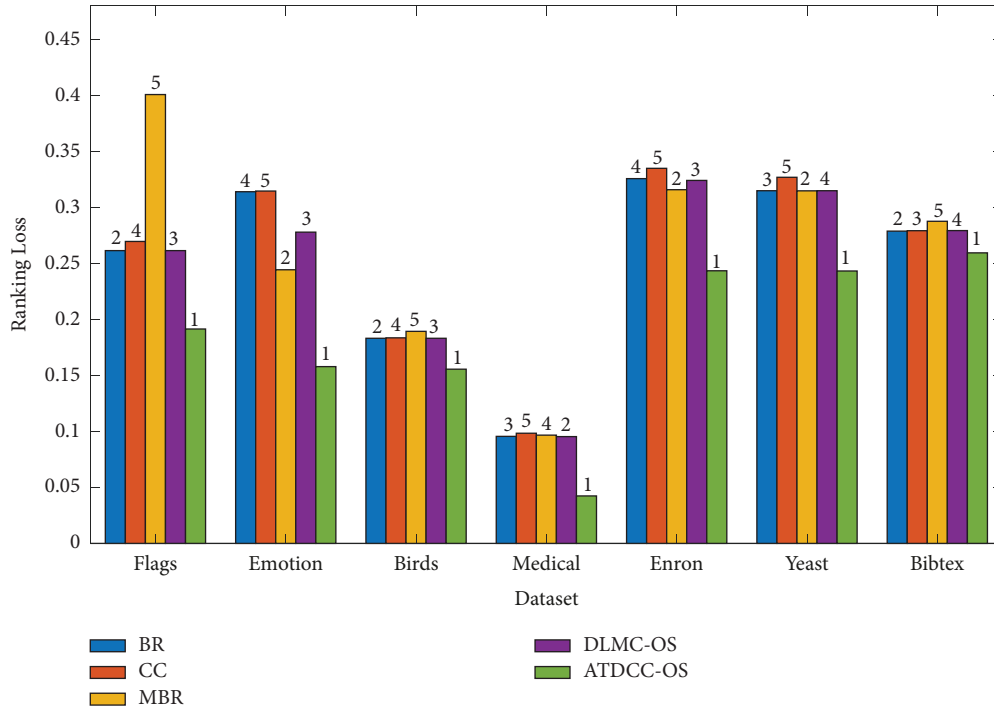


FIGURE 9: The ranking loss performance of algorithms.

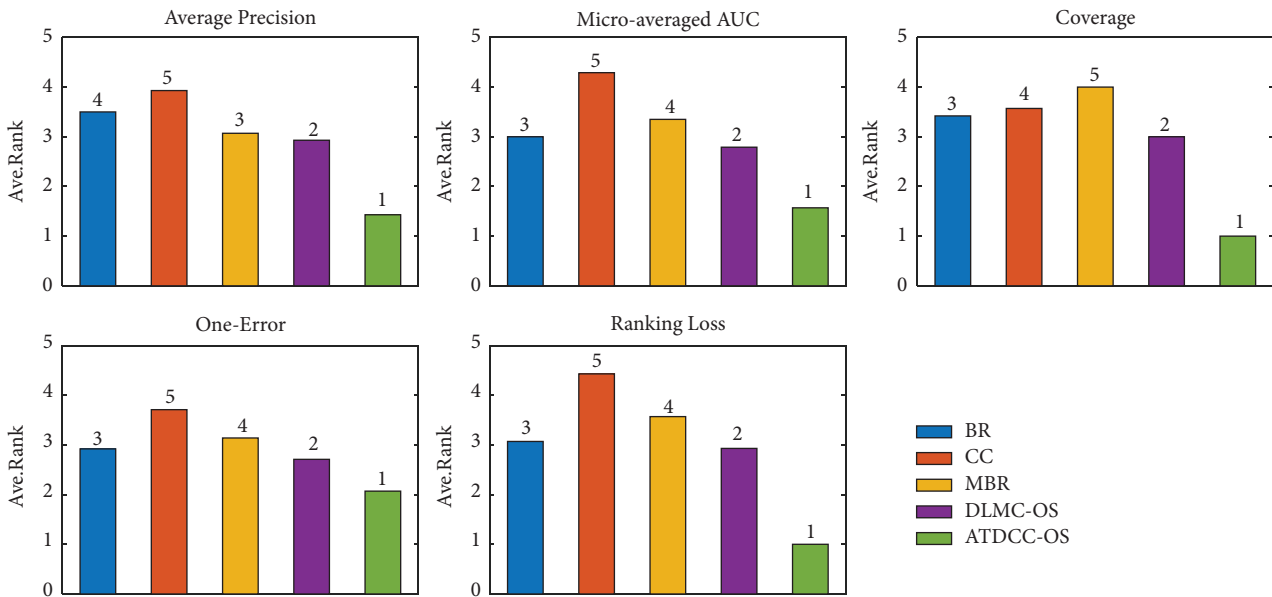


FIGURE 10: The average rank of five classifiers for different metrics.

mean precision metric and microaveraged AUC metric directly indicate the performance of the classifiers. The larger the values, the better the performance of the algorithms. According to Figures 5 and 6, we can see that the algorithm ATDCC-OS proposed in this study and the method DLMC-OS demonstrate much better performance compared with other algorithms. This is because they utilize a two-layer classification structure and the label information interaction

to create detailed classifiers. This design structure takes into consideration the interrelationship between labels. At the same time, the algorithm ATDCC-OS also exploits the classical attention mechanism theory to improve the sensibility of classifiers and adapt them to a variety of tasks. Three indicators, namely, coverage, ranking loss, and the one-error metric are often exploited to decide and find irrelevant labels in classification results. As shown in

Figures 7–9, we find that the algorithm ATDCC-OS and the previous algorithm DLMC-OS also demonstrate better performance compared with the rest of the algorithms, while the BR approach presents a medium performance. The MBR method and the CC approach are the worst in this metric. This is because the algorithm ATDCC-OS and the previous approach DLMC-OS utilize optimization algorithms to train all classifiers in order. The randomness of serials in the CC method and the MBR approach directly leads to poorer performance. On the contrary, the BR method does not take into account the sequence of the labels, while it shows better performance.

The loss performance of each algorithm is shown in Figure 9. Among them, the ATDCC-OS algorithm is the most excellent in terms of loss performance. In all datasets, the performance of this algorithm is one level better than other algorithms. In the medical dataset, the loss performance is about 0.03.

From Figure 10, we can see the comprehensive performance ranking of the comparison algorithms in various indicators. Among all the indexes, the ATDCC-OS algorithm has the best performance. The comprehensive performance of the DLMC-OS algorithm is second only to that of the ATDCC-OS algorithm, and the subsequent performance is different in different algorithms.

Figure 10 shows the mean ranking performance metrics of the five classifiers for mean accuracy, coverage metric, single error metric, ranking loss metric, and microaverage AUC metric.

From our simulations, we can find that our algorithm ATDCC-OS outperforms the rest of the algorithms for most of the datasets, while it performs poorly in yeast and birds. As we all know, this algorithm cannot obtain the best performance for all types of different test datasets [10]. The algorithm performance is related not only to the detailed structure of the algorithm but also to the dataset's detailed type and size, as well as labels' balance in our test dataset.

Figures 11 and 12 show the plots of the percentage of training data versus average precision and ranking loss. These two figures illustrate how the percentage change of training data affects the enhancement of performance. In this experiment, we take the emotions dataset as an example for both comparisons.

Figure 11 shows the change curve for average precision under the two pairs of classifiers scale with respect to the percentage of training data. From Figure 11, we observe that the average precision is elevated for the four classifiers when the percentage of training data increases. When the percentage of training data is between 10% and 30%, the accuracy of all algorithms floats up and down. When the percentage of training data is over 30%, the average precision of the ATDCC-OS and DLMC-OS rises steadily, while MBR needs to reach 40%, and CC and BR need to reach 60%. Overall, as the training data increase, ATDCC-OS shows better performance than DLMC-OS, followed by MBR and BR, while CC is the worst.

From Figure 12, we can see the results of the comparison in terms of ranking loss. In this figure, as the percentage of training data increases, the ranking loss of ATDCC-OS and

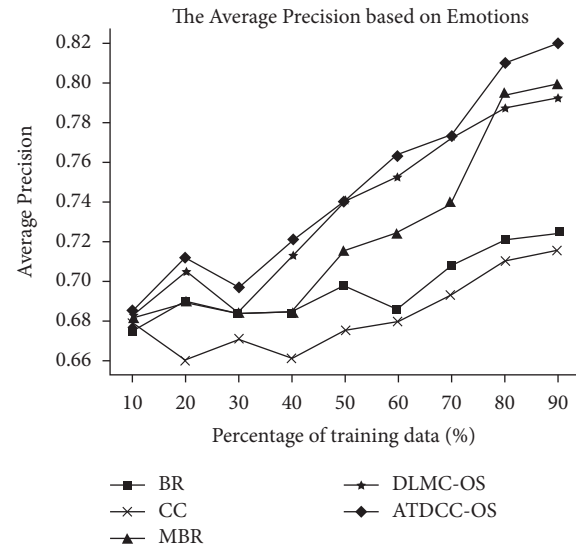


FIGURE 11: Average precision with different percentages of training data.

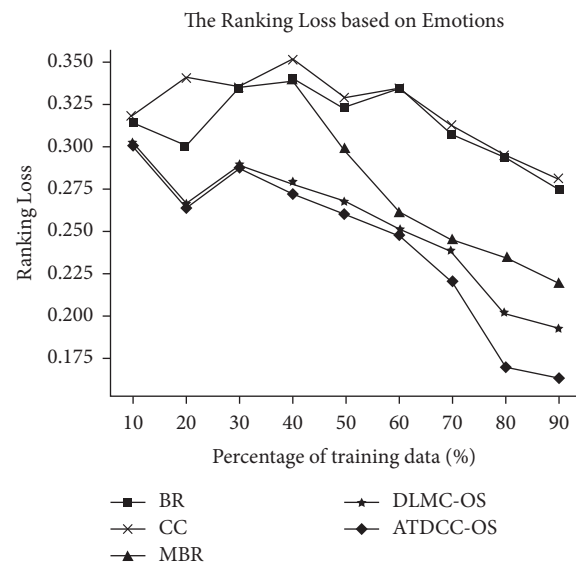


FIGURE 12: Ranking loss with different percentages of training data.

DLMC-OS tend to decrease steadily, compared to MBR, CC, and BR. When the training data is between 10% and 40%, the ranking loss of each algorithm is unstable, among which the MBR fluctuates the most, followed by CC and BR, while ATDCC-OS and DLMC-OS perform better. When the dataset is larger than 40%, the ranking loss curves of all algorithms show a downward trend. ATDCC-OS still presents the lowest loss in such a scenario.

5. Conclusion

In this study, we propose a simple and effective multilabel classification model (ATDCC-OS) that integrates the multilabel classification framework of three classic problem-

conversion types. It fully explores all kinds of advantages of every method to resolve these issues without considering the correlation among labels when performing classifications. In order to further improve the performance of classification, the algorithm solves the problem of nonreal-time label information interaction in the second-layer chained classification model by introducing the idea of “update replacement.” At the same time, the algorithm dynamically calculates the weight values of all feature attributes through an attention mechanism in order to add more important attribute features to the current classification target for each classifier. It is helpful to add the classification sensibility of classifiers, which greatly improves the preciseness of classification. Five different metrics are utilized to describe different algorithms on seven different datasets. The results of the experiments show that the proposed method obtains high predictive performance compared with the state-of-the-art multilabel classification methods in most cases. In terms of average accuracy, the average accuracy of the ATDCC-OS algorithm is basically the highest in all datasets, and the accuracy in flags, emotions, and the medical dataset is more than 80%. In the microaverage AUC performance, the performance of the ATDCC-OS algorithm in all datasets is the best except for that in the bird’s dataset, and the performance in the medical dataset is 0.96. In terms of coverage performance, the ATDCC-OS algorithm has the best coverage performance in all datasets, and its coverage performance is less than 10% in some datasets. In single error performance, this algorithm has the best comprehensive performance. In the loss performance, the algorithm has a loss performance of about 0.03 in the medical dataset. Based on the above results, it is concluded that the performance of the proposed ATDCC-OS algorithm is the best. This is only the preliminary result of this study. In the future, we will further optimize the algorithm to solve the problem of time complexity caused by the model structure, and we will also try to apply the algorithm to solve classification problems in everyday work and life. Finally, we hope that the research work in this study can provide some reference and assistance to researchers or scholars in the field of multilabel classification of problem transformation types.

Data Availability

The data used and/or analyzed during the current study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Key Research Project of Education Department of Sichuan Province of China (18ZA319).

References

- [1] R. Wang, R. Ridley, X. Su, W. Qu, and X. Dai, “A novel reasoning mechanism for multi-label text classification,” *Information Processing & Management*, vol. 58, no. 2, Article ID 102441, 2021.
- [2] J. Cai, W. Sun, J. Guan, and I. You, “Multi-ECGNet for ECG arrhythmia multi-label classification,” *IEEE Access*, vol. 8, pp. 110848–110858, 2020.
- [3] H. Chougrad, H. Zouaki, and O. Alheyane, “Multi-label transfer learning for the early diagnosis of breast cancer,” *Neurocomputing*, vol. 392, pp. 168–180, 2020.
- [4] J. Saha, C. Chowdhury, and S. Biswas, *Review of Machine Learning and Deep Learning Based Recommender Systems for Health Informatics*, pp. 101–126, Springer, Cham, 2020.
- [5] J. Wu, V. S. Sheng, J. Zhang et al., “Multi-label active learning algorithms for image classification: overview and future promise,” *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–35, 2021.
- [6] M. Bogaert, J. Lootens, D. Van den Poel, and M. Ballings, “Evaluating multi-label classifiers and recommender systems in the financial service sector,” *European Journal of Operational Research*, vol. 279, no. 2, pp. 620–634, 2019.
- [7] G. N. Karagoz, A. Yazici, T. Dokeroglu, and A. Cosar, “A new framework of multi-objective evolutionary algorithms for feature selection and multi-label classification of video data,” *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 1, pp. 53–71, 2021.
- [8] Z. M. Chen, X. S. Wei, and P. Wang, “Multi-label image recognition with graph convolutional networks,” *CVPR*, vol. 27, 2019.
- [9] G. Wu, R. Zheng, Y. Tian, and D. Liu, “Joint ranking SVM and binary relevance with robust low-rank learning for multi-label classification,” *Neural Networks*, vol. 122, pp. 24–39, 2020.
- [10] J. Read, B. Pfahringer, G. Holmes, and E. Frank, “Classifier chains: a review and perspectives,” *Journal of Artificial Intelligence Research*, vol. 70, pp. 683–718, 2021.
- [11] K. Feng, C. Li, M. Zhang, and X. Liu, “Simulation and computational study of CFD on tube MBR membrane assembly,” *Discrete Dynamics in Nature and Society*, vol. 2021, no. 7–8, pp. 1–8, 2021.
- [12] J. Li, P. Wei, S. Yang, J. Wu, P. Liu, and X. He, “Crystal-KMC: parallel software for lattice dynamics Monte Carlo simulation of metal materials,” *Tsinghua Science and Technology*, vol. 23, no. 4, pp. 501–510, 2018.
- [13] C. Pan, J. Tan, D. Feng, and Y. Li, “Very short-term solar generation forecasting based on LSTM with temporal attention mechanism,” in *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*, Chengdu, China, December 2019.
- [14] Q. Ai, Y. Kang, A. Wang, X. Li, and F. Li, “An effective semi-supervised multi-label least squares twin support vector machine,” *IEEE Access*, vol. 8, pp. 213460–213472, 2020.
- [15] X. Xu, D. Shan, S. Li, T. Sun, P. Xiao, and J. Fan, “Multi-label learning method based on ML-RBF and laplacian ELM,” *Neurocomputing*, vol. 331, pp. 213–219, 2019.
- [16] F. Javed, J. Ahmed, and M. Hayat, “ML-RBF: predict protein subcellular locations in a multi-label system using evolutionary features,” *Chemometrics and Intelligent Laboratory Systems*, vol. 203, Article ID 104055, 2020.

- [17] X. Zhu, C. Ying, J. Wang, J. Li, X. Lai, and G. Wang, "Ensemble of ML-KNN for classification algorithm recommendation," *Knowledge-Based Systems*, vol. 221, Article ID 106933, 2021.
- [18] D. Zhu, H. Zhu, X. Liu et al., "CREDO: efficient and privacy-preserving multi-level medical pre-diagnosis based on ML-kNN," *Information Sciences*, vol. 514, pp. 244–262, 2020.
- [19] D. Jiaman, Z. Shujie, L. Runxin, F. Xiaodong, and J. Lianyin, "Association rules-based classifier chains method," *IEEE Access*, vol. 10, pp. 18210–18221, 2022.
- [20] E. Hüllermeier, J. Fürnkranz, and M. E. Loza, "Rule-based multi-label classification: challenges and opportunities," in *Proceedings of the International Joint Conference on Rules and Reasoning*, Oslo, Norway, July 2020.
- [21] P. Teisseyre, "Classifier chains for positive unlabelled multi-label learning," *Knowledge-Based Systems*, vol. 213, Article ID 106709, 2021.
- [22] N. K. Mishra and P. K. Singh, "Linear ordering problem based classifier chain using genetic algorithm for multi-label classification," *Applied Soft Computing*, vol. 117, Article ID 108395, 2022.
- [23] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.
- [24] E. Lieskovská, M. Jakubec, R. Jarina, and M. Chmulik, "A review on speech emotion recognition using deep learning and attention mechanism," *Electronics*, vol. 10, no. 10, p. 1163, 2021.
- [25] S. He, H. R. Tavakoli, A. Borji, and N. Pugeault, "Human attention in image captioning: dataset and analysis," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8528–8537, Seoul, Korea (South), October 2019.
- [26] Z. Wu, D. Pi, J. Chen, M. Xie, and J. Cao, "Rumor detection based on propagation graph neural network with attention mechanism," *Expert Systems with Applications*, vol. 158, Article ID 113595, 2020.
- [27] Y. Bazi, L. Bashmal, M. M. A. Rahhal, R. A. Dayil, and N. A. Ajlan, "Vision transformers for remote sensing image classification," *Remote Sensing*, vol. 13, no. 3, p. 516, 2021.
- [28] Y. Zhou, L. Pan, C. Bai, S. Luo, and Z. Wu, "Self-selective attention using correlation between instances for distant supervision relation extraction," *Neural Networks*, vol. 142, pp. 213–220, 2021.
- [29] Q. Xu, L. Zhu, T. Dai, and C. Yan, "Aspect-based sentiment classification with multi-attention network," *Neurocomputing*, vol. 388, pp. 135–143, 2020.
- [30] J. Gan, "Music feature classification based on recurrent neural networks with channel attention mechanism," *Mobile Information Systems*, vol. 202110 pages, 2021.
- [31] A. V. Savchenko, K. V. Demochkin, and L. V. Savchenko, "Neural attention mechanism and linear squeezing of descriptors in image classification for visual recommender systems," *Optical Memory & Neural Networks*, vol. 29, no. 4, pp. 297–304, 2020.
- [32] Y. Xiao, Y. Li, J. Yuan, S. Guo, Y. Xiao, and Z. Li, "History-based attention in Seq2Seq model for multi-label text classification," *Knowledge-Based Systems*, vol. 224, Article ID 107094, 2021.
- [33] D. Wang, Z. Zhang, Y. Jiang et al., "DM3Loc: multi-label mRNA subcellular localization prediction and analysis based on multi-head self-attention mechanism," *Nucleic Acids Research*, vol. 49, no. 8, p. e46, 2021.
- [34] A. Hashemi, M. B. Dowlatshahi, and H. Nezamabadi-Pour, "A bipartite matching-based feature selection for multi-label learning," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 2, pp. 459–475, 2021.
- [35] P. Zhang, W. Gao, J. Hu, and Y. Li, "Multi-label feature selection based on high-order label correlation assumption," *Entropy*, vol. 22, no. 7, p. 797, 2020.
- [36] R. Xia, Y. Chen, and B. Ren, "Improved Anti-occlusion Object Tracking Algorithm Using Unscented Rauch-Tung-Striebel Smoother and Kernel Correlation Filter," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, 2022.
- [37] J. Kudler-Flam, L. Nie, and S. Ryu, "Conformal field theory and the web of quantum chaos diagnostics," *Journal of High Energy Physics*, vol. 2020, no. 1, pp. 175–233, 2020.
- [38] O. B. Ericok and J. K. Mason, "Foundations of a Finite Non-equilibrium Statistical Thermodynamics: Extrinsic Quantities," 2022, <https://arxiv.org/abs/2201.02777>.
- [39] C. Wen, Q. Y. Hu, B. Y. Pu, and Y. Y. Huang, "Acceleration of an adaptive generalized Arnoldi method for computing PageRank," *AIMS Mathematics*, vol. 6, no. 1, pp. 893–907, 2021.
- [40] Y. Gao, X. Yu, and H. Zhang, "Overlapping community detection by constrained personalized PageRank," *Expert Systems with Applications*, vol. 173, no. 1, p. 114682, 2021.
- [41] X. Zhang, J. You, H. Xue, and J. Wang, "A decentralized PageRank based content dissemination model at the edge of network," *International Journal of Web Services Research*, vol. 17, no. 1, pp. 1–16, 2020.
- [42] U. S. Premarathne, "Preference biased edge weight assignment for connectivity-based resilience computation in telecommunication networks," *IT Professional*, vol. 22, no. 6, pp. 26–32, 2020.
- [43] J. Zhang, W. Feng, T. Yuan, J. Wang, and A. K. Sangaiah, "SCSTCF: spatial-channel selection and temporal regularized correlation filters for visual tracking," *Applied Soft Computing*, vol. 118, Article ID 108485, 2022.
- [44] D. Di Caprio, A. Ebrahimnejad, H. Alrezaamiri, and F. J. Santos-Arteaga, "A novel ant colony algorithm for solving shortest path problems with fuzzy arc weights," *Alexandria Engineering Journal*, vol. 61, no. 5, pp. 3403–3415, 2022.
- [45] J. Dutta, P. S. Barma, S. Kar, and T. De, "A Modified Kruskal's algorithm to improve genetic search for open vehicle routing problem," *International Journal of Business Analytics*, vol. 6, no. 1, pp. 55–76, 2019.
- [46] Y. Chen, V. Phonevilay, J. Tao et al., "The face image super-resolution algorithm based on combined representation learning," *Multimedia Tools and Applications*, vol. 80, no. 20, pp. 30839–30861, 2021.
- [47] J. P. Zhou, L. Chen, T. Wang, and M. Liu, "iATC-FRAKEL: a simple multi-label web server for recognizing anatomical therapeutic chemical classes of drugs with their fingerprints only," *Bioinformatics*, vol. 36, no. 11, pp. 3568–3569, 2020.
- [48] J. Yang, S. Hu, Q. Wang, and S. Fong, "Discriminable multi-label attribute selection for pre-course student performance prediction," *Entropy*, vol. 23, no. 10, p. 1252, 2021.
- [49] A. Sinha Roy, R. Biswas, and A. S. Dhar, "On fast and exact computation of error metrics in approximate LSB adders," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 28, no. 4, pp. 876–889, 2020.
- [50] M. C. Pinto, A. Rodriguez-Ruiz, K. Pedersen et al., "Impact of artificial intelligence decision support using deep learning on breast cancer screening interpretation with single-view wide-angle digital breast tomosynthesis," *Radiology*, vol. 300, no. 3, pp. 529–536, 2021.

- [51] M. L. Welch, C. McIntosh, A. Traverso et al., “External validation and transfer learning of convolutional neural networks for computed tomography dental artifact classification,” *Physics in Medicine and Biology*, vol. 65, no. 3, p. 035017, 2020.
- [52] M. Feurer and F. Hutter, *Hyperparameter optimization// Automated Machine Learning*, pp. 3–33, Springer, New York, 2019.
- [53] A. Zeng, H. Yu, Q. Da, Y. Zan, and C. Miao, “Accelerating ranking in E-Commerce search engines through contextual factor selection,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 08, pp. 13212–13219, 2020.