**Algorithm Introduction**

|  |
| --- |
| **Algorithm 1 Hybrid Piecewise Affine (HyPA)** |
| 1. Initialize the number of segment *K* and the number of samples to be fitted *Np*. 2. The coefficients  and intercepts  are obtained by solving the quadratic programming problem: . |
| 1. Based on **Step 1**, the frequency nadir is predicted . |
| 1. Input  and  into **Algorithm 2** to obtain the optimal classification threshold . |
| 1. . |
| 1. Perform SVM training on ,  and are obtained. |

|  |
| --- |
| **Algorithm 2 Optimal Classification Threshold** |
| 1. Initialize best\_info\_gain and . |
| 1. sorted\_indices = np.argsort(). |
| 1. = ,  # Sort in ascending order. |
| 1. for *i* range (len()):   gain = information\_gain (, , ) # Calculate the information gain.  if gain > best\_info\_gain:  best\_info\_gain = gain  = ( + ) / 2 |
| 1. Output |

**Algorithm Validation**

According to the HyPA algorithm introduced in Section II, a total of 96 training sessions are performed for the 24 scheduling periods of the four grids. Each training session involves 16,000 training samples and 4,000 testing samples. The quadratic programming problem in the HyPA algorithm is set with a segment number (*K*) of 3, and 100 samples are uniformly extracted from the training set for fitting. Additionally, it is clear that PWL is a component of the HyPA algorithm, so they share the same parameter settings (*K*=3 and *Np*=100).

The performance of the HyPA algorithm is compared with PWL, SVM, and ASVMTREE in terms of accuracy and F1-score. We calculate the average of these 96 sets for ease of display. As shown in Table Ⅰ, HyPA achieves the highest accuracy (0.9878), slightly outperforming ASVMTREE (0.9870) and SVM (0.9868), while PWL shows the lowest accuracy (0.9765).

In terms of the F1-score, which reflects a balance between precision and recall, HyPA again performs best with a score of 0.9535. This suggests that HyPA not only provides accurate predictions overall but also maintains a good balance between identifying positive and negative instances. ASVMTREE and SVM follow with similar F1-scores (0.9505 and 0.9502, respectively), whereas PWL scores the lowest (0.9189). These results indicate that HyPA is the most effective algorithm for ensuring both high accuracy and balanced predictions when it comes to convexifying the FN constraint.

TABLE I

The average performance of the four algorithms

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | PWL | SVM | ASVMTREE | HyPA |
| Accuracy | 0.9765 | 0.9868 | 0.9870 | 0.9878 |
| F1-score | 0.9189 | 0.9502 | 0.9505 | 0.9535 |