超高维数据分析

杜睿

首都经济贸易大学，北京，100071；

摘要：本研究基于 MNIST 手写数字数据集，探讨了超高维数据下支持向量机（SVM）与随机森林（Random Forest）分类模型的性能差异，并分析了主成分分析（PCA）降维技术对模型效率与准确率的影响。通过 PCA 保留 95%方差降维后，SVM 模型在测试集上的准确率仍达 0.9644，与未降维模型性能接近，但显著提升了计算效率。研究进一步通过混淆矩阵、ROC 曲线及 AUC 值评估了模型的分类能力，发现降维后模型在保持高准确率的同时，降低了数据存储与计算复杂度。结论表明，在计算资源有限或需增强可解释性时，降维是有效策略，但需权衡潜在特征丢失风险。本文为高维数据处理与分类模型优化提供了实践参考。

关键词：MNIST 数据集降维；支持向量机；随机森林；主成分分析

DOI：10.69979/3041-0673.25.08.014

1 研究的背景及意义

MNIST 是一个手写体数字的图片数据集，该数据集由美国国家标准与技术研究所发起整理，一共统计了来自 250 个不同的人手写数字图片，其中 50%是高中生，50%来自人口普查局的工作人员。该数据集的收集目的是希望通过算法，实现对手写数字的识别。

其中训练集一共包含了 60000 张图像和标签，而测试集一共包含了 10000 张图像和标签。测试集中前 5000 个来自最初 NIST 项目的训练集，后 5000 个来自最初 NIST 项目的测试集。该数据集自 1998 年起，被广泛地应用于机器学习和深度学习领域，用来测试算法的效果，例如线性分类器、K-近邻算法、支持向量机、神经网络、卷积神经网络等等。

在算法与模型创新上，手写数字识别可以为研究者提供一个统一的评估标准，可用于比较不同算法在同一任务上的性能，有助于算法的发展和改进；MNIST 的简单性和易用性也有利于研究者尝试新的机器学习方法和模型架构，便于验证新方法的可行性和有效性，为复杂视觉任务提供基础。在 MNIST 上探索不同的特征提取和降维技术，可以为更复杂的图像识别任务提供经验和洞察。

2 模型介绍与实证分析

2.1 不降维建立分类模型

在 python 中导入手写数据集，并将其划分为训练集和测试集。

2.1.1 支持向量机

用支持向量机 SVM 进行分类，首先需要对特征数据进行归一化处理，归一化后加快了梯度下降求最优解的速度，也有可能提高精度。而概率模型（树形模型）不需要归一化，因为它们不关心变量的值，只关心变量的分布和变量之间的条件概率，如决策树。

SVM 在高维空间中依然具有高效性，它可以处理具有大量特征的数据集，并且不易受到维度灾难的影响。SVM 的泛化能力强，通过最大化分类间隔来构建最优的决策边界，从而在训练数据之外的新数据上表现出很好的泛化能力。对于未知数据的预测效果往往较好。SVM 在处理线性可分和线性不可分的数据时都能够取得良好的效果。通过使用核函数，将数据映射到高维特征空间，从而处理非线性问题。

但是 SVM 对参数敏感，受核函数、惩罚系数等参数影响较大，选择不当的参数值可能导致模型性能下降，不同的数据集可能需要不同的核函数，选择合适的核函数和调整模型参数对于实现最佳性能至关重要；对于大规模数据集，SVM 的训练时间较长；SVM 对于缺失数据比较敏感，如果数据中存在缺失值，需要进行额外的处理，否则可能影响模型的性能。

先对手写数据集进行标准差标准化（standardScale），使手写数据集经过处理的数据符合标准正态分布，即均值为 0，标准差为 1。使用 SVM 的运行结果显示准确率为 0.963。

2.1.2 随机森林

随机森林通过集成多个决策树的结果来进行分类，能够有效降低误差。在手写数字分类中，每个决策树可能关注不同的像素特征组合，通过综合多个树的预测，可以提高分类的准确性；此外，随机森林的鲁棒性强，对异常值和噪声有较好的容忍度；随机森林通过随机选择样本和特征来构建决策树，增加了模型的多样性，减

少了过拟合的风险；它具有更好的泛化能力，能够适应不同风格和书写方式的手写数字样本。

但是随机森林计算复杂度较高，需要构建多个决策树，这使得训练时间和内存需求大幅增加。对于手写数字分类任务而言，当数据集规模较大时，训练随机森林需要较长时间和大量内存资源，导致训练过程缓慢；而且随机森林的可解释性相对较弱，虽然随机森林中的每个决策树是可解释的，但作为一个整体，其模型的解释性不如单个决策树直观。在手写数字分类中，很难像解释决策树那样清晰地阐述随机森林是如何根据像素特征进行分类决策的，比如在研究手写数字识别的原理和特征重要性时，随机森林的解释难度相对较大。

对手写数据集使用随机森林分类器训练模型，指定随机森林中决策树的数量为 100，因为更多的决策树通常会让模型的预测效果更加稳定准确。在测试集数据 X_test 上进行预测，将预测结果存放在 y_pred_rf 中，最终计算出准确率约为 0.97。

2.2 降维后建立分类模型

2.2.1 主成分分析 PCA

通过线性变换将原始数据转换到新的坐标系统中，使得任何投影的第一大方差在第一个坐标（第一主成分）上，第二大方差在第二个坐标（第二主成分）上，依此类推。PCA 的基本思想是对于原始的 p 个变量，想要找出 k 个新变量（两两不相关）来代替原始变量，并在尽可能保留原有信息的基础上，使得 k 尽可能小，保留数据集中对方差贡献最大的特征来降低数据的维度。

PCA 计算方法简单，容易实现，可以减少指标筛选的工作量；消除变量间的多重共线性；在一定程度上能减少噪声数据。但是特征必须是连续型变量（可以在一定范围内连续取值的变量）；也无法解释降维后的数据是什么；有时候贡献率小的成分有可能更重要，但是会被降维的时候舍弃掉。

对手写数据集使用 scikit_learn 库进行 PCA 分析，保留 95%的方差解释率来确定降维后的维度数量。接着使用训练集调用 fit_transform 方法进行拟合和转换，得到降维后的训练数据。对于测试数据，则直接使用已经拟合好的 PCA 对象调用 transform 方法进行转换，得到 X_test_pca，确保测试数据的转换是基于训练数据拟合的模型来进行的。然后创建了一个支持向量机分类器对象，使用降维后的训练数据以及对应的训练标签进行拟合训练。之后分别在降维后的训练数据和测试数据上进行预测，最终训练集的准确率为 0.9855，测试集的准确率约为 0.9644

3 评估模型

3.1 数据抽取

前面均使用原始的手写数据集对数据进行降维分类处理，但是由于数据量巨大导致模型处理时间过长。所以采用抽取数据集的方法，从中随机抽取了 6000 个数据，减少样本量。

3.2 不降维情况下的最优超参数

先创建一个 SVM 分类器的实例，由于数据量仍然不少，所以设置 cv=3，即采用 5 折交叉验证的方式来评估不同超参数组合下模型的性能。利用网格搜索在给定的超参数范围内寻找最优的 SVM 超参数组合，为'C': 0.1, 'gamma': 0.01, 'kernel': 'linear'。

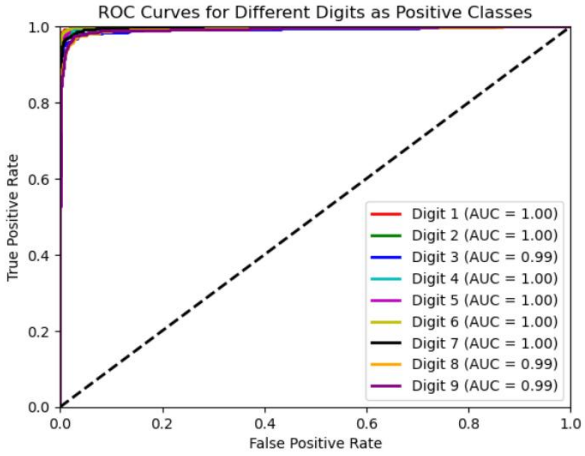
3.2.1 混淆矩阵（ROC 曲线、AUC 值）

二值化标签，循环遍历数字 1-9 作为正类进行处理，AUC 的值（仅以 digit1 为例）与该模型的准确率如下：

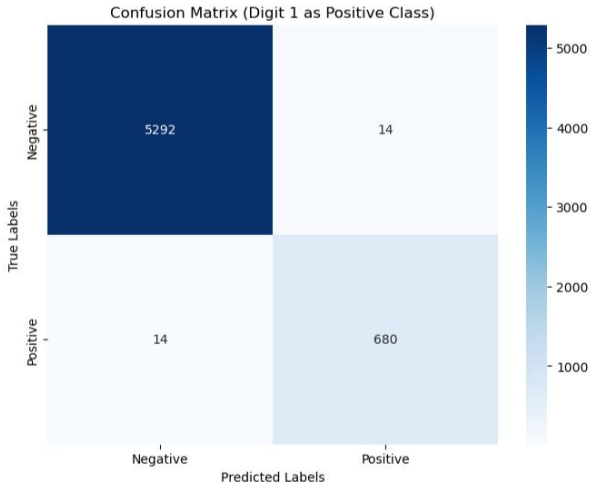
AUC value when digit 1 is positive class: 0.9976605112154407

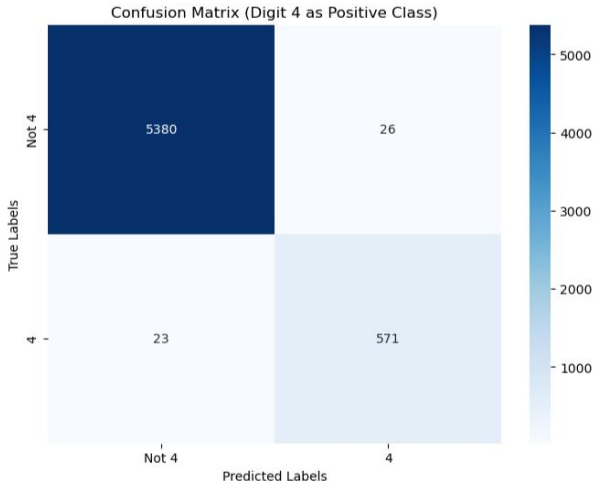
Accuracy when digit 1 is positive class: 0.9945

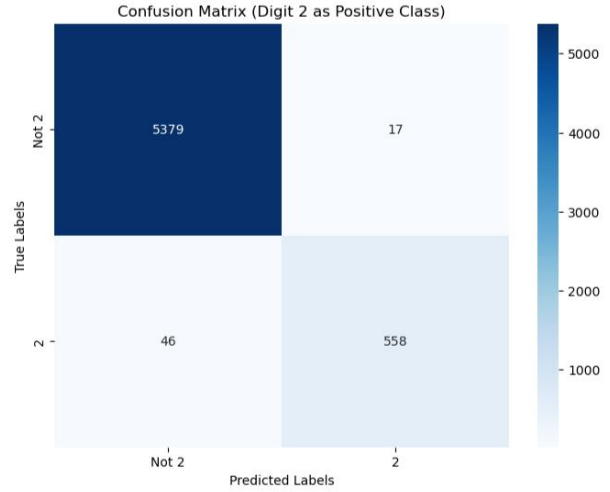
二值化标签，循环遍历数字 1-9 作为正类进行处理，使用抽取后的训练子集训练模型，计算 ROC 曲线的假正率、真正率和阈值。绘制出的 ROC 曲线非常靠近左上角，如下所示：

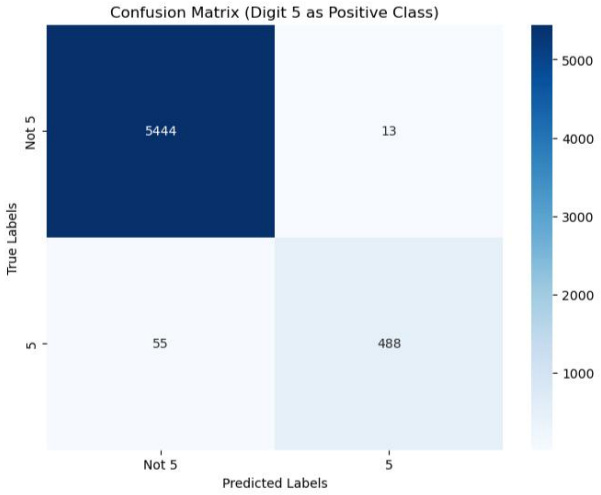


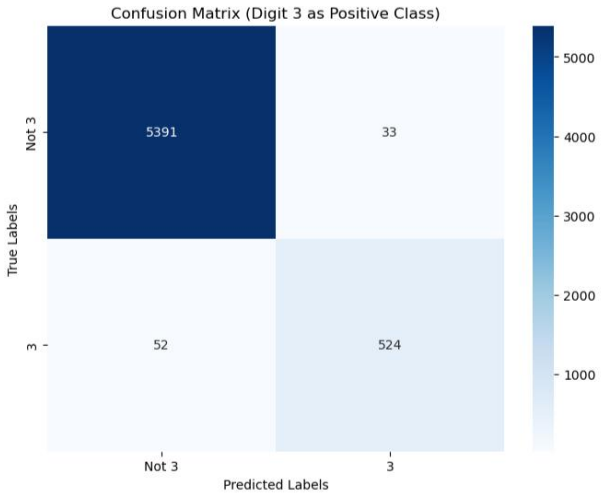
混淆矩阵主要用于比较分类结果和实际测得值，可以把分类结果的精度显示在一个矩阵里面。每一列代表了预测类别，每一列的总数表示预测为该类别的数据的数目。每一行代表了数据的真实归属类别，每一行的数据总数表示该类别的数据实例的数目，每一列中的数值表示真实数据被预测为该类的数目。计算混淆矩阵。如下：

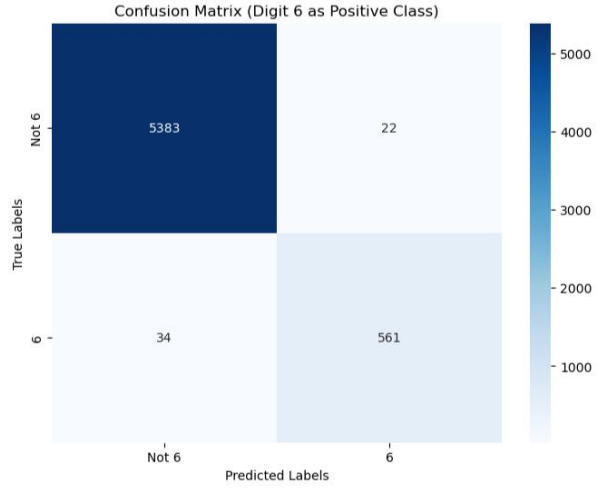

Confusion Matrix (Digit 1 as Positive Class)
True Labels
Negative
Positive
5292
14
14
680
Negative
Positive
Predicted Labels
5000
4000
3000
2000
1000

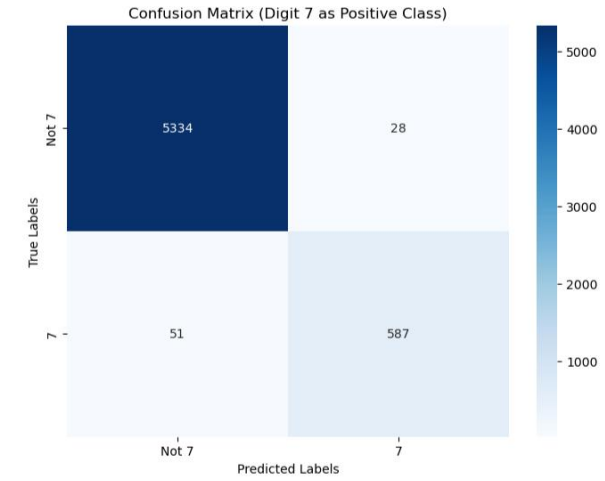

Confusion Matrix (Digit 4 as Positive Class)
True Labels
Not 4
4
5380
26
23
571
Not 4
4
Predicted Labels
5000
4000
3000
2000
1000

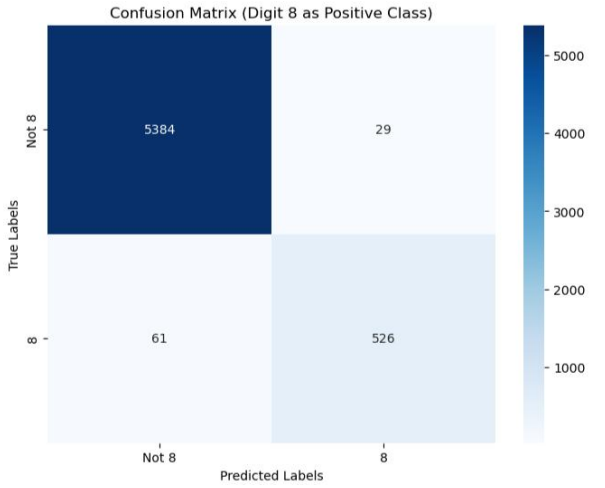

Confusion Matrix (Digit 2 as Positive Class)
True Labels
Not 2
2
5379
17
46
558
Not 2
2
Predicted Labels
5000
4000
3000
2000
1000

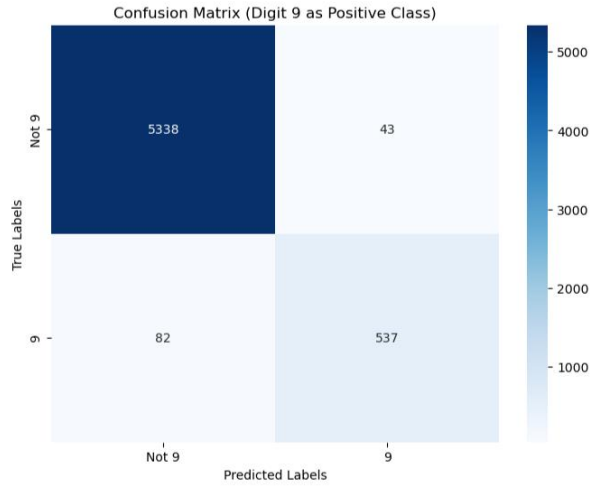

Confusion Matrix (Digit 5 as Positive Class)
True Labels
Not 5
5
5444
13
55
488
Not 5
5
Predicted Labels
5000
4000
3000
2000
1000


Confusion Matrix (Digit 3 as Positive Class)
True Labels
Not 3
3
5391
33
52
524
Not 3
3
Predicted Labels
5000
4000
3000
2000
1000


Confusion Matrix (Digit 6 as Positive Class)
True Labels
Not 6
6
5383
22
34
561
Not 6
6
Predicted Labels
5000
4000
3000
2000
1000







3.3 降维后的最优超参数

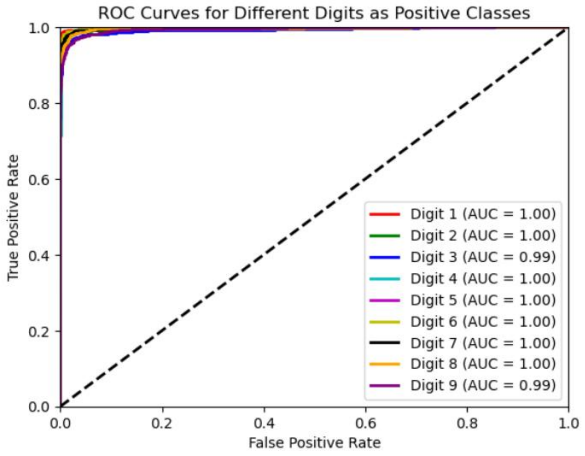
使用 PCA 进行降维，这里设置保留的主成分数量为 95。也创建一个 SVM 分类器的实例，为提高数据处理速度，设置 cv=3，即采用 3 折交叉验证的方式。把数据集划分为 3 份，每次去其中 2 份作为训练集，剩下的为验证集，循环三次，最终确定表现最优的超参数组合为'C': 0.1, 'gamma': 0.01。

3.3.1 混淆矩阵（ROC 曲线、AUC 值）

使用 seaborn，将混淆矩阵可视化。在一个多分类问题上绘制 ROC 曲线，由于 ROC 曲线只支持二分类问题，需要将多分类标签转化成二值化标签，为每个类别分别绘制 ROC 曲线图。召回率（TPR）越高，分类器产生的假正类率（FPR）就越多。虚线表示纯随机分类器的 ROC 曲线，一个优秀的分类器应该离这条线越远越好，由图知，ROC 曲线均靠近左上角。二值化标签，循环遍历数字 1-9 作为正类进行处理。AUC 是 ROC 曲线下面积的一个数值表示，它提供了一个定量的指标，用来衡量分类模型的整体表现。最终计算出 AUC 的值（仅以 digit1 为例）与准确率如下：

AUC value when digit 1 is positive class: 0.9982508608661826

Accuracy when digit 1 is positive class: 0.9956666666666667



混淆矩阵如下（未可视化）：

```
Best parameters when digit 1 is positive class: {'C': 0.1, 'gamma': 0.01}
Confusion matrix when digit 1 is positive class:
[[5329    0]
 [ 671    0]]
Best parameters when digit 2 is positive class: {'C': 0.1, 'gamma': 0.01}
Confusion matrix when digit 2 is positive class:
[[5413    0]
 [ 587    0]]
Best parameters when digit 3 is positive class: {'C': 0.1, 'gamma': 0.01}
Confusion matrix when digit 3 is positive class:
[[5392    0]
 [ 608    0]]
Best parameters when digit 4 is positive class: {'C': 0.1, 'gamma': 0.01}
Confusion matrix when digit 4 is positive class:
[[5434    0]
 [ 566    0]]
Best parameters when digit 5 is positive class: {'C': 0.1, 'gamma': 0.01}
Confusion matrix when digit 5 is positive class:
[[5454    0]
 [ 546    0]]
Best parameters when digit 6 is positive class: {'C': 0.1, 'gamma': 0.01}
Confusion matrix when digit 6 is positive class:
[[5384    0]
 [ 616    0]]
Best parameters when digit 7 is positive class: {'C': 0.1, 'gamma': 0.01}
Confusion matrix when digit 7 is positive class:
[[5373    0]
 [ 627    0]]
Best parameters when digit 8 is positive class: {'C': 0.1, 'gamma': 0.01}
Confusion matrix when digit 8 is positive class:
[[5392    0]
 [ 608    0]]
Best parameters when digit 9 is positive class: {'C': 0.1, 'gamma': 0.01}
Confusion matrix when digit 9 is positive class:
[[5409    0]
 [ 591    0]]
```

4 模型结果分析

在对手写数据集建立分类模型的过程中，发现降维前模型准确率为和降维后的模型准确率相差不大。如果更关注模型训练和预测的效率，比如在处理大规模数据集或者计算资源有限的情况下，降维是个不错的选择。因为降维可以减少数据存储量、加快运算速度，即便准确率没有显著提升，但效率方面会受益。要是对模型的可解释性有较高要求，降维后的低维数据在一定程度上可能会更便于理解和解释手写数据的特征结构，也可以考虑降维。然而，要是担心降维过程可能会丢失潜在的、后续可能有用的细微特征，并且当前的准确率和效率已经能够满足需求，那么也可以选择保持数据的原始维度，暂不进行降维。

5 总结与感悟

由于数据的属性很多或者维数很高，其维数甚至高达上千上万维，而缓解维数灾难的重要途经是降维，把高维数据映射到低维空间进行处理。在最大限度地保持数据之间差异的前提下，消除不相关或冗余的维。这样就避免了空空间的现象，在子空间中样本密度高，计算距离、近邻点、相似度变得容易。

在对数据进行处理完毕后，也会有相应的方法对模型进行评估，如建立混淆矩阵，观察 ROC 曲线是否接近于左上角来检验该分类器的性能，AUC 的值是否大于 0.5，大于的话才能确保正样本属于正类的置信度大于负样本属于正类的置信度的概率。再如，若想了解算法为什么具有这样的性能，可以使用偏差-方差分解这一工具。为了取得更好的泛化性能，则需使偏差和方差尽可能小，才能充分拟合数据，产生小影响的数据扰动。

总而言之，机器学习是计算机科学的重要分支领域。它致力于研究如何通过计算的手段，利用经验来改善系统自身的性能，在计算机系统中，“经验”通常以“数据”形式存在，因此机器学习所研究的主要内容，是关于在计算机上从数据中产生“模型”的算法，即“学习算法”。有了学习算法，把经验数据提供给它，它就能基于这些数据产生模型，在面对新的情况时，模型会给我们提供相应的判断。

参考文献

[1]杨舟,崔彩霞.基于异质 Stacking 集成学习的大学生学业风险预测及早预警[J].太原师范学院学报(自然科学版),2025,24(01):22-29+39.
[2]张晏濒.基于深度学习的英文手写词汇数据集构建方法[J].电脑知识与技术,2025,21(03):34-38.DOI:10.14004/j.cnki.ckt.2025.0119.
[3]郭尚志,廖晓峰,李刚,等.基于PCA的大数据降维应用[J].计算机仿真,2024,41(05):483-486.
[4]翁雪慧,汪晓锋,应鹏,等.基于 SVM 与 K-means 的多层级雷达信号开集识别[J/OL].北京航空航天大学学报,1-12[2025-04-14].https://doi.org/10.13700/j.bh.1001-5965.2024.0369