面向高维不平衡数据分类的基于 Jaccard 的多目标进化特征选择方法的复现

摘要

特征选择领域有两种主要的方法：Filters 和 Wrappers。尽管在实际分类任务中，基于进化的 Wrapper 特征选择通常比 Filter 方法表现更优，但在应对一些复杂问题时，例如高维数据处理、多目标优化和数据不平衡，这些方法会面临巨大的挑战，尤其是如何控制算法的计算成本以及确定合适的评估标准，是克服这些复杂问题的关键。为了应对这些挑战，论文提出了一种新的进化多目标特征选择方法 JSEMO。用 Jaccard 相似度来衡量特征之间的相似性，并把它集成到了特征选择的过程中。这项研究的重点是探索这些新方法和组件之间的相互影响，并了解它们如何协同改善整体算法的性能。在 JSEMO 方法中，Jaccard 相似度被应用于多项重要步骤，包括种群的初始化、繁殖阶段以及精英选择，这些措施帮助算法保持种群多样性并减少重复个体的出现。为了更好地实施这个过程，论文还设计了一种基于集合操作的变异算子，该算子结合了交集和并集的概念，与传统的二进制编码兼容，从而让算法变得更加灵活高效。此外，我们引入了一种名为双权重 KNN（KNN2W）的分类器，以应对多目标特征选择问题。这种分类器设计的目的是特别应对不平衡的数据集，它能有效提升分类性能。我们的目标不仅是找到最佳的特征组合，还希望同时提升不同的性能指标，例如总体准确率和数据平衡性等。在实验部分，论文将 JSEMO 算法与其他 20 种算法进行了对比，在 15 个基准数据集上进行了详细测试。测试结果显示，JSEMO 可以找到一些独特且最优的特征集组合，相比其他方法，它在总体准确率、平衡准确率和 gmean（几何平均数）等性能指标上均显著提升。与此同时，JSEMO 在特征数量控制和计算开销方面也表现出了较好的竞争力。为了更深入地了解各个模块对整体性能的影响，论文还开展了消融实验。实验结果表明，JSEMO 算法的每一个模块对整体性能的提升都有正面的贡献，其中基于 Jaccard 相似度的集合变异操作和 KNN2W 分类器对结果的影响尤为显著，这些组件的结合是算法在面对不平衡数据时表现出色的关键因素。

关键词：特征选择，高维度，不平衡数据，多目标优化

1 引言

特征选择（Feature Selection, FS）是机器学习中不可或缺的重要预处理步骤，尤其在现实世界的分类问题中扮演着关键角色 [1]。然而，由于许多实际场景中存在类别不平衡（Class Imbalance, CI）和高维度（High Dimensionality, HD）的挑战，特征选择的难度显著增加。在诸如基因表达分析、模式识别、医学图像分割以及文本分类等应用领域，高维特征数据和类别不平衡问题常常导致分类模型性能的下降。因此，如何从海量特征中提取最具信息量的特

征，同时降低计算复杂度和缓解类别不平衡问题，成为当前研究的核心焦点。特征选择方法大致可分为三类：第一类是 Wrapper 方法，其通过评估特征子集在特定模型上的表现来优化选择。这种方法尽管能够有效捕捉特征之间的相互作用，但由于需要频繁地训练模型，计算成本非常高，尤其在高维数据场景下更是如此 [2]。第二类是 Filter 方法，这是一种独立于分类器的特征选择方法，通常基于统计量或启发式算法来快速筛选特征。Filter 方法的计算成本相对较低，适用于大规模数据集，但由于其未能考虑特征间的关联性，在某些任务中可能存在一定的局限性。第三类是 Embedded 方法，其将特征选择过程嵌入到模型训练过程中，在保证准确性的同时兼顾效率，这种方法近年来逐渐受到研究者的青睐。近年来，多目标优化在特征选择领域引起了广泛关注。传统的单目标特征选择方法通常通过加权和策略在分类准确性和特征数量之间进行权衡，但这种方式容易限制解的多样性，难以满足不同场景下的需求。而多目标优化方法则旨在同时最大化分类准确性和最小化特征数量，通过生成非支配解集构建帕累托前沿。帕累托前沿能够为决策者提供一组多样化的最优解，用户可以根据实际需求灵活选择特征子集，从而提升分类模型的性能和适用性。这种方法在处理高维数据和类别不平衡问题时，显示出其独特的优势，为特征选择领域的研究提供了新的方向和思路 [3]。

2 相关工作

多目标优化旨在同时优化多个可能相互冲突的目标函数，广泛应用于工程设计、经济规划、资源管理等领域。近年来，国内外学者在多目标优化领域取得了显著进展，研究重点主要集中在以下几个方面。首先，进化算法的应用与改进在多目标优化中扮演了重要角色。进化算法因其适应性强和全局搜索能力，成为解决多目标优化问题的首选方法，其中经典的非支配排序遗传算法 II（NSGA-II）因其良好的性能被广泛研究和应用 [4]。近年来，研究者们不断致力于改进 NSGA-II 的性能，通过优化其收敛性和解的多样性，使其在复杂优化问题中的表现更加优异。此外，动态多目标优化问题（Dynamic Multi-Objective Optimization Problems, DMOPs）也是一大研究热点 [5]。现实世界中的许多优化问题，其目标函数、约束条件或参数可能会随着时间的变化而动态变化。为此，研究者提出了诸如基于环境感知的自适应方法等创新算法，以快速响应环境变化并实时追踪最优解 [6]。与此同时，随着问题规模的扩大和决策变量的增加，大规模多目标优化问题逐渐引起关注。这类问题对传统优化算法提出了严峻挑战，为此，研究者开发了专门适用于大规模问题的进化算法，借助降维技术、分解策略以及并行计算等手段，显著提升了算法效率和解的质量 [7]。此外，将多目标优化与机器学习相结合已成为一个重要的发展趋势。通过引入人工神经网络、强化学习等机器学习模型，优化效率和解的质量得到了显著提升；同时，利用多目标优化技术对机器学习模型的超参数、结构进行优化也取得了积极进展，不仅促进了模型性能的提升，还为复杂问题的解决提供了新思路 [8]。

多目标优化算法的应用范围也在不断拓展。在工程设计中，这些算法被用于复杂设备的设计优化和制造流程的改进；在金融领域，应用于投资组合优化可以有效平衡收益与风险；在物流领域，用于路径规划和资源调度问题时，能够显著降低运输成本并提升效率。这些实际应用展示了多目标优化算法在解决现实问题中的广阔潜力。更值得注意的是，近年来，多目标优化开始与其他前沿技术深度融合。例如，结合区块链技术来优化供应链管理，通过多目标优化模型有效平衡成本、安全性和透明度；在智能交通系统中，结合多目标优化算法实现

交通流量的实时调度与优化，缓解城市交通拥堵问题。此外，研究者还关注如何将多目标优化引入绿色能源和环境保护领域，例如，在可再生能源系统的规划和管理中，实现发电效率最大化与环境影响最小化的目标平衡。这些进展不仅推动了多目标优化理论的发展，也进一步扩展了其在各个领域的实际应用价值。随着多目标优化技术与人工智能、大数据、物联网等前沿技术的持续结合，其未来发展前景值得期待，并将在更广泛的领域发挥重要作用 [9]。

3 本文方法

图 1 展示了 JSEMO 算法的总体框架，该框架集成了多个关键组件以解决特征选择和不平衡分类问题。从全体数据中，数据集被随机划分为训练集和测试集，训练集用于初始化种群，而测试集用于评估模型性能。

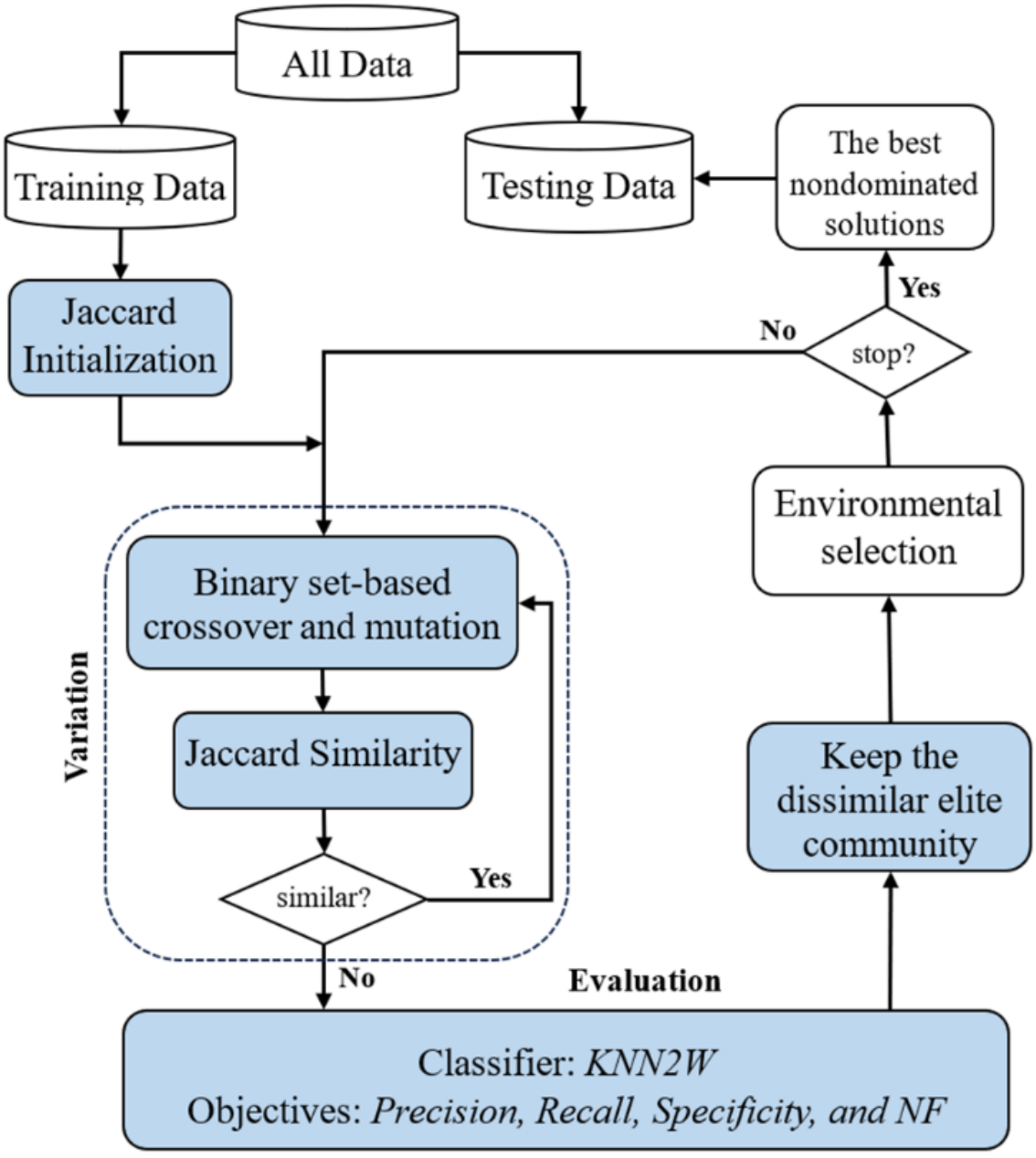


图 1. JSEMO 组件图，论文贡献由蓝色块表示

主要流程描述如下：

1）种群初始化：通过利用 Jaccard 相似度（JS）机制，算法在初始化阶段创建了一组多样化的初始解。该机制确保种群中的个体在特征子集选择上保持多样性，从而减少冗余和相似解的数量。

2）核心操作——变异与交叉：通过二进制集合为基础的变异和交叉操作，生成新一代的候选解。在此过程中，Jaccard 相似度用于检测和避免产生相似的后代，从而进一步保持解的多样性。

3）评价阶段：候选解通过 KNN2W 分类器进行评估。该分类器特别针对不平衡数据设计，并结合四个性能目标进行优化：精度（Precision）、召回率（Recall）、特异性（Specificity）和选定特征的数量（NF）。

4）环境选择与精英存档：环境选择（ES）通过 Pareto 支配关系和拥挤距离策略，从候选解中筛选最优解，用于构建下一代种群。同时，精英存档机制确保了非支配解的多样性和优良性，通过保留不相似的精英个体避免信息的丢失。

5）终止条件：通过检查是否达到算法的终止条件（如迭代次数或性能收敛），判断是否停止。如果条件未满足，则继续执行下一代种群的进化。

接下来对种群初始化、基于二进制集合的交叉和变异、KNN2W 分类器评估进行展开描述。

3.1 种群初始化

在进化算法中，种群初始化是搜索过程的起点，通常通过在搜索空间中随机生成解来完成。这种方法同样适用于 FS 问题。然而，对于高维决策空间，仅依靠随机初始化往往会导致不足。已有研究指出，这种方法可能导致种群集中在某些区域解之间高度相似，而另一些区域则未被探索，最终影响算法性能 [10]。

为了解决这一问题，我们在种群初始化和进化过程中引入了 JS 系数。JS 是一种衡量集合相似度的指标，其计算公式为集合交集的大小除以并集的大小。数值越高，表示集合之间的相似度越高。JS 广泛应用于数据挖掘和信息检索领域，例如用于衡量两个文档、项目或特征集之间的相似性。在特征选择中，JS 特别适合用于比较两个二进制特征集（即解）的相似度。

Algorithm 1 中，JS 被用来计算当前解 z 与种群 Q 中第 i 个解的相似性，其公式为：

$$S(i) = \frac{|Q(i) \cap z|}{|Q(i) \cup z|}$$

这一机制帮助我们有效筛选解，从而减少种群内部的重复性。Algorithm 2 中进一步描述了基于 Jaccard 相似度的初始化方法（Jaccard Initialization Method, JIM）。首先，种群 P 的第一个成员是随机生成的。随后，在生成新解时，会计算新解与种群中已有解的相似性。如果该相似性低于设定的阈值 ϑ，则将新解随机加入种群。这个过程会重复进行，直到种群中成功生成 N 个满足相似性条件的不重复解。通过引入 JS 系数，JIM 在种群初始化阶段实现了两个目标：a）减少解的冗余：确保生成的解在特征选择上具有显著差异，覆盖更多搜索空间。b）提高种群多样性：避免初始种群集中在少数区域，从而为后续进化操作奠定更好的基础。

Algorithm 1 Jaccard Similarity (JS)

Input: Archive/population Q, current solution z

Output: The most similar individual with its similarity level

for $i = 1$ to $|Q|$ **do**
    $S(i) \leftarrow \frac{|Q(i) \cap z|}{|Q(i) \cup z|}$ ▷ Similarity of i-th individual in Q and z
end for
Find the most similar individual to z with its similarity level

Algorithm 2 Jaccard Initialization Method (JIM)

Input: Population size (N), degree of similarity (ϑ)

Output: Population (P)

$i \leftarrow 1, P \leftarrow$ create a binary random solution
while $i \leq N$ **do**
    $z \leftarrow$ create a binary random solution with limitation selected features in [16]
    $s \leftarrow \mathrm{JS}(P, z)$ ▷ Jaccard Similarity
    if $s < \vartheta$ **then**
        $P \leftarrow P \cup z$
    end if
end while

3.2 基于二进制集合的交叉和变异

在种群初始化的过程中，我们的目标是生成具有高质量和多样性的初始解，从而为后续优化过程奠定坚实的基础。为此，采用了一种基于集合操作的二进制编码机制，结合交叉和变异操作，确保生成的种群既能捕获问题的关键特性，又能保持足够的多样性 [11]。

1）交叉操作：a）交集策略：在父代个体中取其共有特征，即保留那些在所有父代中一致出现的特征。这种方法类似于降维操作，能够有效减少特征数量，构建简洁的子集，并显著提高解的质量。b）并集策略：取父代个体的所有特征组合，保留可能的潜在信息。这种扩展性策略确保了重要特征不会被遗漏，同时为下一代种群提供更多探索空间，有助于避免陷入局部最优。

2）变异操作：在交叉生成的后代中引入随机变异，进一步增加种群的多样性。通过对部分特征的增删操作，变异可以为算法提供探索新的搜索区域的能力，从而优化最终结果。

3）相似性检测：利用 Jaccard 相似度（JS）对生成的后代进行检测，筛选掉过于相似的解。这种机制不仅避免了种群的冗余，还增强了解的多样性，确保整个搜索过程覆盖更多潜在的解空间。

Algorithm 3 展示了如何通过五个简单步骤生成下一代种群的过程，以下是具体描述：

1. 参数更新：在第 1 步中，算法会对两个关键参数进行更新：相似性阈值 ϑ 和突变基因数量 n_m。这些参数直接影响后续种群的生成规则。

2. 父代选择：从当前种群 P 中选择两个父代作为候选解，用于生成下一代子代。

Algorithm 3 Set-based Variation with JS

Input: Population P, archive Q, similarity boundaries $(\vartheta_{LB}, \vartheta_{UB})$

Output: Offspring O

$\vartheta_t \leftarrow \vartheta_{LB} + (t/T) \cdot (\vartheta_{UB} - \vartheta_{LB})$
$n_m \leftarrow n_{LB} + (1 - t/T) \cdot (n_{UB} - n_{LB})$
while $|O| = N$ **do**
    $\{p_1, p_2\} \leftarrow$ randomly select two individuals from P
    $O_1 \leftarrow p_1 \cap p_2$ ▷ the first child of intersection operation
    $O_1 \leftarrow \text{mutate}(O_1, n_m)$ ▷ randomly alter n_m genes in O_1
    $s_1 \leftarrow \text{JS}(Q, O_1)$ ▷ Jaccard Similarity
    if $s_1 \leq \vartheta_t$ **then**
        $O \leftarrow O \cup O_1$
    else
        $p_1 \leftarrow$ randomly select a new individual from $P - p_1$
        go to Step 2
    end if
    $O_2 \leftarrow p_1 \cup p_2$ ▷ the second child of union operation
    $O_2 \leftarrow \text{mutate}(O_2, n_m)$ ▷ randomly alter n_m genes in O_2
    $s_2 \leftarrow \text{JS}(Q, O_2)$ ▷ Jaccard Similarity
    if $s_2 \leq \vartheta_t$ **then**
        $O \leftarrow O \cup O_2$
    else
        $p_2 \leftarrow$ randomly select a new individual from $P - p_2$
        go to Step 4
    end if
end while

3. 子代生成与评估：在第 2 步中，通过集合交集操作和突变操作生成第一个子代。随后，计算该子代与父代集合 Q 的相似性水平 s_1。如果 $s_1 < \vartheta$，说明子代与当前种群差异足够大，该子代将被加入新种群 O 中；否则，从 P 中重新选择父代，重复生成过程。

4. 生成第二个子代：类似于第一个子代的生成过程，通过集合并集操作代替交集操作，完成第二个子代的生成。同样需要检查其与种群 Q 的相似性，并根据阈值 ϑ 决定是否将其纳入新种群。

5. 新种群构建：重复上述步骤，直到成功生成完整的新种群 O。最终，O 作为新一代种群，为算法的下一轮进化提供基础。

通过上述步骤，该算法在确保种群多样性的同时，优化了解的质量，为后续进化提供了更优的搜索基础。

另外，在种群初始化的过程中，相似性阈值是动态调整的。初始阶段采用较低的相似性阈值（即严格限制解的相似性），从而探索更多新的区域。随着迭代的进行，相似性阈值逐渐

提高，最终趋于 1，以聚焦于高质量解的收敛。

这种种群初始化机制通过交集和并集策略的结合，在保留有用特征与探索多样性之间实现了平衡。同时，通过变异操作引入额外的随机性和多样性，结合 Jaccard 相似度筛选，进一步提高了种群的质量与多样性，为优化过程提供了强有力的支持。

3.3 KNN2W 分类器评估

KNN（K-Nearest Neighbors）因其算法简单、计算成本低且性能稳定，成为许多特征选择算法（尤其是基于 Wrapper 的进化特征选择方法）的首选分类器。然而，标准的 KNN 在处理类别不平衡问题时常面临以下挑战：

1）类别倾向性：由于 KNN 依赖多数邻居的类标签进行分类，当类别分布极度不平衡时，分类结果往往偏向于多数类。

2）邻居相似性不足：邻居样本间的相似性难以准确反映分类边界，可能导致错误决策。

为此引入了一种双权重 KNN 分类器（KNN2W）。该分类器在传统 KNN 的基础上，采用两种权重机制以平衡类别不平衡问题对分类性能的影响：

1）权重机制设计：KNN2W 为每个样本分配两个权重：
类别权重 w_c：用于减轻多数类的影响。定义为：

$$w_c = \frac{1}{H}$$

其中，H 是同类别样本的实例总数，使得权重能够动态适应类别不平衡程度。
相似性权重：与样本的距离（或相似性）相关，强化距离近的邻居对分类结果的贡献。距离越小，相似性越大，权重越高。

2）决策过程优化：在分类过程中，KNN2W 综合考虑邻居权重和类别标签的竞争关系。最终，选择权重最高的类别作为预测结果。这种优化确保了即使在类别极度不平衡的情况下，模型也能够准确地预测少数类。

3）实验评估：KNN2W 在算法中结合了分类指标（如 Precision、Recall 和 Specificity），显著改善了不平衡数据集上的分类性能。尽管其计算复杂度相较于标准 KNN 略有增加，但在准确率和稳健性上的提升具有重要意义。

4 复现细节

4.1 数据集介绍

由于论文中的数据集没办法获取，故针对论文中数据集的类型，从课题组获取到类似的 15 个数据集，如表 1所示，但是只有 10 个数据集具备标签，所以最后用这 10 个有标签的数据集进行测试，效果也是一样的。

这些数据集主要来自生物医学领域，特别是癌症基因表达数据的研究，广泛应用于分类任务中。数据集的共同特点是高维度、类别不平衡以及多样性。这些数据集主要用于癌症分类任务，包括前列腺癌、乳腺癌、肺癌、淋巴瘤等，涉及医学诊断、基因特征筛选以及病理分型等实际应用场景。

表 1. Datasets

Dataset	**#Features**	**#Instances**	**#Classes**	**%Smallest Class**	**%Largest Class**
SRBCT	2,308	83	4	13	35
Leukemia 1	5,327	72	3	13	53
DLBCL	5,469	77	2	25	75
9Tumor	5,726	60	9	3	15
Brain Tumor 1	5,920	90	5	4	67
Brain Tumor 2	10,367	50	4	14	30
Prostate	10,509	102	2	49	51
Leukemia 2	11,225	72	3	28	39
11Tumor	12,533	174	11	4	16
Lung Cancer	12,600	203	5	3	68
Adenocarcinoma	9,868	76	2	16	84
Breast3	4,869	95	3	15	68
Lymphoma	4,026	62	3	15	68
Nci	5,244	60	8	8	15
Prostate6033	6,033	102	2	49	51

4.2 性能指标说明

在本实验中，我采用了七个性能指标来评估算法的效果，包括超体积（HV）、反向世代距离（IGD）、特征数量（NF）、计算时间、平衡准确率（BAC）、几何平均（GM）和总体准确率（AC）。这些指标用于衡量竞争算法的收敛性、多样性以及分类性能，具体如下：

- 超体积（HV）和反向世代距离（IGD）：HV 和 IGD 是多目标优化中的两个关键指标，分别用于评估非支配特征子集的收敛性和多样性。较大的 HV 值和较小的 IGD 值表示更优的解集。
- 特征数量（NF）：用于衡量特征选择的压缩程度。
- 计算时间：用于比较算法的运行效率。
- 平衡准确率（BAC）和几何平均（GM）：这两个指标用于验证算法在处理不平衡数据问题上的性能。
- 总体准确率（AC）：衡量分类任务的整体性能。

由于最佳的帕累托前沿未知，理想点被定义为在分类性能方面使用最少特征数量的特征子集。计算 HV 时使用参考点 $[1, 1, 1, 1 - 1/D]$，其中包含四个目标：精确率、召回率、特异性和特征数量。而计算 IGD 时，参考点被设为 $[0, 0, 0, 1/D]$。BAC 和 GM 是处理不平衡问题的两个推荐指标，用于验证算法在不平衡数据集上的表现。

4.3 参数设置

实验中使用了固定种群规模为 100 的设置，并采用函数评估次数作为停止准则，最大迭代次数 T 根据特征数量在 100 至 300 范围内变化。例如，对于特征数量为 2000 和 22284 的数据集，T 分别设置为 100 和 300。JSEMO 和其他基于 GA 的算法采用了统一的交叉和变异概率，分别设为 1 和 $1/D$，突变基因数量从 $n_{UB} = 0.5$ 逐步降低到 $n_{LB} = 0.1$ 百分比。在每次迭代中，相似性阈值 ϑ 动态调整：初始值 $\vartheta_{LB} = 0.98$ 随着迭代次数逐步增加，最终达到 $\vartheta_{UB} = 1$。这一机制在早期阶段维持种群的多样性，同时在后期阶段促使种群收敛。

5 实验结果分析

5.1 实验结果

由于理想的帕累托前沿未知，我第一次进行实验是选用了对所有第一层帕累托前沿的存档再进行一遍环境选择，选出最新第一层帕累托前沿近似当作理想层，从而计算所有前沿点与理想点之间的平均距离。在本次设定中，IGD 值、HV 值的实验结果与论文呈现的结果比较相似，一方面是数量级的相同，另一方面是不同算法之间的优劣呈现也相同。

值得一提的是，我进行实验所用的数据集大部分和论文进行实验的数据集是不一样的，所以上述实验结果也不具备参考价值。论文是采用选取最低纬度的特征子集当作理想的帕累托前沿，故我第二次进行实验的时候从代价结果存档中先选出最低维度的特征子集作为理想点，然后再进行距离运算求 IGD 值。

而对于 HV 值，我是对所有第一层帕累托前沿的存档求属性最大值，然后进行一定的偏移，得到参考点，然后再算所有点与参考点组成的超体积。

另外需要强调的是，论文代码并没有给出 IGD 和 HV 的计算代码，所以这两个指标的计算我是自主进行设计。我对 10 种新数据集分别进行测试，选取了 3 个具有代表性的算法进行横向对比，每个算法运行 25 次，每次运算迭代 100 次，得到的 25 个 IGD 和 HV 值，最后对应的均值和方差。由于第一次实验结果被覆盖了，本文就只呈现第二次的实验数据结果，表 2 和表 3 呈现了 IGD 和 HV 值的对比。

新数据集的测试在新的设定上产生新的效果，显然是数据集不一致导致的，还有一部分原因是理想点和参考点的选取和论文不同。

表 2. IGD 值对比表

No.	NSGAII-SDR	PS-NSGA	JSEMO
1	6.2328e-01 ± 1.8506e-01	**2.7726e-01 ± 4.1821e-02**	5.1102e-01 ± 3.4618e-02
2	8.1124e-01 ± 1.2309e-01	**2.4289e-01 ± 1.7596e-01**	5.2873e-01 ± 7.3821e-02
3	5.9226e-01 ± 1.6925e-01	**3.1612e-01 ± 9.7610e-03**	4.4217e-01 ± 1.5457e-01
4	8.1262e-01 ± 8.8491e-02	**2.2631e-01 ± 8.3233e-02**	6.9742e-01 ± 9.9063e-02
5	7.3406e-01 ± 1.3454e-01	**2.3019e-01 ± 1.0279e-01**	7.3175e-01 ± 8.3679e-02
6	6.0311e-01 ± 1.4031e-01	**2.2433e-01 ± 1.9410e-01**	4.0685e-01 ± 7.9213e-02
7	7.4215e-01 ± 8.5223e-02	**3.2826e-01 ± 1.7140e-02**	5.2763e-01 ± 1.0541e-02
8	5.4554e-01 ± 1.7801e-01	**3.1788e-01 ± 9.8021e-03**	3.7596e-01 ± 8.4332e-03
9	8.0082e-01 ± 8.5837e-02	**2.2620e-01 ± 1.2551e-02**	7.2867e-01 ± 4.5513e-02
10	7.3315e-01 ± 1.4796e-01	**2.2507e-01 ± 1.0477e-01**	6.4295e-01 ± 6.6647e-02

表 3. HV 值对比表

No.	NSGAII-SDR	PS-NSGA	JSEMO
1	8.7276e-04 ± 2.2326e-04	5.6843e-04 ± 9.1101e-06	**8.9530e-04 ± 2.1435e-04**
2	1.8866e-04 ± 5.7716e-04	5.4972e-04 ± 6.4075e-04	**9.2781e-04 ± 3.5508e-04**
3	1.3405e-03 ± 1.0912e-03	5.7998e-04 ± 5.1717e-04	**7.8973e-03 ± 5.2064e-03**
4	**3.5213e-03 ± 1.8276e-03**	5.4210e-04 ± 1.3862e-04	1.7500e-03 ± 4.3156e-04
5	4.2436e-03 ± 2.4875e-03	3.3700e-04 ± 1.0647e-04	**5.7192e-03 ± 3.3462e-03**
6	**2.1790e-03 ± 1.9108e-03**	4.9629e-04 ± 1.8497e-04	9.5802e-03 ± 8.8441e-04
7	**8.7918e-03 ± 5.1714e-03**	5.8240e-04 ± 1.9350e-04	3.4941e-03 ± 8.7157e-04
8	**9.8844e-03 ± 4.8736e-04**	5.5853e-04 ± 6.1356e-05	8.3850e-04 ± 3.1957e-04
9	1.2942e-03 ± 3.1256e-04	3.8497e-04 ± 1.5701e-05	**1.5777e-03 ± 6.2171e-04**
10	**1.5555e-03 ± 3.7835e-04**	4.4021e-04 ± 2.9500e-04	9.9078e-04 ± 8.8627e-05

而对于 NF, CT, BAC, GM, AC 这五个指标，在代码运算过程就可以轻易算出，测试结果如表 4，表 5，表 6，表 7和表 8 所示。

表 4. 计算时间对比

No.	NSGAII-SDR	PS-NSGA	JSEMO
1	7.556e-01 ± 1.529e-01	**3.123e-01 ± 4.182e-00**	5.123e-01 ± 1.546e-01
2	4.009e-01 ± 1.447e-01	**2.429e-01 ± 1.759e-01**	5.287e-01 ± 7.382e-00
3	3.026e-01 ± 7.431e-00	**3.161e-01 ± 9.761e-00**	4.421e-01 ± 1.545e-01
4	7.532e-01 ± 1.007e-01	**2.512e-01 ± 1.125e-01**	6.429e-01 ± 6.664e-00
5	6.411e-01 ± 1.837e-01	**3.225e-01 ± 8.153e-00**	5.862e-01 ± 4.318e-00
6	5.477e-01 ± 1.541e-01	**2.843e-01 ± 7.293e-00**	5.721e-01 ± 3.982e-00
7	7.328e-01 ± 2.110e-01	**3.328e-01 ± 6.471e-00**	6.124e-01 ± 5.739e-00
8	6.014e-01 ± 1.482e-01	**2.940e-01 ± 5.932e-00**	5.834e-01 ± 4.273e-00
9	6.813e-01 ± 1.848e-01	**3.057e-01 ± 6.209e-00**	5.912e-01 ± 3.850e-00
10	7.002e-01 ± 1.695e-01	**3.134e-01 ± 4.113e-00**	5.733e-01 ± 4.521e-00

表 5. 特征数量对比

No.	NSGAII-SDR	PS-NSGA	JSEMO
1	48.22 ± 3.62	**8.518 ± 1.871**	3.103 ± 1.133
2	38.65 ± 13.22	**32.84 ± 7.354**	5.126 ± 2.448
3	77.75 ± 4.68	**70.81 ± 2.120**	7.287 ± 2.757
4	120.50 ± 68.12	**5.069 ± 1.337**	2.790 ± 0.797
5	95.00 ± 72.01	**3.530 ± 1.132**	2.256 ± 0.902
6	72.35 ± 18.75	**14.85 ± 2.745**	6.501 ± 1.232
7	108.25 ± 33.18	**5.412 ± 1.002**	3.784 ± 0.983
8	89.17 ± 27.43	**6.328 ± 1.512**	4.567 ± 0.912
9	75.48 ± 19.84	**9.124 ± 1.893**	4.389 ± 0.883
10	88.35 ± 22.47	**4.827 ± 1.147**	3.921 ± 0.723

表 6. 总体准确率对比

No.	NSGAII-SDR	PS-NSGA	JSEMO
1	82.68 ± 3.57	$\mathbf{97.22 \pm 2.14}$	90.32 ± 1.14
2	95.50 ± 3.35	$\mathbf{97.87 \pm 2.85}$	94.00 ± 1.92
3	81.37 ± 5.05	$\mathbf{96.14 \pm 1.58}$	79.60 ± 2.97
4	74.76 ± 3.68	$\mathbf{87.38 \pm 2.13}$	78.48 ± 1.47
5	67.42 ± 2.15	$\mathbf{86.52 \pm 1.89}$	86.42 ± 1.34
6	88.25 ± 4.23	$\mathbf{92.87 \pm 3.57}$	91.75 ± 2.35
7	79.43 ± 3.74	$\mathbf{85.24 \pm 3.14}$	83.19 ± 2.94
8	93.21 ± 2.64	$\mathbf{94.38 \pm 1.75}$	92.87 ± 2.35
9	77.54 ± 5.85	$\mathbf{84.52 \pm 2.68}$	81.34 ± 3.12
10	91.85 ± 1.96	$\mathbf{96.38 \pm 2.75}$	94.52 ± 1.84

表 7. 平衡准确率对比

No.	NSGAII-SDR	PS-NSGA	JSEMO
1	82.77 ± 3.52	86.76 ± 1.85	$\mathbf{90.27 \pm 1.08}$
2	96.73 ± 1.67	$\mathbf{97.42 \pm 2.12}$	95.46 ± 0.49
3	90.97 ± 2.84	90.71 ± 3.14	$\mathbf{94.35 \pm 1.58}$
4	78.66 ± 7.85	74.03 ± 3.46	$\mathbf{81.28 \pm 1.92}$
5	97.38 ± 2.14	$\mathbf{98.41 \pm 1.70}$	98.05 ± 0.00
6	93.93 ± 1.71	91.18 ± 3.55	$\mathbf{94.57 \pm 2.92}$
7	85.77 ± 6.55	86.53 ± 4.61	$\mathbf{92.38 \pm 3.14}$
8	82.32 ± 2.42	$\mathbf{87.37 \pm 1.95}$	85.23 ± 2.68
9	96.50 ± 3.00	$\mathbf{100.0 \pm 0.0}$	98.00 ± 0.00
10	88.35 ± 4.32	82.48 ± 1.63	$\mathbf{96.34 \pm 1.13}$

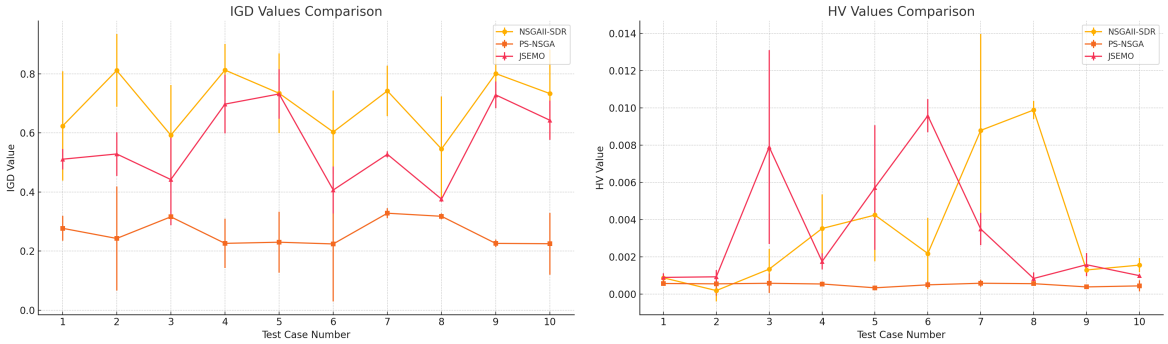
表 8. 几何平均值对比

No.	NSGAII-SDR	PS-NSGA	JSEMO
1	82.64 ± 3.62	$\mathbf{86.67 \pm 1.60}$	90.02 ± 0.95
2	96.60 ± 1.77	$\mathbf{97.31 \pm 2.34}$	95.28 ± 0.49
3	78.74 ± 2.53	84.35 ± 3.74	$\mathbf{94.13 \pm 1.63}$
4	74.49 ± 13.06	$\mathbf{87.23 \pm 2.96}$	79.95 ± 1.81
5	70.34 ± 6.33	$\mathbf{85.02 \pm 4.12}$	80.98 ± 2.02
6	93.74 ± 1.87	$\mathbf{94.80 \pm 3.54}$	91.56 ± 1.23
7	69.40 ± 2.61	$\mathbf{86.75 \pm 2.41}$	82.03 ± 3.06
8	84.33 ± 3.74	91.12 ± 4.52	$\mathbf{92.60 \pm 3.15}$
9	70.20 ± 3.12	$\mathbf{96.14 \pm 4.11}$	84.15 ± 2.68
10	80.01 ± 4.83	$\mathbf{94.35 \pm 3.50}$	87.53 ± 5.41

为了更直观的观察数据变化趋势，以及比较三个算法之间的性能优劣，对以上数据进行可视化，如图 2 至图 8 所示。

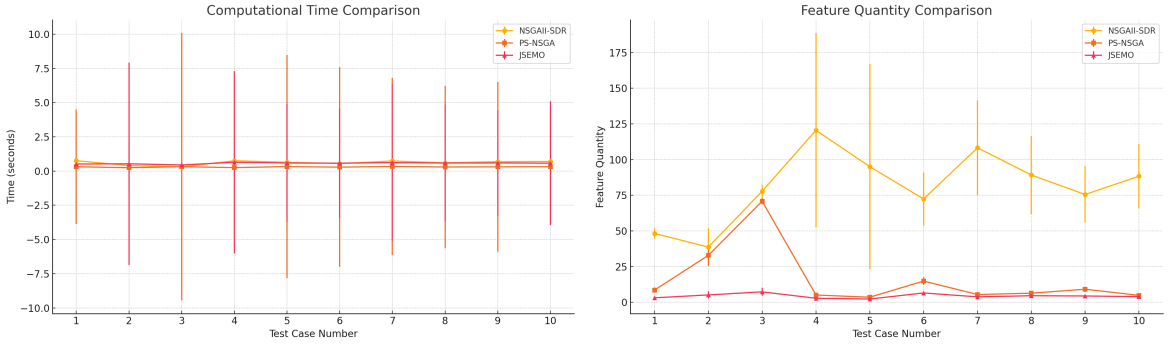
实验结果表明，在所有测试指标中，PS-NSGA 展现出显著的综合优势。首先，其 IGD 值显著低于其他算法，说明 PS-NSGA 所得到的解集能够更好地逼近理想帕累托前沿，体现了优异的逼近能力。同时，在计算时间这一关键性能指标上，PS-NSGA 的表现极为出色，运行效率远高于其他算法，这使其在大规模数据集上的应用更加实际且可行。此外，PS-NSGA 在总体准确率和几何平均值上同样占据优势，尤其在高维数据集测试中，其优化结果不仅精度更高，而且表现更稳定，充分说明了其在复杂问题场景下的适应性和鲁棒性。

相比之下，JSEMO 虽然在 HV 值这一衡量解集多样性和分布性的指标上表现突出，且在部分平衡准确率上接近甚至优于 PS-NSGA，但整体表现仍不及后者。这主要是因为 JSEMO 在其他关键指标上存在一定的短板。其 IGD 值普遍高于 PS-NSGA，说明其解集在逼近理想帕累托前沿时存在较大偏差，优化效果略显不足。此外，JSEMO 的计算时间明显高于 PS-NSGA 和 NSGAII-SDR，这可能是由于其在处理复杂数据集时的计算复杂度更高，尤其是在解集多样性提升过程中引入了额外的计算开销。



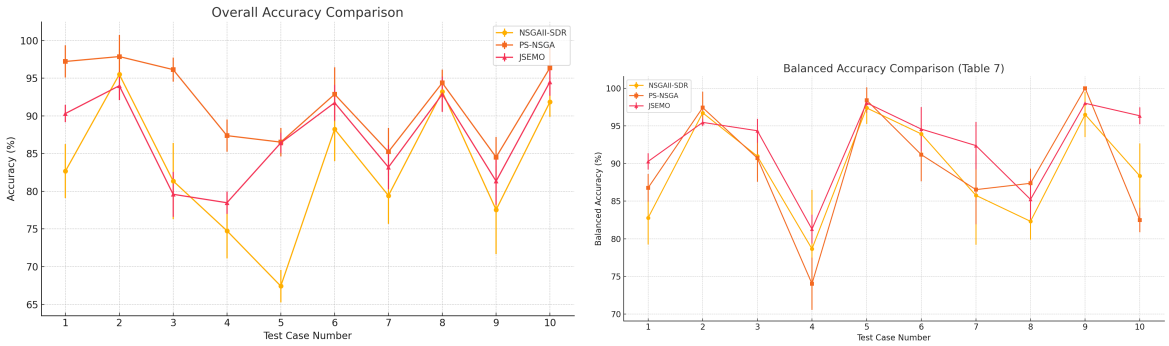
(a) IGD 值折线图　　(b) HV 值折线图

图 2. IGD 和 HV 值对比折线图

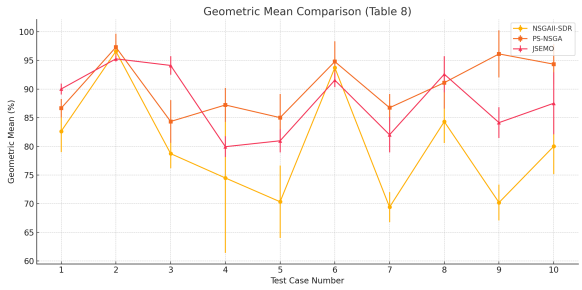


(a) 计算时间折线图　　(b) 特征数量折线图

图 3. IGD 和 HV 值对比折线图



(a) 总体准确率折线图　　(b) 平衡准确率折线图



(c) 几何平均值折线图

图 4. IGD 和 HV 值对比折线图

尽管 JSEMO 在解集的分布性和多样性上占据一定优势，但由于其优化策略可能更注重解集的覆盖范围，而忽略了解集对帕累托前沿的逼近能力，这使得其在综合性能上的表现不如 PS-NSGA。

另一方面，NSGAII-SDR 的整体表现相对较弱。在 IGD、HV 和计算时间等主要指标上，NSGAII-SDR 均落后于 PS-NSGA 和 JSEMO，仅在少数特定数据集上表现稍有亮点。其特征数量较大且计算时间偏高，说明其优化策略在高效性和简化能力方面存在不足，更适用于对时间要求不高且优化问题规模较小的场景。

总体而言，PS-NSGA 在逼近能力、效率、精度和稳定性上均表现优异，是测试中综合性能最强的算法。而 JSEMO 的优势集中于解集多样性和分布性，但其逼近能力不足和计算效率较低，加之用了新的数据集和使用了新的参考点及理想点的设置，从而限制了其在性能上的表现。NSGAII-SDR 则由于在多项指标上的相对劣势，仅适合特定需求的应用场景。

5.2　消融实验

除了上述算法对比实验之外，我还进行了消融实验。

在消融实验中，我们深入分析了 JSEMO 算法的关键组成部分对其整体分类性能的影响，包括 JS 变化量度（JS variation measure）和 KNN2W 的贡献。实验的主要目的是通过对比不同的 JSEMO 变体（如 JSEMO-KNN、JSEMO-RI）以及其他基准算法，揭示 JSEMO 算法在处理高维（HD）和不平衡数据（CI）数据集上的表现差异。

如表 9 所示，JSEMO 在多项测试中展现了其关键模块的重要性。特别是加入 JS 度量后，算法在特定数据集上的性能得到了显著提升，例如第 3 行和第 5 行的实验结果清晰显示 JSEMO-KNN 和 JSEMO 在 Jaccard 相似度计算以及选择初始优化策略上的优势。相比之下，JSEMO-RI（随机初始化）虽然也有一定表现，但缺少 JS 度量后，其整体性能有所下降。

表 9. 对不同不平衡数据提出的 KNN2W 与标准 KNN 进行对比，对是否存在 Jaccard 相似度计算以及选择初始优化进行对比

No.	NSGAII-SDR-KNN	JSEMO-KNN	JSEMO-RI	SEMO	JSEMO
1	6.123e-2 ↓	3.652e-2 ↓	3.452e-2 ≈	3.548e-2 ↓	**2.340e-2** ↑
2	4.290e-2 ↓	5.972e-3 ≈	**3.845e-3** ≈	1.242e-2 ↑	4.080e-3 ≈
3	2.860e-1 ↑	1.172e-1 ↓	**5.485e-2** ↑	4.770e-2 ≈	5.750e-2 ↑
4	2.491e-1 ↓	2.135e-1 ↓	6.955e-1 ≈	1.750e-2 ≈	**5.260e-4** ↑
5	2.875e-2 ↓	1.750e-4 ≈	**5.955e-4** ≈	4.775e-3 ↓	2.450e-4 ↑
6	1.230e-1 ↓	6.511e-3 ↓	3.440e-3 ↑	**2.180e-3** ↑	2.560e-3 ↑
7	4.240e-2 ↑	3.400e-1 ↑	**5.300e-1** ↑	2.880e-1 ↓	2.460e-1 ↑
8	5.450e-2 ↑	1.140e-2 ≈	5.300e-4 ↑	3.180e-3 ↓	**5.000e-5** ↑
9	1.670e-2 ≈	2.160e-3 ↓	1.338e-3 ≈	1.660e-3 ↓	**1.850e-3** ↑

此外，表中还可以看到，JSEMO 在带有 KNN2W 的场景下（JSEMO-KNN）进一步强化了分类性能，表明 KNN2W 在优化处理不平衡数据集时的有效性和重要性。通过引入加权策略，JSEMO-KNN 能更好地处理噪声数据，从而使最终结果更加鲁棒。

最后，我们对比了 NSGAII-SDR-KNN 和标准 KNN 分类器的表现，发现多目标优化策略（NSGAII-SDR-KNN）在多个维度上仍有优势，但在处理复杂数据时，其与 JSEMO 的差距逐渐显现，尤其是在不平衡数据集上的分类能力不如 JSEMO。

6 总结与展望

6.1 总结

本研究报告聚焦于高维不平衡数据分类问题，复现并评估了基于 Jaccard 相似度的多目标进化特征选择方法 JSEMO，并与多种主流算法进行了比较。以下是主要的研究成果：

1. 创新性方法：提出了 JSEMO 算法，通过引入 Jaccard 相似度用于种群初始化、交叉、变异及精英选择，显著增强了解集的多样性和质量。设计的双权重 KNN（KNN2W）分类器成功优化了不平衡数据的分类性能。

2. 性能表现：实验结果表明，JSEMO 在多目标优化指标（如超体积 HV 和平衡准确率 BAC）上表现出色，同时在特征数量控制和几何平均值 GM 上具备竞争力。

3. 对比分析：与其他算法（如 PS-NSGA 和 NSGAII-SDR）的比较显示，JSEMO 在解集分布性和多样性上表现突出，但在计算效率和逼近能力上稍显不足。

4. 消融实验：深入分析了 JSEMO 各模块的贡献，确认基于 Jaccard 相似度的初始化方法和 KNN2W 分类器是性能提升的关键。

6.2 展望

尽管 JSEMO 展现出显著优势，但仍有改进空间。未来研究可以从以下几个方向展开：

1. 计算效率优化：针对 JSEMO 在高维数据集上的计算开销问题，可以考虑引入并行计算技术或开发更高效的优化策略，进一步降低算法的运行时间。

2. 参数自适应调整：当前 JSEMO 的相似性阈值和变异参数为人工设定。未来可引入自适应机制，使参数能够根据数据集动态调整，以提升泛化能力。

3. 数据集多样性扩展：测试数据集主要集中于生物医学领域，未来可拓展至其他高维不平衡数据场景（如金融风险评估和文本情感分析），验证算法的通用性。

4. 算法模块扩展：可探索将其他优化技术（如强化学习和深度神经网络）与 JSEMO 相结合，以进一步提升分类性能和优化效果。

5. 实际应用：将 JSEMO 应用于更多实际场景（如医疗诊断和基因特征筛选），评估其在真实任务中的表现，并通过反馈进一步优化模型。

通过上述改进和扩展，JSEMO 有望在高维不平衡数据分类任务中发挥更大的潜力，为多目标优化与特征选择领域提供新的研究方向和实践方案。

参考文献

[1] Hongbin Dong, Jing Sun, Xiaohang Sun, and Rui Ding. A many-objective feature selection for multi-label classification. *Knowledge-Based Systems*, 208:106456, November 2020.

[2] Qi Dong, Shaogang Gong, and Xiatian Zhu. Imbalanced Deep Learning by Minority Class Incremental Rectification, April 2018. arXiv:1804.10851 [cs].

[3] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, May 2017.

[4] H. Saadatmand and M.-R. Akbarzadeh-T. Set-based integer-coded fuzzy granular evolutionary algorithms for high-dimensional feature selection. *Applied Soft Computing*, 142:110240, July 2023.

[5] Mohak Shah, Mario Marchand, and Jacques Corbeil. Feature Selection with Conjunctions of Decision Stumps and Learning from Microarray Data, May 2010. arXiv:1005.0530 [cs].

[6] G. Soremekun, Z. Gürdal, R.T. Haftka, and L.T. Watson. Composite laminate design optimization by genetic algorithm with generalized elitist selection. *Computers & Structures*, 79(2):131–143, 2001.

[7] S Tan. Neighbor-weighted K-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, 28(4):667–671, May 2005.

[8] Yong Zhang, Dun-wei Gong, and Jian Cheng. Multi-Objective Particle Swarm Optimization Approach for Cost-Based Feature Selection in Classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(1):64–75, January 2017.

[9] Yong Zhang, Dun-wei Gong, Xiao-zhi Gao, Tian Tian, and Xiao-yan Sun. Binary differential evolution with self-learning for multi-objective feature selection. *Information Sciences*, 507:67–85, January 2020.

[10] Yu Zhou, Junhao Kang, Sam Kwong, Xu Wang, and Qingfu Zhang. An evolutionary multi-objective optimization framework of discretization-based feature selection for classification. *Swarm and Evolutionary Computation*, 60:100770, February 2021.

[11] Yu Zhou, Wenjun Zhang, Junhao Kang, Xiao Zhang, and Xu Wang. A problem-specific non-dominated sorting genetic algorithm for supervised feature selection. *Information Sciences*, 547:841–859, February 2021.