applied sciences

MDPI

Article

Interclass Interference Suppression in Multi-Class Problems

Jinfu Liu [1,*], Mingliang Bai [1], Na Jiang [1], Ran Cheng [1], Xianling Li [2], Yifang Wang [3] and Daren Yu [1]

[1] Harbin Institute of Technology, Harbin 150001, China; mingliangbai@outlook.com (M.B.); 18S102144@stu.hit.edu.cn (N.J.); 20S102134@stu.hit.edu.cn (R.C.); yudaren@hit.edu.cn (D.Y.)
[2] Science and Technology on Thermal Energy and Power Laboratory, Wuhan 430205, China; xianlingliwh@126.com
[3] Systems Engineering Research Institute, China State Shipbuilding Corporation Limited, Beijing 100000, China; wyf_0772@163.com
* Correspondence: jinfuliu@hit.edu.cn

Abstract: Multi-classifiers are widely applied in many practical problems. But the features that can significantly discriminate a certain class from others are often deleted in the feature selection process of multi-classifiers, which seriously decreases the generalization ability. This paper refers to this phenomenon as interclass interference in multi-class problems and analyzes its reason in detail. Then, this paper summarizes three interclass interference suppression methods including the method based on all-features, one-class classifiers and binary classifiers and compares their effects on interclass interference via the 10-fold cross-validation experiments in 14 UCI datasets. Experiments show that the method based on binary classifiers can suppress the interclass interference efficiently and obtain the best classification accuracy among the three methods. Further experiments were done to compare the suppression effect of two methods based on binary classifiers including the one-versus-one method and one-versus-all method. Results show that the one-versus-one method can obtain a better suppression effect on interclass interference and obtain better classification accuracy. By proposing the concept of interclass inference and studying its suppression methods, this paper significantly improves the generalization ability of multi-classifiers.

Keywords: interclass interference; multi-class classification problem; suppression method; one-versus-all (OVA); one-versus-one (OVO); generalization ability

Citation: Liu, J.; Bai, M.; Jiang, N.; Cheng, R.; Li, X.; Wang, Y.; Yu, D. Interclass Interference Suppression in Multi-Class Problems. *Appl. Sci.* **2021**, *11*, 450. https://doi.org/10.3390/app11010450

Received: 13 November 2020
Accepted: 28 December 2020
Published: 5 January 2021

1. Introduction

Classification tasks exist widely in real-world applications, such as computer vision [1], fault diagnosis [2,3], human action recognition [4], face recognition [5], image recognition [6], material science [7], big data analysis, [8,9] etc. Many of them are classification tasks that include more than two classes, which are called multi-class problems [10]. Multi-class problems are usually more difficult to solve than binary classification problem because multi-class problems need to distinguish more classes. Multi-class problems are quite common in the real world [11–17]. For example, an image recognition device needs to distinguish many different kinds of images. An industrial fault diagnosis system [18,19] needs to diagnose what fault occurs in the machine. A sentiment analysis system [20,21] needs to classify different attitudes of people according to the information in social media such as Twitter and Facebook. Improving the classification accuracy of multi-class problems has great significance in actual applications [22].

Multi-class problems can be addressed in many ways. Among the well-recognized classifiers, some classifiers can handle multi-class problems directly, such as decision tree [23,24], rough set-based classifiers [25–28], neural networks [29,30], naive Bayes [31], K-nearest neighbor (KNN) [32] and so forth. Others can only deal with binary classification problems like support vector machine (SVM) [33,34]. Although there are some studies on how to use binary classifiers to solve multi-class problems, these studies usually focus on the situations where some classifiers cannot solve multi-class problems by themselves.

For these classifiers, current studies focus on how to address multi-class problems by decomposing multi-class problems into binary classification problems. When binary classifiers are used to address multi-class problems, the most common strategies are currently one-versus-one (OVO) strategies [35] and one-versus-all (OVA) strategies [36,37]. For the OVO strategy, Galar et al. [38] developed a distance-based combination strategy, which weights the competence of the outputs of the base classifiers depending on the closeness of the query instance to each one of the classes. Galar et al.'s method reduced the effect of the non-competent classifiers, enhancing the results obtained by the state-of-the-art combinations for the OVO strategy. Galar et al. [39] proposed a dynamic classifier selection strategy for the OVO scheme that tries to avoid non-competent classifiers. For the OVA approach, Dinh et al. [40] proved new fast learning rates for OVA multi-class plug-in classifiers trained either from exponentially strongly mixing data or from data generated by a converging drifting distribution. They reported that their results retain the optimal learning rate in the independently identically distributed case in contrast to previous works for least squares SVMs under the binary-class setting. In literature [41], Rebetez et al. used an ensemble of binary OVA neural network classifiers and reported that the performances of their methods are comparable to lazy learning methods that require the whole dataset.

Currently, the generalization ability of multi-classifiers remains a longstanding challenge [42]. Many scholars have carried out relevant research on it. Eiadon et al. [43] decomposed the classes into subsets by embedding a structure of binary trees and put forward a novel splitting criterion based on minimizing generalization errors and greedy search procedures across the classes. Lei et al. [44] established data-dependent error bounds in terms of complexities of a linear function class defined on a finite set induced by training examples, for which they showed tight lower and upper bounds, applied the results to several prominent multi-class learning machines and exhibited a tight dependency on the number of classes. Kantavat et al. [45] proposed new methods for support vector machine (SVM) using tree architecture for multi-class classification and reported that their proposed methods run much faster than the traditional techniques but still provide comparable accuracy. Dhifli et al. [46] introduced a novel multi-class classification method for the open-set problem and proved the efficiency of their approach in classifying novel instances from known as well as unknown classes through experiments on benchmark datasets and synthetic datasets.

From the literature above, it can be noticed that many researchers focus on improving the generalization ability of classifiers. However, these researchers merely focus on generalized classification problems. There is little research specifically for multi-class problems. Researchers usually regard multi-class problems as ordinary classification problems to improve their generalization ability. Few researchers focus on the unique characteristics of multi-class problems. Although there are some studies on how to use binary classifiers to solve multi-class problems, these studies usually focus on the situations where some classifiers cannot solve multi-class problems by themselves. The differences between common binary classifiers and multi-classifiers are not studied sufficiently. This is the problem that this paper deals with.

This paper observes that there exists a special phenomenon in multi-class problems, which are quite different from binary classification problems. The features that can significantly discriminate a certain class from others are often deleted in the process of feature selection because they cannot discriminate among other classes. By contrast, the features that can discriminate among all classes are often reserved after feature selection. Although there are small errors in the training set, there may be large errors in test sets because reserved features cannot reflect the essence of a certain class. This phenomenon is an inherent problem in feature selection of multi-class problems and becomes more serious as the number of classes increases. This phenomenon significantly decreases the generalization ability of multi-classifiers. To improve the generalization ability of multi-classifiers better, this phenomenon must be eliminated or suppressed. Therefore, this paper elaborates this special phenomenon in multi-class problems, names it interclass interference, analyzes its

reason systematically, designs its suppression methods and compares their suppression effects on interclass interference through 10-fold cross-validation experiments in 14 UCI datasets. The main contributions of this paper are listed as follows.

Firstly, this paper observes a special phenomenon in the feature selection of multi-classifiers and names it interclass inference. To the best of our knowledge, this is the first time that the concept of interclass inference is proposed. Secondly, the reasons for interclass inference are analyzed. The essence of interclass inference is revealed in this paper. The negative influence of interclass inference on the generalization ability of multi-class classifiers is also analyzed in detail. Thirdly, this paper summarizes the possible methods for suppressing interclass inference and compares their effects on interclass inference suppression.

The rest of this paper is organized as follows. In Section 2, the concept of interclass interference in multi-classifiers is proposed and its reason is analyzed systematically. In Section 3, suppression methods of interclass interference are summarized. In Section 4, interclass interference suppression algorithms used for the comparison experiments in this paper are designed. Section 5 shows the results of our comparison experiments. Section 6 concludes the paper.

2. Interclass Interference of Multi-Class Problems

Interclass interference is an inherent problem in feature extraction of multi-class problems. The removal of key features of some classes as redundant features is the direct cause of interclass interference in multi-class problems.

In this section, this paper will use an example of multi-class vibration fault diagnosis of steam turbine shown in Table 1 to introduce the concept of interclass interference and reveal its essence. According to expert knowledge, the vibration signal of 0.4f~0.6f, 1f and 2f, namely the features a_1, a_2 and a_3, are the essential features that can significantly discriminate oil film whirl, unbalance and misalignment from other faults, respectively [47–49]. Note that f denotes the rotation frequency in Table 1.

Table 1. Decision table for vibration fault diagnosis of a steam turbine.

Fault Instances	**Conditional Attributes (*C*)**			**Decision Attributes (*D*)**
U	**0.4f~0.6f (a_1)**	**1f (a_2)**	**2f (a_3)**	**Fault (d)**
x_1	Low	High	Low	Unbalance
x_2	Low	High	Medium	Unbalance
x_3	Low	High	High	Unbalance
x_4	Low	Medium	Low	Unbalance
x_5	Low	Medium	Medium	Unbalance
x_6	Low	Low	High	Misalignment
x_7	Low	Medium	Medium	Misalignment
x_8	Low	Medium	High	Misalignment
x_9	High	Low	Low	Oil Film Whirl
x_{10}	High	Low	Medium	Oil Film Whirl

In machine learning, feature selection is an essential step in classification, which can reduce dimensionality. In Table 1, a_1 is the feature that can significantly distinguish the oil film whirl fault from other two fault classes including unbalance and misalignment. Meanwhile, the fact that attribute a_1 is the intrinsic characteristics of oil film whirl also meets the physical laws [50,51]. However, feature a_1 cannot distinguish unbalance and misalignment faults. In the feature selection process, features that can distinguish all classes have the highest priorities to be preserved. Feature a_1 may be deleted because it cannot distinguish between unbalance and misalignment fault. If feature a_1 is deleted, the generalization ability of this multi-classifier will be bad because the essential feature of the oil film whirl is deleted. This phenomenon is called interclass inference in this paper. In the

following part of this section, this paper will use a rough set for feature selection, elaborate the interclass inference and analyze the bad influence of interclass inference in detail.

Rough set is a popular feature selection method. This paper uses rough sets for feature selection because the attribute reduction of rough sets removes unnecessary and unimportant features and has great advantages in feature selection. In rough set theory, data are usually stored in the form of a decision table $< U, A = C \cup D, V, f >$, where U is the universe, A is the set of attributes, C is the set of conditional attributes, D is the set of decision attributes, V is the set of all attributive values and f is the information function. For any conditional attribute subset $B \subseteq C$, there exits an equivalence relation $IND(B)$, and it is defined by $IND(B) = \{(x, y) \in U \times U | f(x, a) = f(y, a), \forall a \in B\}$. The set of all equivalence classes is denoted as U/B. For every $x \in U$, the equivalence classes of x, denoted by $[x]_B$, are defined by $[x]_B = \{y \in U | (x, y) \in IND(B)\}$. Let X be a subset of U; then, the B-lower approximation $\underline{B}(X)$ and B-upper approximation $\overline{B}(X)$ of X are defined by $\underline{B}(X) = \{x \in U | [x]_B \subseteq X\}$ and $\overline{B}(X) = \{x \in U | [x]_B \cap X \neq \varnothing\}$, respectively. The B-positive region $POS_B(D)$ in the relation $IND(D)$ is defined by $POS_B(D) = \bigcup_{X \in U/D} \underline{B}(X)$. The dependency degree $\gamma_B(D)$ of U/D on B is defined by $\gamma_B(D) = |POS_B(D)|/|U|$, where $|F|$ denotes the cardinality of set F. If B is a subset of C such that $\gamma_B(D) = \gamma_C(D)$, then B is a reduct of C [50]. Currently, there are many attribute reduction algorithms, among which one of the most common-used ones is the dependency degree-based algorithm proposed in [51]. Therefore, this paper used it for feature selection.

According to the feature selection algorithm proposed in the literature [51], a feature subset $\{a_2, a_3\}$ is obtained after attribute reduction. In this feature subset $\{a_2, a_3\}$, two rules namely $'(a_2 = low) \curlywedge (a_3 = medium) \Rightarrow (d = oil\ film\ whirl)'$ and $'(a_2 = low) \curlywedge (a_3 = low) \Rightarrow (d = oil\ film\ whirl)'$, are needed to diagnose the fault of oil film whirl. The support coefficients of two rules are both $1/10$. If the essential feature of oil film whirl, namely feature a_1 is used, only one rule $'(a_1 = high) \Rightarrow (d = oil\ film\ whirl)'$ is needed to diagnose the fault of oil film whirl, and the support coefficient of this rule increases to $2/10$. The rule obtained by using essential features and non-essential features can both realize the perfect classification of the existing instances of oil film whirl in Table 1. However, we can obtain shorter rules and larger support coefficients based on the essential feature a_1. In general, it can reflect the characteristics of data better and contribute to better generalization ability if there are fewer rules and larger support coefficients. Thus, the rule $'(a_1 = high) \Rightarrow (d = oil\ film\ whirl)'$ is a more direct reflection of the oil film whirl compared with the previous two rules and accords with well expert knowledge. However, in the process of feature selection, the essential feature a_1 of oil film whirl is deleted. Obviously, the deletion of essential features deviates from the intrinsic characteristics and thus seriously decreases the generalization ability of multi-classifiers. In this paper, the deletion of essential features that can significantly discriminate a certain class from the other classes in the feature selection of multi-classifiers is called interclass interference.

Next, we will analyze the reason for interclass interference, namely the reason why essential features are deleted. We performed feature selection through the algorithm in [50]. First, we calculate the significance of conditional attribute a_1, a_2 and a_3 in Table 1: we can obtain that $\gamma_{a_1}(d) = \frac{2}{10}$, $\gamma_{a_2}(d) = \frac{3}{10}$ and $\gamma_{a_3}(d) = 0$. The feature a_2 has the greatest significance, so it is selected first. Then, on the basis of attribute a_2, we calculate the significance of a_1 and a_3. We obtain $\gamma_{\{a_1, a_2\}}(d) = \frac{6}{10}$ and $\gamma_{\{a_2, a_3\}}(d) = \frac{8}{10}$. The feature a_3 has a greater significance, so it is selected. Because of $\gamma_{\{a_2, a_3\}}(d) = \gamma_{\{a_1, a_2, a_3\}}(d)$, the feature selection ends and the selected feature subset is $\{a_2, a_3\}$. In this process, the reason for the deletion of attribute a_1 is not because it has no contribution to classification. The essential features of the imbalance a_2 and misalignment a_3 are capable of classifying not only the instances of imbalance and misalignment but also the existing instances of oil film whirl correctly. After selecting feature a_2 and a_3, the essential feature a_1 becomes redundant in terms of the existing instances of oil film whirl and thus is deleted.

From the above analysis, the essence of interclass inference can be revealed. The phenomenon of interclass interference occurs in the feature selection process of multi-class problems. Usually, features with the ability to distinguish all classes are prioritized in

feature selection. The features that can significantly discriminate a certain class from others are called essential features of a certain class in this paper. Some essential features not only have strong classification ability for corresponding classes but also have classification ability for other classes to some extent. Thus, they are often reserved. By contrast, other essential features, which can distinguish a certain class significantly but cannot discriminate among other classes well, are often deleted as reductant features in feature selection. Once essential features are deleted, the phenomenon of interclass inference occurs. Interclass interference becomes more serious as the number of attributes increases. Although there are small training errors, there are usually large test errors because reserved features cannot reflect the essence of a certain class. The phenomenon of interclass inference seriously decreases the generalization ability of multi-classifiers. Therefore, interclass interference must be eliminated or suppressed in order to improve the generalization performance of multi-classifiers.

3. Interclass Interference Suppression Methods

3.1. All-Features-Based Approach

In the feature selection process of multi-classifiers, the essential features of a certain class are often deleted as redundant features, which causes interclass inference. Therefore, an intuitive solution to interclass inference suppression is to forgo feature selection and retain all features. This idea is intuitive and can eliminate interclass inference completely. However, this method goes against the conventional way that is widely adopted in the machine learning field, and it also seriously increases the computational cost. In the field of machine learning, many studies have shown that feature selection is an important way to improve the generalization performance of learning machines. As a result, for many machine learning methods, feature selection has become an indispensable part in the learning process [52]. Feature selection can greatly reduce the dimension of input space by removing those unimportant or redundant features, thus reducing the complexity of functions implemented by machine learning methods. Feature selection can generally improve the generalization performance of machine learning methods. Therefore, reserving all attributes are not very suitable to suppress the interclass inference.

3.2. One-Class Classifier Based Approach

One-class classifier [53] only uses the information of a certain classifier, so there is no interclass interference problem. Therefore, methods based on one-class classifiers can completely eliminate the interclass interference in multi-classifiers.

The typical example of a one-class classifier is the one-class support vector machine (SVM) [54]. One-class SVM uses a kernel function to map the original normal data to a high-dimensional space, where one-class SVM tries to find a hyperplane that enables the normal data to be as far from the origin as possible. Let the distance between the hyperplane and the origin be ρ. Then, the samples whose distance from the origin is smaller than ρ is detected as abnormal samples [54,55]. If there are m features and N training samples in the training set, let $x_i (i = 1, \ldots, N)$ denote the training data, then one-class SVM can be denoted by the following optimization problem.

$$\begin{array}{c} \min \frac{1}{2}\|w\|^2 - \rho + \frac{1}{vN}\sum\limits_{i=1}^{N}\xi_i \\ \text{s.t.}(w \cdot \varphi(x_i)) \geq \rho - \xi_i \\ \xi_i \geq 0, i = 1, \ldots, N \end{array} \quad (1)$$

where ξ_i is the slack variable, $v \in (0, 1)$ is the error rate, $\varphi(.)$ is a nonlinear mapping that is usually realized by a kernel function.

Research studies have shown that a one-class classifier can only use the information of the target class when defining the classification boundary, unlike multi-classifiers that can use the information of other classes. Therefore, in general, the performance of a one-class

classifier is difficult to make comparable to that of multi-classifiers [56]. Thus, one-class classifier-based approaches are not very suitable to suppress the interclass inference, either.

3.3. Binary-Classifier-Based Approach

There is also interclass interference in binary classifiers; that is, the essential features of a certain class are deleted as redundant features. Nevertheless, compared with multi-classifiers, the interclass interference of the binary classifiers is significantly reduced. Therefore, it is possible to solve multi-class problems by constructing binary classifiers to suppress interclass interference. Currently, two popular ways of solving multi-class problems through binary classifiers are one-versus-one (OVO) strategies [35] and one-versus-all (OVA) strategies [40].

For an M-class problem, the OVA strategy requires M binary classifiers, each of them distinguishes one class from the rest. Taking a three-class classification problem as an example, there are three classes: Class 1, Class 2 and Class 3 in the problem. OVA strategy uses three binary classifiers to solve this multi-class problem. The first binary classifier distinguishes the samples from belonging to class 1 and not belonging to class 1. The second binary classifier distinguishes the samples from belonging to class 2 and not belonging to class 2. The third binary classifier distinguishes the samples from belonging to class 3 and not belonging to class 3. This method is simple and direct in the construction of binary classifiers. However, the samples processed by each of the binary classifiers constructed by OVA are usually class-imbalanced. Unlike the OVA strategy, the OVO strategy builds binary classifiers between any two classes of the original multi-class problem. For an M-class problem, OVO needs to construct M(M-1)/2 binary classifiers, which are usually much larger than the number of classifiers required by OVA.

Although OVO requires more classifiers, each classifier built by OVO only needs to distinguish any two classes in the original multi-class problem. Thus, each classifier processes fewer instances, and the problems to be learned by each classifier are usually simpler compared with the original multi-class problem and the OVA strategy. In addition, the class-imbalanced problem existing in OVA strategy does not exist in the OVO strategy. Many practical applications have shown that the OVO construction method usually achieves the best performance when dealing with multi-class problems.

To sum up, although the all-features-based approach and the one-class classifier-based approach can eliminate interclass interference, their classification performances are usually not good. The binary classifier-based approach is a good choice to suppress the interclass inference.

4. Design of Interclass Interference Suppression Algorithms

In this section, this paper designs interclass interference suppression algorithms in order to perform the comparison experiments between the interclass inference methods better. The all-features-based approach and one-classifier based approach are easy to implement, so this section focuses on the binary-classifier-based approach only. The binary classifier-based approach is discussed in detail, and corresponding algorithms for the comparison experiments are presented in this section.

4.1. Construction of Binary Classifier

The binary-classifier-based approach decomposes multi-class problems into several binary classification problems. OVA and OVO are two commonly used strategies.

Firstly, OVA strategy is presented. For a given M-class problem, OVA strategy uses the instances belonging to class $i (i = 1, 2, \ldots, M)$ and the instances not belonging to class i to construct M binary classifiers. In this paper, other classes except class i are called the negative class of class i, denoted as class $\bar{i}$. Algorithm 1 shows the detailed procedure of experiments based on OVA in this paper.

Secondly, the OVO strategy is presented. For a given M-class problem, OVO strategy uses to construct M(M-1)/2 binary classifiers respectively. Algorithm 2 shows the detailed procedure of experiments based on OVO strategy in this paper.

For OVO and OVA method, there are three decision strategies. The first one is based on the voting of classification results. The second one is based on the voting of a certain coefficient. The third one is based on the voting of the support coefficient. For voting based on the support coefficient, the OVA method can use the support coefficient of either the positive class or the negative class.

Algorithm 1 Binary classifier construction procedure based on OVA strategy

Input: the original dataset

Output: M binary classifiers; the corresponding feature subset $iRedu$, rule set $iRuleset$ and support coefficient $iSupp$ of M binary classifiers; the average number of attributes $aveNum\_Redu$; the overall feature subsets $used\_Redu$; the overall number of rules Num; average support coefficient $aveSupp$; the average rule length $aveLen$; the overall number of rules $\overline{Num}$ for negative class; average support coefficient $\overline{aveSupp}$ for negative class and the average rule length $\overline{aveLen}$ for negative class.

Begin

$i = 0$;

while $i \leq M$; //M is the number of classes in multi-class problems

1. Mark all instances not belonging to class i as class $\bar{i}$ and obtain the dataset $i\_Dataset$ of i-th classifier. // class $\bar{i}$ is denoted as the negative class of class i here.
2. Obtain the feature subset i_Redu of $i\_Dataset$ via attribute reduction and denote the number of attributes in feature subset as $iNum\_Redu$.
3. Extract rules from feature subset, and denote the rule set of i-th classifier as $iRuleset$
4. Compute the support coefficient of each rule in rule set $iRuleset$ and denote the set of support coefficients as $iSupp$.
5. Compute the rule number $iNum$, the sum of support coefficient $iSum\_Supp$ and the sum of rule length $iSum\_Len$ of the class i.
6. Compute the rule number $\overline{iNum}$, the sum of support coefficient $\overline{iSum\_Supp}$ and the sum of rule length $\overline{iSum\_Len}$ of the class $\bar{i}$.
7. $i = i + 1$;

end {while}

$aveNum\_Redu \leftarrow \underset{i}{sum}(iNum\_Redu)/M$ // the average number of attributes

$used\_Redu \leftarrow \underset{i}{\cup} iRedu$ // used feature subset for M-class problem

$Num \leftarrow \underset{i}{sum}(iNum)$ // the number of rules for M-class problem

$aveSupp \leftarrow \underset{i}{sum}(iSum\_Supp)/Num$ // average support coefficient of rules

$aveLen \leftarrow \underset{i}{sum}(iSum\_Len)/Num$ // average rule length

$\overline{Num} \leftarrow \underset{i}{sum}(\overline{iNum})$ // the number of rules for negative class

$\overline{aveSupp} \leftarrow \underset{i}{sum}(\overline{iSum\_Supp})/\overline{Num}$ // average support coefficient of negative class

$\overline{aveLen} \leftarrow \underset{i}{sum}(\overline{iSum\_Len})/\overline{Num}$ // average rule length for negative class

End

Algorithm 2 Binary classifier construction procedure based on OVO strategy

Input: the original dataset
Output: $M(M-1)/2$ binary classifiers; the corresponding feature subset $ijRedu$; rule set $ijRuleset$; support coefficient $ijSupp$ of M binary classifiers; the average number of attributes $aveNum\_Redu$; the overall feature subsets $used\_Redu$; the equivalent overall number of rules Num; average support coefficient $aveSupp$; average rule length $aveLen$.
Begin
for each class i in the original class problem ($1 \leq i \leq M-1$)
for each class i in the original class problem ($i+1 \leq j \leq M$)
1. Search all instances of class i and class j in the original dataset and construct new dataset $ij\_Dataset$;
2. Perform feature selection for dataset $ij\_Dataset$ through rough set, obtain the feature subset ijRed$u$ and denote the number of attributes in feature subset ijRed$u$ as $ijNum_$Red$u$.
3. Extract rules and obtain the rule set $ijRuleset$ and calculate the rule's support set $ijSupp$.
4. For class i, calculate the rule number $iNum$, the sum of the rule support $iSum\_Supp$ and the sum of the rule length $iSum\_Len$. Then store $iNum$, $iSum\_Supp$ and $iSum\_Len$ in the hash table Sum_Num, Sum_Supp and Sum_Len respectively with the key being class i.
5. For class j, calculate the rule number $jNum$, the sum of support coefficient $jSum\_Supp$ and the sum of rule length $jSum\_Len$. Then store $jNum$, $jSum\_Supp$, $jSum\_Len$ in the above hash table Sum_Num, Sum_Supp and Sum_Len respectively with the key being class j.
end {for}
end {for}
$aveNum_$Red$u \leftarrow \underset{i,j}{sum}(ijNum_Redu)\ (M(M-1)/2)$.
$used_$Red$u \leftarrow \underset{i,j}{\cup} ijRedu$. // actually used reduction subset
$Num \leftarrow \underset{i}{sum}(Sum\_Num\ (M-1))$. // equivalent overall rule number
$aveSupp \leftarrow \underset{i}{sum}(Sum\_Supp)\ \underset{i}{sum}(Sum\_Num)$. // average support coefficient
$aveLen \leftarrow \underset{i}{sum}(Sum\_Len)\ \underset{i}{sum}(Sum\_Num)$. // average rule length
End

4.2. Multi-Classifier Unified Collaborative Decision Algorithm

Algorithms for constructing binary classifiers based on OVA and OVO have been elaborated in Algorithm 1 and Algorithm 2. In order to classify new instances of multi-class problems, it is necessary to design a collaborative decision algorithm for these binary classifiers. Algorithm 3 gives a unified classifier collaborative decision algorithm. In Algorithm 3, the classification decision scoring indexes of each classifier can be classified in terms of support coefficients, certain coefficients and classification decision results. Different scoring indicators are chosen to get different collaborative decision-making strategies. It is necessary to explain that, for the binary classifiers based on OVA, the decision-making algorithm can make decisions based on either the classification decision score of each category or the negative class of each category. Meanwhile, we can still use the unified storage structure and algorithm described in Figure 1 to classify decision; the difference is that the classification decision scoring index value needs to be negative, and the algorithm is based on the absolute minimum principle of categorical decision scoring of negative categories to classify new model instances. The classification decision based on each class of negative classes is usually based on the majority of the class rules, so the rules usually have good statistical properties.

Algorithm 3 Multi-classifier unified collaborative decision algorithm.

Input: two types of classifiers for M class problems (iRedu, $iRuleset$ and $iSupp$) and new instance x
Output: the final classification result of x
Begin
for the i classifier in all binary classifiers do
1. According to the relevant classification decision method, the classifier is used to classify x;
2. According to the classification decision scoring index, the classification decision score of the classifier is stored in the corresponding classification decision of the unified structure of Figure 1 through the accumulative method.
end{for}
Selecting the largest category of cumulative classification decision score from the structure as the final classification result of x
End

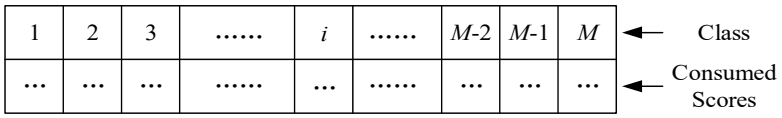

Figure 1. Store Structure of Consumed Score for the M-Class Problem.

The binary-classifier-based approach decomposes multi-class problems into several binary classification problems. The OVO strategy needs M(M-1)/2 binary classifiers, and the OVA strategy needs M classifiers. The classification of a new instance is made by voting of these binary classifiers. The OVO strategy has three voting methods, including classification results, certain coefficient and support coefficient of each binary classifier. The OVA method has one method besides the above three methods, namely voting based on the support coefficient of negative classes of each binary classifier. The consumed score of each class is stored in the structure shown in Figure 1 after selecting one voting method. The final classification decision is made in terms of the class with the highest consumed score.

5. Experiments

5.1. Configurations of Experiments

In Section 4, binary classifiers construction algorithms based on OVO and OVA are presented, and three different classifier collaborative decision strategies can be selected for each algorithm. In order to verify the effect of these methods on interclass inference, this paper carries out 10-fold cross-validation experiments [57] of 14 UCI datasets [58]. Table 2 summarizes the information of these datasets. In this section, this paper compares the classification accuracy, the number of attributes, the number of rules, the length of rules and the support coefficient of various classification algorithms to compare the effect of these methods on interclass inference.

5.2. Comparison Among Different Interclass Interference Suppression Methods

In this section, this paper compares the OVA method and the OVO method with conventional multi-classifiers. This paper uses the algorithm in the literature [51] for feature selection and to classify through LEM2 algorithm [59], a common-used rule-based classification algorithm. For the conventional algorithm, feature selection is made once only. For the OVA and OVO methods, we need to construct M and $M(M-1)/2$ classifiers, respectively, and we need to make feature selection for each constructed classifier. The classification accuracy obtained from the various methods is given in Table 3. OVA_V, OVA_C, OVA_PS and OVA_NS represent the strategies of voting based on classification results, voting based on certain coefficients of rule, voting based on the support coefficient of positive classes and voting based on the support coefficient of negative classes for the OVA method, respectively. Meanwhile, OVO_V, OVO _C and OVO _S represent the strategies of voting based on classification results, voting based on certain coefficients of rule and voting

based on the support coefficient for the OVO method, respectively. This paper performs 10-fold cross-validation to select the parameters of these algorithms and compares the interclass inference suppression effects. Specifically, the data are randomly divided into ten equal parts. Each unique part is selected as the test set, and the other nine parts are used as the training set. Thus, ten experiments are performed in total. The classification accuracy of 10-fold cross-validation is the mean value of the ten experiments. Corresponding 10-fold cross-validation classification accuracies of these methods are shown in Table 3. From Table 3, the following phenomena can be observed.

Table 2. Datasets used for experiments.

No.	Name	Size	Conditional Attribute Number	Number of Classes
1	zoo	101	16	7
2	lymphography	148	18	4
3	wine	178	13	3
4	flags	194	28	8
5	autos	205	23	6
6	machine	209	7	8
7	images	210	19	7
8	glass	214	9	6
9	audiology	226	69	24
10	heart	303	13	5
11	solar	323	10	3
12	soybean	683	35	19
13	vehicle	846	18	4
14	anneal	898	38	5

(1) Based on the OVA and OVO methods, the classification accuracy is significantly improved compared with the conventional algorithm. This shows that the binary-classifier-based method can effectively suppress the interclass interference in multi-classifiers.

(2) The OVO method obtains the best classification accuracy among OVO, OVA and conventional algorithms. Compared with conventional algorithms, the OVA method can also improve the classification accuracy, but the improvement is not obvious. This shows that the OVO method can suppress the interclass inference better than the OVA method. The reason for this is that the OVA method usually causes the class-imbalanced problem in classification.

(3) In all kinds of decision strategies, the strategy of voting based on classification results obtains the optimal accuracy, and the strategy of voting based on certain coefficient of rules obtains the suboptimal accuracy. The strategy of voting based on the support coefficient of rules is the worst. For the OVA method, classification based on negative classes obtains a worse classification accuracy than that based on positive classes. This shows that voting based on classification results can be chosen as the optimal collaborative decision strategy.

(4) The OVO-based binary classification method requires building more classifiers than the OVA-based binary classification method and thus usually costs more computational time. Literature [60] points out the computational burden of the two methods. For a multi-class problem with M classes, OVO requires $M(M-1)/2$ base binary classifiers and the computational complexity can be regarded as $O(M^2)$. By contrast, OVA requires M base binary classifiers, and the computational complexity can be regarded as $O(M)$. Although the OVO-based binary classification method costs more time, OVO can obtain better classification accuracy and thus better interclass inference suppression performance than the OVA method and is more suitable for the case where users require high classification accuracy.

Table 3. Classification accuracy of OVA and OVO approach.

Dataset	Conventional Algorithm	OVA				OVO		
		OVA_V	OVA_C	OVA_PS	1vR_NS	OVO_V	OVO_C	OVO_S
zoo	0.9400	0.9509	0.9409	0.9509	0.9309	0.9509	0.9509	0.9009
lymphography	0.8186	0.8252	0.8324	0.8319	0.8319	0.8257	0.8257	0.7786
wine	0.9389	0.9444	0.9389	0.9441	0.9219	0.9549	0.9549	0.9157
flags	0.5937	0.5884	0.5837	0.5984	0.5982	0.6495	0.6287	0.4700
autos	0.7438	0.7731	0.7833	0.7636	0.7590	0.7633	0.7586	0.5221
machine	0.6552	0.6267	0.6505	0.6457	0.6410	0.6886	0.6886	0.5074
images	0.8667	0.8667	0.8714	0.8429	0.8524	0.8667	0.8667	0.6238
glass	0.6955	0.6771	0.6768	0.6675	0.5781	0.6823	0.6777	0.5799
audiology	0.7615	0.7447	0.7490	0.7316	0.7708	0.7800	0.7800	0.5002
heart	0.5216	0.5246	0.5246	0.5244	0.5051	0.5544	0.5443	0.5841
solar	0.8671	0.8609	0.8607	0.8578	0.8483	0.8761	0.8731	0.7865
soybean	0.8462	0.8960	0.9033	0.8843	0.8814	0.9254	0.9298	0.4802
vehicle	0.6631	0.6855	0.7020	0.6843	0.6962	0.7175	0.7210	0.6192
anneal	1.0000	1.0000	1.0000	1.0000	1.0000	0.9989	0.9989	0.9232
Mean	0.7794	0.7832	0.7870	0.7805	0.7725	0.8024	0.7999	0.6565

Next, we will analyze the reasons for the performance improvement through some basic evaluation indexes of classifiers. Tables 4 and 5 give the number of selected features, rule number, rule length and rule support coefficient of various methods. From Tables 4 and 5, we can obtain the following results:

(1) The number of attributes and rules obtained by the OVO interclass interference suppression method is significantly smaller than that of the conventional algorithm. Meanwhile, the rule length is shorter, and the support coefficient is larger. Obviously, OVO method can reflect the intrinsic characteristics better and obtain better generalization ability.

(2) Compared with the OVO approach, the above-mentioned indexes obtained by the OVA approach are similar to the conventional multi-classifiers, which explains why the performance improvement is not obvious to some extent. In addition, although the negative-classes-based classification obtains larger support coefficients and shorter rules than the positive-classes-based classification, its classification accuracy is not ideal since it is an indirect decision-making strategy.

Table 4. Numbers of attributes and rules obtained by different methods.

Dataset	Numbers of Attributes			Numbers of Rules			
	Conventional Algorithm	OVA	OVO	Convention	OVA		OVO
					P	N	
zoo	4.9000	2.4571	1.0857	12.2000	10.0000	20.7000	7.4500
lymphography	6.0000	3.7000	2.1000	34.9000	33.3500	39.2500	15.7333
wine	4.0000	3.5667	2.5000	13.5000	12.5000	16.5000	9.1500
flags	8.8000	5.1625	2.7571	73.8000	76.2500	144.2500	38.9857
autos	9.2000	5.5667	2.9600	51.1000	52.3500	82.9500	28.6400
machine	6.7000	3.7125	1.7250	37.0000	35.0000	69.1000	17.7286
images	6.3000	3.2143	1.7048	28.3000	27.6500	53.6500	16.2500
glass	6.8000	5.1667	3.0467	32.4000	29.9000	50.6000	18.5400
audiology	13.3000	2.9716	1.1023	58.1000	55.3000	122.4000	28.7445
heart	9.8000	9.1800	8.0100	98.2000	96.3000	135.5000	59.8000
solar	9.0000	7.9000	6.5667	52.1000	52.1000	63.4000	37.5500
soybean	11.3000	3.0421	1.1234	115.7000	91.0500	151.1500	28.5278
vehicle	14.2000	11.5250	8.8500	181.5000	181.9000	207.8000	91.6333
anneal	3.0000	1.2000	1.0000	7.0000	5.0000	8.0000	5.5250
Mean	8.0929	4.8832	3.1808	56.8429	54.1893	83.2321	28.8756

Table 5. Length and support coefficient of rules obtained by different methods.

Dataset	Rule Length				Rule Support Coefficient			
	Conventional Algorithm	OVA		OVO	Convention	OVA		OVO
		P	N			P	N	
zoo	2.3088	2.0904	1.1108	1.0402	0.0823	0.1024	0.3679	0.4726
lymphography	2.5843	2.5264	2.3725	2.0472	0.0457	0.0471	0.0919	0.1438
wine	2.1845	2.1782	1.6184	1.5169	0.1070	0.1027	0.1755	0.2150
flags	2.8297	2.7584	1.8370	1.6013	0.0175	0.0155	0.0739	0.1219
autos	2.8248	2.6671	1.9757	1.7014	0.0246	0.0210	0.0831	0.1202
machine	2.8934	2.9594	2.0499	1.6984	0.0334	0.0266	0.1263	0.2206
images	2.5950	2.5412	1.6086	1.3716	0.0423	0.0370	0.1373	0.2281
glass	3.1033	3.1045	2.1525	1.9460	0.0335	0.0314	0.1433	0.1732
audiology	3.2776	2.9791	1.4881	1.0709	0.0223	0.0198	0.2522	0.4143
heart	4.8349	4.9566	3.8769	3.5863	0.0155	0.0125	0.0657	0.0649
solar	3.3148	3.4920	3.0869	2.9350	0.0272	0.0222	0.0442	0.0503
soybean	4.1696	3.8603	2.1925	1.2280	0.0105	0.0112	0.1676	0.3356
vehicle	4.4220	4.5494	3.9427	3.6300	0.0087	0.0069	0.0266	0.0287
anneal	1.7143	1.2000	1.0000	1.0000	0.1429	0.2000	0.5888	0.4526
Mean	3.0755	2.9902	2.1652	1.8838	0.0438	0.0469	0.1674	0.2173

5.3. Performance of the All-Features-Based Approach

When dealing with multi-class problems, deletion of essential features leads to interclass interference. If all features are reserved, interclass inference can be eliminated. Thus, this paper evaluates the effect of this method on interclass interference through experiments.

Table 6 shows the classification accuracy, number of rules, rule length and the support coefficient of rules obtained by the all-features-based method and the OVO method. In Table 6, CV_A represents the all-features-based methods and OVO_VA represents the method that reserves all features and uses OVO strategy and makes classification by voting based on the classification results of each classifier. By comparing it with Tables 3–5, the following phenomena are observed.

(1) By reserving all features, the classification accuracy of multi-class problems is obviously improved. Further comparison shows that all-features-based method obtains fewer rules and a larger support coefficient than the conventional algorithm. This shows that each class can be expressed by its own essential features and that the intrinsic characteristics of data can be reflected better when reserving all features. This can explain the why all-features-based method obtains better classification accuracy to some extent.

(2) The OVO_VA method and OVO_V method has few differences in classification accuracy. Thus, reserving all features has little effect on the classification accuracy of OVO method. This is because the OVO method itself can make use of the essential features to express the knowledge and suppress interclass interference.

(3) The all-features-based method still obtains worse classification accuracy than the OVO_V method shown in Table 3. Compared with the conventional algorithm and the all-features-based algorithm, the OVO method extracts simpler and clearer classification knowledge and stronger obtained rules. Thus, the OVO method obtains better generalization ability.

Table 6. Classification accuracy obtained by the method based on all features.

Dataset	Classification Accuracy		Number of Rules		Rule Length		Support Coefficient	
	CV_A	OVO_VA	CV_A	OVO_VA	CV_A	OVO_VA	CV_A	OVO_VA
zoo	0.9518	0.9518	9.3000	7.3500	2.2239	1.0544	0.1110	0.4798
lymphography	0.8319	0.8452	25.6000	12.9000	2.9551	2.2118	0.0744	0.2199
wine	0.9497	0.9552	10.1000	8.1500	2.1206	1.6022	0.1961	0.3247
flags	0.6187	0.6134	57.1000	31.3429	3.3845	1.9894	0.0266	0.1994
autos	0.7640	0.7933	48.2000	23.8800	2.8919	1.8957	0.0285	0.1761
machine	0.6505	0.6743	37.2000	16.7000	2.8934	1.7571	0.0335	0.2673
images	0.8857	0.8857	25.0000	13.5167	2.5977	1.4600	0.0488	0.2854
glass	0.6955	0.6677	32.3000	18.0200	3.1229	1.9913	0.0336	0.1792
audiology	0.7972	0.8241	40.7000	24.6133	3.6348	1.1128	0.0362	0.4837
heart	0.5216	0.5283	98.8000	58.6000	4.8474	3.5952	0.0152	0.0731
solar	0.8580	0.8672	51.6000	36.3500	3.2850	2.9984	0.0277	0.0531
soybean	0.9165	0.9370	61.7000	22.6667	4.0635	1.2661	0.0207	0.4276
vehicle	0.6760	0.7151	177.2	86.2667	4.4825	3.6881	0.0092	0.0381
anneal	1.0000	1.0000	5.0000	5.0000	1.2000	1.0000	0.2000	0.5000
Mean	0.7941	0.8042	48.5571	26.0969	3.1217	1.9730	0.0615	0.2648

Compared with conventional algorithms, the all-features-based algorithm can effectively suppress interclass interference and achieve better generalization ability in multi-class problems. However, the method also preserves redundant features, leading to more computational cost. Table 7 shows the number of attributes obtained by the conventional method, the all-features-based method and the OVO method when dealing with multi-class problems. It is observed that the number of attributes used by the all-features-based method is significantly higher than that of the conventional method. Although the number of attributes obtained by the OVO method is also larger than by the conventional algorithm, it uses much fewer attributes than all-features-based method. This shows that the OVO method can not only effectively suppress the interclass interference of multi-class problems but also delete redundant features and avoid the additional cost caused by these features. Therefore, when dealing with multi-class problems, the OVO method is a better choice to suppress interclass inference than the all-features-based method.

Table 7. Numbers of attributes obtained by the method based on all features and other methods.

Dataset	Conventional Algorithm	CV_A	OVO_V
zoo	4.9000	16	8.7000
lymphography	6.0000	18	8.1000
wine	4.0000	13	5.6000
flags	8.8000	28	14.9000
autos	9.2000	23	12.5000
machine	6.7000	7	7.0000
images	6.3000	19	11.4000
glass	6.8000	9	7.0000
audiology	13.3000	69	23.8000
heart	9.8000	13	10.0000
solar	9.0000	10	8.9000
soybean	11.3000	35	24.9000
vehicle	14.2000	18	16.1000
anneal	3.0000	38	4.9000
Mean	8.0929	22.5714	11.7000

5.4. Comparison Among the One-Class Classifier and Binary Classifier-Based Approach

The method based on one-class classifiers can completely eliminate the class interference existing in multi-classifiers. However, a one-class classifier only uses the information of the target class in the definition of the classification boundary. Many studies show that

the performance of one-class classifiers is usually worse than that of multi-classifiers. In order to confirm that the OVO method is the best choice to suppress interclass interference, we used a binary support vector machine (SVM) based on the OVO method and one-class SVM [61] to carry out a comparison experiment. In the experiment, we did not carry out feature selection. Radial basis function was used for binary SVM and one-class SVM. We first compared the classification accuracy of binary and one-class SVM under different parameters to determine their optimal parameters and obtained their best classification accuracy. Then, the best classification accuracy of the two methods was compared to evaluate the performance of binary and one-class SVM. Results are shown in Table 8.

Table 8. Comparison between classification accuracies obtained by binary SVM and one-class SVM.

Dataset	Binary SVM (OVO)			One-Class SVM		
	C = 1	C = 100	C = 1000	ν = 0.001	ν = 0.01	ν = 0.3
zoo	0.9209	0.9609	0.9609	0.9418	0.8718	0.8218
lymphography	0.7852	0.8457	0.8457	0.7233	0.7100	0.7171
wine	0.4611	0.4892	0.4892	0.5062	0.5229	0.5173
flags	0.3766	0.3768	0.3768	0.2226	0.2637	0.2639
autos	0.3469	0.3469	0.3469	0.0433	0.1705	0.0690
machine	0.6031	0.6031	0.6031	0.1533	0.1629	0.1819
images	0.3048	0.3190	0.3190	0.2905	0.2952	0.2952
glass	0.6675	0.7000	0.6855	0.5461	0.5506	0.5240
audiology	0.5045	0.8198	0.8198	0.3974	0.4334	0.4557
heart	0.5412	0.5412	0.5412	0.1218	0.1218	0.1218
solar	0.8855	0.8640	0.8483	0.7925	0.8139	0.6469
soybean	0.9398	0.9428	0.9355	0.8536	0.8653	0.8404
vehicle	0.2895	0.2989	0.2989	0.3002	0.3013	0.3013
anneal	0.9254	0.9599	0.9599	0.8508	0.8530	0.8408
Mean	0.6109	0.6477	0.6451	0.4817	0.4955	0.4712

From Table 8, we can draw the following conclusions. When we take C = 100 in the experiment, SVM based on the OVO method acquires the best performance. When we take ν = 0.01, one-class SVM obtains the better performance. By comparing the best performance obtained by the two methods, it is observed that the classification accuracy of the binary SVM is significantly better than one-class SVM, which is consistent with the previous research results of one-class classifiers. The performance of one-class classifiers is usually worse than binary classifiers, and it is difficult to implement one-class classification algorithm in many multi-classifiers. Therefore, the interclass interference suppression method based on binary classifiers is the best choice to suppress the interclass interference of multi-classifiers.

6. Conclusions

The generalization ability of classifiers is a crucial problem in pattern recognition. In multi-classifiers, there is a special phenomenon that essential features with the ability to discriminate a certain class from others are often deleted in feature selection. This phenomenon seriously decreases the classification accuracy of multi-classifiers. To address this problem, this paper called this phenomenon interclass inference, analyzed its reasons in detail and summarized three interclass inference suppression methods including all-features-based method, the one-class classifier-based method and the binary-classifier-based method. By comparing the three methods in 14 UCI datasets through 10-fold cross-validation, the following conclusions can be drawn.

Firstly, the concept of interclass inference is proposed and the essence of interclass inference is revealed. The deletion of essential features leads to interclass inference.

Secondly, the three methods can all improve the classification accuracy to some extent when they are compared with the conventional algorithm. This shows that suppressing the interclass inference is an effective method to improve the generalization ability of

multi-classifiers. The binary classifier-based method can suppress the interclass inference best and obtain the best generalization ability.

Thirdly, this paper compares the suppression effects of two binary classifier-based methods, including one-versus-one (OVO) and one-versus-all (OVA) on interclass inference. OVO method obtains better classification accuracy than the OVA method. Thus, it is the best method to suppress the interclass inference. By studying the interclass inference and its suppression methods, the generalization ability of multi-classifiers is significantly improved.

Author Contributions: Conceptualization, M.B. and J.L.; writing, N.J.; methodology, M.B., X.L., Y.W. and R.C.; supervision, D.Y. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China under Grant No. 51976042, National Science and Technology Major Project of China under Grant No. 2017-I-0007-0008, National Science and Technology Major Project of China under Grant No. 2017-V-0005-0055, National Key R&D Program of China No. 2017YFB0902100 and Science and Technology on Thermal Energy and Power Laboratory Foundation No. TPL2017CA010.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable suggestions to refine this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sugaya, Y.; Sugibuchi, K.; Omachi, S. Effectiveness of Integration of Multiple Classification Methods within the AdaBoost Framework (Special Issue on Computer Vision and Applications). *IIEE J. Trans. Image Electron. Vis. Comput.* **2015**, *3*, 174–184.
2. Zhao, W.; Wang, Z.; Ma, J.; Li, L. Fault Diagnosis of a Hydraulic Pump Based on the CEEMD-STFT Time-Frequency Entropy Method and Multiclass SVM Classifier. *Shock Vib.* **2016**, *8*. [CrossRef]
3. Hang, J.; Zhang, J.; Cheng, M. Application of multi-class fuzzy support vector machine classifier for fault diagnosis of wind turbine. *Fuzzy Sets Syst.* **2016**, *297*, 128–140. [CrossRef]
4. Zhang, B.; Yang, Y.; Chen, C.; Yang, L.; Han, J.; Shao, L. Action Recognition Using 3D Histograms of Texture and A Multi-class Boosting Classifier. *IEEE Trans. Image Process.* **2017**, *26*, 4648–4660. [CrossRef] [PubMed]
5. Nan, D.; Xu, Z.; Bian, S.Q. Face Recognition Based on Multi-classifier Weighted Optimization and Sparse Representation. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2013**, *6*, 423–436.
6. Qian, H.; Mao, Y.; Xiang, W.; Wang, Z. Recognition of human activities using SVM multi-class classifier. *Pattern Recognit. Lett.* **2010**, *31*, 100–111. [CrossRef]
7. Wick, P.; Louw-Gaume, A.E.; Kucki, M.; Krug, H.F.; Kostarelos, K.; Fadeel, B.; Dawson, K.A.; Salvati, A.; Vazquez, E.; Ballerini, L.; et al. Classification framework for graphene-based materials. *Angew. Chem. Int. Ed.* **2014**, *53*, 7714–7718. [CrossRef]
8. De Melo, G.; Varde, A.S. Scalable Learning Technologies for Big Data Mining. In Proceedings of the 20th International Conference on Database Systems for Advanced Applications, DASFAA, Hanoi, Vietnam, 20–23 April 2015; Springer: Berlin/Heidelberg, Germany, 2015.
9. Varde, A.S.; Ma, S.; Maniruzzaman, M.D.; Brown, D.C.; Rundensteiner, E.A.; Sisson, R.D., Jr. Comparing mathematical and heuristic approaches for scientific data analysis. *AI EDAM Artif. Intell. Eng. Des. Anal. Manuf.* **2008**, *22*, 53. [CrossRef]
10. Burnap, P.; Colombo, G.; Amery, R.; Hodorog, A.; Scourfield, J. Multi-class machine classification of suicide-related communication on Twitter. *Online Soc. Netw. Media* **2017**, *2*, 32–44. [CrossRef]
11. Tang, L.; Tian, Y.; Pardalos, P.M. A novel perspective on multiclass classification: Regular simplex support vector machine. *Inf. Sci.* **2019**, *480*, 324–338. [CrossRef]
12. Karthikeyan, D.; Varde, A.S.; Wang, W. Transfer learning for decision support in Covid-19 detection from a few images in big data. In Proceedings of the IEEE Big Data Conf., Atlanta, GA, USA, 10 December2020.
13. Ang, J.H.; Guan, S.U.; Tan, K.C.; Al-Mamun, A. Interference-less neural network training. *Neurocomputing* **2008**, *71*, 3509–3524. [CrossRef]
14. Basavaraju, P.; Varde, A.S. Supervised learning techniques in mobile device apps for Androids. *ACM Sigkdd Explor. Newsl.* **2017**, *18*, 18–29. [CrossRef]

15. Har-Peled, S.; Roth, D.; Zimak, D. *Constraint Classification: A New Approach to Multiclass Classification. International Conference on Algorithmic Learning Theory*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 365–379.
16. Amit, Y.; Fink, M.; Srebro, N.; Ullman, S. Uncovering Shared Structures in Multiclass Classification. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; pp. 17–24.
17. Tewari, A.; Bartlett, P.L. On the consistency of multiclass classification methods. *J. Mach. Learn. Res.* **2007**, *8*, 1007–1025.
18. Bai, M.; Liu, J.; Chai, J.; Zhao, X.; Yu, D. Anomaly detection of gas turbines based on normal pattern extraction. *Appl. Eng.* **2020**, *166*, 114664. [CrossRef]
19. Lei, Y.; Yang, B.; Jiang, X.; Jia, F.; Li, N.; Nandi, A.K. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mech. Syst. Signal Process.* **2020**, *138*, 106587. [CrossRef]
20. Gandhe, K.; Varde, A.S.; Du, X. Sentiment Analysis of Twitter Data with Hybrid Learning for Recommender Applications. In Proceedings of the 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 8–10 November 2018; IEEE: New York, NY, USA, 2018; pp. 57–63.
21. Aly, M. Survey on multiclass classification methods. *Neural Netw.* **2005**, *19*, 1–9.
22. Pawara, P.; Okafor, E.; Groefsema, M.; He, S.; Schomaker, L.; Wiering, M.A. One-vs-One classification for deep neural networks. *Pattern Recognit.* **2020**, *108*, 107528. [CrossRef]
23. Katuwal, R.; Suganthan, P.N.; Zhang, L. An ensemble of decision trees with random vector functional link networks for multi-class classification. *Appl. Soft Comput.* **2018**, *70*, 1146–1153. [CrossRef]
24. Wu, K.; Zheng, Z.; Tang, S. BVDT: A Boosted Vector Decision Tree Algorithm for Multi-Class Classification Problems. *Int. J. Pattern Recognit. Artif. Intell.* **2017**, *31*, 1750016. [CrossRef]
25. Vluymans, S.; Fernández, A.; Saeys, Y.; Cornelis, C.; Herrera, F. Dynamic affinity-based classification of multi-class imbalanced data with one-versus-one decomposition: A fuzzy rough set approach. *Knowl. Inf. Syst.* **2017**, *56*, 1–30. [CrossRef]
26. Vluymans, S.; Cornelis, C.; Herrera, F.; Saeys, Y. Multi-label classification using a fuzzy rough neighborhood consensus. *Inf. Sci.* **2018**, *433*, 96–114. [CrossRef]
27. Liu, J.; Bai, M.; Jiang, N.; Yu, D. Structural risk minimization of rough set-based classifier. *Soft Comput.* **2020**, *24*, 2049–2066. [CrossRef]
28. Liu, J.; Bai, M.; Jiang, N.; Yu, D. A novel measure of attribute significance with complexity weight. *Appl. Soft Comput.* **2019**, *82*, 105543. [CrossRef]
29. Melin, P.; Amezcua, J.; Valdez, F.; Castillo, O. A new neural network model based on the LVQ algorithm for multi-class classification of arrhythmias. *Inf. Sci.* **2014**, *279*, 483–497. [CrossRef]
30. Zhang, Z.L.; Luo, X.G.; García, S.; Herrera, F. Cost-Sensitive back-propagation neural networks with binarization techniques in addressing multi-class problems and non-competent classifiers. *Appl. Soft Comput.* **2017**, *56*, 357–367. [CrossRef]
31. Tang, B.; Kay, S.; He, H. Toward optimal feature selection in naive Bayes for text categorization. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 2508–2521. [CrossRef]
32. Zhang, S.; Li, X.; Zong, M.; Zhu, X.; Cheng, D. Learning k for knn classification. *ACM Trans. Intell. Syst. Technol.* **2017**, *8*, 1–19. [CrossRef]
33. Li, K.; Wu, Y.; Nan, Y.; Li, P.; Li, Y. Hierarchical multi-class classification in multimodal spacecraft data using DNN and weighted support vector machine. *Neurocomputing* **2017**, *259*, 55–65. [CrossRef]
34. Wang, G.S.; Ren, Q.H.; Su, Y.Z. The Interference Classification and Recognition Based on SF-SVM Algorithm. In Proceedings of the 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), GuangZhou, China, 6–8 May 2017; IEEE: New York, NY, USA, 2017; pp. 835–841.
35. Knerr, S.; Personnaz, L.; Dreyfus, G. *Single-Layer Learning Revisited: A Stepwise Procedure for Building and Training a Neural Network. Neurocomputing*; Springer: Berlin/Heidelberg, Germany, 1990; pp. 41–50.
36. Clark, P.; Boswell, R. *Rule Induction with CN2: Some Recent Improvements. European Working Session on Learning*; Springer: Berlin/Heidelberg, Germany, 1991; pp. 151–163.
37. Anand, R.; Mehrotra, K.; Mohan, C.K.; Ranka, S. Efficient classification for multiclass problems using modular neural networks. *IEEE Trans. Neural Netw.* **1995**, *6*, 117–124. [CrossRef]
38. Galar, M.; Fernández, A.; Barrenechea, E.; Herrera, F. DRCW-OVO: Distance-based relative competence weighting combination for One-versus-one strategy in multi-class problems. *Pattern Recognit.* **2015**, *48*, 28–42. [CrossRef]
39. Galar, M.; Fernández, A.; Barrenechea, E.; Bustince, H.; Herrera, F. Dynamic classifier selection for One-versus-one strategy: Avoiding non-competent classifiers. *Pattern Recognit.* **2013**, *46*, 3412–3424. [CrossRef]
40. Dinh, V.; Ho, L.S.T.; Cuong, N.V.; Nguyen, D.; Nguyen, B.T. Learning from Non-Iid Data: Fast Rates for the One-Versus-All Multiclass Plug-in Classifiers. In *Proceedings of the International Conference on Theory and Applications of Models of Computation*; Springer: Cham, Germany, 2015; pp. 375–387.
41. Rebetez, J.; Perez-Uribe, A. Indoor Activity Recognition by Combining One-vs.-All Neural Network Classifiers Exploiting Wearable and Depth Sensors. In *Proceedings of the International Conference on Artificial Neural Networks: Advences in Computational Intelligence*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 216–223.
42. Shen, X.; Wang, L. Generalization error for multi-class margin classification. *Electron. J. Stat.* **2007**, *1*, 307–330. [CrossRef]

43. Eiadon, M.; Pipanmaekaporn, L.; Kamonsantiroj, S. Mining discriminative class codes for multi-class classification based on minimizing generalization errors. First International Workshop on Pattern Recognition. *Int. Soc. Opt. Photonics* **2016**, *10011*, 100111D.
44. Lei, Y.; Dogan, Ü.; Zhou, D.; Kloft, M. Generalization error bounds for extreme multi-class classification. *CoRR* **2017**, *abs/1706.09814*.
45. Kantavat, P.; Kijsirikul, B.; Songsiri, P.; Fukui, K.-I.; Numao, M. Efficient Decision Trees for Multi–Class Support Vector Machines Using Entropy and Generalization Error Estimation. *Int. J. Appl. Math. Comput. Sci.* **2018**, *28*, 705–717. [CrossRef]
46. Dhifli, W.; Diallo, A.B. Galaxy-X: A Novel Approach for Multi-class Classification in an Open Universe. *arXiv* **2015**, arXiv:1511.00725.
47. Liu, S.; Chen, J.; Wang, F.; Feng, Y.X.; Hong-Bo, G.U.; Han, J.F.; Wu, J.B. Analysis and Treatment of Oil Whirl on 1000MW Ultra-supercritical Unit. *Turbine Technol.* **2010**, *52*, 373–376.
48. Dong, X. Vibration Analysis and Experiment Research on Misalignment of Rotor System. Master's Thesis, Northeastern University, Shenyang, China, 2010.
49. Huang, Y. An Analysis of Rotor Unbalance. *Power Equip.* **2009**, *23*, 164–169.
50. Pawlak, Z.; Skowron, A. Rudiments of rough sets. *Inf. Sci.* **2007**, *177*, 3–27. [CrossRef]
51. Liu, J.; Hu, Q.; Yu, D. Weighted Rough Set Learning: Towards a Subjective Approach. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 696–703.
52. Xu, F.; Zhang, Y.J. Integrated patch model: A generative model for image categorization based on feature selection. *Pattern Recognit. Lett.* **2007**, *28*, 1581–1591. [CrossRef]
53. Tax, D.; Juszczak, P. Kernel Whitening for one-Class Classification. *Int. J. Pattern Recognit. Artif. Intell.* **2003**, *17*, 333–347. [CrossRef]
54. Tax, D.; Duin, R. Support Vector Data Description. *Mach. Learn.* **2004**, *54*, 45–66. [CrossRef]
55. Bai, M.; Liu, J.; Ma, Y.; Zhao, X.; Long, Z.; Yu, D. Long short-term memory network-based normal pattern group for fault detection of three-shaft marine gas turbine. *Energies* **2021**, *14*, 13. [CrossRef]
56. Yu, H. SVMC: Single-class Classification with Support Vector Machines. In Proceedings of the 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 9–15 August 2003; pp. 567–572.
57. Ma, S.; Li, H.; Zhang, H.; Xie, X. *Reverberation Level Recognition by Formants Based on 10-fold Cross Validation of GMM. International Forum on Digital TV and Wireless Multimedia Communications*; Springer: Singapore, 2017; pp. 161–171.
58. Asuncion, A.; Newman, D. *Uci Machine Learning Repository*; University of California, School of Information and Computer Science: Irvine, CA, USA, 2007; p. 1994.
59. Grzymala-Busse, J.W. *LERS—A System for Learning from Examples Based on Rough Sets*; Intelligent Decision Support; Springer: Dordrecht, The Netherland, 1992; pp. 3–18.
60. Rocha, A.; Goldenstein, S.K. Multiclass from Binary: Expanding One-Versus-All, One-Versus-One and Ecoc-Based Approaches. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *25*, 289–302. [CrossRef] [PubMed]
61. Manevitz, L.M.; Yousef, M. One-class SVMs for Document Classification. *J. Mach. Learn. Res.* **2001**, *2*, 139–154.