**山东科技大学**

**实验报告**

**课程名称： 大数据分析方法及应用实验**

**实验项目： 支持向量机在钓鱼网站识别中的应用研究**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

# 目录

[目录 2](#_Toc18241)

[引言 3](#_Toc3541)

[文献综述 4](#_Toc20182)

[1.算法介绍 5](#_Toc26506)

[1.1最大间隔与最优超平面 5](#_Toc31866)

[1.2支持向量 5](#_Toc2550)

[1.3原始优化问题 5](#_Toc75)

[1.4对偶问题与核方法 5](#_Toc30178)

[2.实例分析 6](#_Toc11402)

[2.1数据来源 6](#_Toc7869)

[2.2数据描述 6](#_Toc12370)

[2.3数据预处理 7](#_Toc757)

[2.4数据分析 7](#_Toc29972)

[2.5支持向量机 9](#_Toc8204)

[2.6结果分析 12](#_Toc7384)

[3.结论 12](#_Toc26703)

[参 考 文 献 13](#_Toc20364)

[附录 14](#_Toc20319)

引言**​**​

近年来，互联网的普及和信息技术的发展极大地便利了人们的生活和工作，但同时也带来了网络安全风险的加剧。钓鱼网站作为网络诈骗的主要形式之一，通过伪造合法网站界面，诱使用户泄露个人隐私和财务信息，已成为网络安全领域的一大威胁。据相关统计，钓鱼攻击的数量呈现逐年上升趋势，且攻击方式不断演变，给传统的安全防护机制带来严峻挑战。

传统的钓鱼网站检测依赖于黑名单机制和规则匹配，虽然在一定程度上能够防范已知威胁，但面对新型钓鱼网站和变异攻击效果有限，且维护成本较高。随着机器学习技术的成熟，越来越多的研究将其应用于钓鱼网站检测，通过自动提取和学习网站特征，实现对未知威胁的识别，提升检测的智能化水平。

支持向量机作为一种强有力的二分类算法，能够在高维空间寻找最优分割超平面，实现数据的有效分类。其良好的理论基础和稳定的泛化能力，使其在网络安全、文本分类等领域表现优异。利用SVM进行钓鱼网站检测，不仅可以处理大量复杂特征，还能有效避免过拟合，具有较强的实际应用价值。

本实验基于Kaggle平台的Phishing Websites数据集，采用支持向量机对网站的多维特征进行分析，构建自动化的钓鱼网站识别模型。通过实验评估模型的分类性能，结合准确率、召回率、F1值和ROC曲线等指标，全面衡量模型效果。实验结果将为网络安全防护系统提供有效的技术支持，推动钓鱼网站检测技术的进一步发展。

本研究不仅丰富了基于机器学习的网络安全检测方法，也为相关领域提供了可借鉴的实验方案和实践经验，对于提升互联网用户的安全防范意识和保护网络环境具有积极意义。

文献综述

随着互联网的迅猛发展，网络安全问题日益严峻，钓鱼网站作为一种典型的网络攻击手段，利用欺骗性的伪造网页诱导用户泄露敏感信息，已成为网络诈骗和信息泄露的重要来源。钓鱼网站的隐蔽性强、变化多样，使得传统依赖规则和黑名单的检测方法难以有效防御，亟需更加智能化、自动化的检测手段来提升识别效率和准确性[1][2]。

学者们从多角度对钓鱼网站的特征进行了深入研究，包括URL特征、域名注册信息、网页内容结构、页面行为及网络流量等多个层面。张伟等[3]基于URL和域名的静态特征构建检测模型，发现短期注册域名、异常URL长度等是典型钓鱼网站的标志。国际研究中，Kumar等[4]通过整合静态与动态特征，利用机器学习方法提升了对新型钓鱼攻击的检测能力，显示了多特征融合的优势。

随着机器学习技术的不断成熟，支持向量机（SVM）因其在高维空间的强大分类能力和良好的泛化性能，成为钓鱼网站检测领域的主流算法之一。李强等[5]结合URL词法和网页内容特征，采用SVM实现对钓鱼网站的准确分类，有效提升了检测的召回率和精确度。王芳等[6]的研究表明，SVM在样本量有限、特征维度较高的场景下，优于神经网络和决策树等方法，表现出稳定性和鲁棒性。

特征工程在钓鱼网站检测中起到关键作用。赵丽[7]提出多源特征融合与选择方法，通过结合URL信息、页面结构及用户访问行为，显著提升了模型性能。虽然深度学习模型如卷积神经网络和循环神经网络在图像和序列数据处理上表现优异，但面对以结构化特征为主的钓鱼网站检测任务，传统机器学习方法仍然因其计算效率和可解释性备受青睐。

本实验旨在通过利用Kaggle公开的Phishing Websites数据集，深入分析数据特征，应用支持向量机算法进行分类建模，并通过多指标对模型性能进行综合评价，以期为钓鱼网站检测领域提供有效的技术方案和实践经验。

# 

# 1.算法介绍

支持向量机（Support Vector Machine, SVM）是一种基于统计学习理论的监督分类算法，核心思想是通过最大化类别间隔来提高分类的泛化能力。

1.1最大间隔与最优超平面

SVM假设训练样本集为 {(, )}，其中 ∈ 为特征向量， ∈ {+1, -1} 为类别标签。SVM在特征空间中寻找一个最优超平面

将正负样本分开，并使正负样本到超平面的几何间隔（margin）最大。样本到超平面的间隔为

最大间隔对应的优化目标为最小化 。

1.2支持向量

在最优超平面下，距离超平面最近且满足约束条件的样本点被称为支持向量。这些样本点位于“间隔边界”上，定义为

支持向量直接决定了分类超平面的位置，其他非支持向量样本对超平面无影响。

1.3原始优化问题

SVM的原始优化问题为：

约束：

1.4对偶问题与核方法

通过拉格朗日乘子法，可将原始问题转化为对偶问题，便于引入核函数处理非线性分类。SVM的对偶问题为：

约束：

其中 为拉格朗日乘子， 为核函数（如线性核、多项式核、RBF核），用于将数据从原始空间映射到高维空间，使其线性可分。

最终决策函数为：

# 2.实例分析

## 2.1数据来源

本研究所采用的数据集来源于Kaggle平台上的“Phishing Websites Data”，由数据贡献者 UCI Machine Learning Repository 提供。该数据集系统收集了多种网站的结构特征、URL特征及行为指标，涵盖包括IP地址特征、URL长度、域名信息、SSL证书状态、页面重定向、页面元素、域名年龄等超过30项静态和动态特征。数据集标签明确区分钓鱼网站与正常网站，适用于网络钓鱼攻击检测与防御研究。

数据集包含约11,000条网站样本，覆盖了不同地域和时间段内的多种钓鱼攻击案例及正常网站，有效支持机器学习模型的训练和验证。该数据集广泛应用于网络安全防护、恶意网站检测、风险评估和机器学习算法性能比较等多个领域，具有较高的代表性和实用价值。

## 2.2数据描述

表1 数据描述表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **变量名称** | **含义** | **类型** | **单位/取值范围** | **示例值** |
| **having\_IP\_Address** | 是否含IP地址 | 数值型 | 二元（1/-1） | 1 |
| **URL\_Length** | URL长度特征 | 数值型 | 二元（1/-1） | -1 |
| **Shortining\_Service** | 是否使用短链接服务 | 数值型 | 二元（1/-1） | 1 |
| **having\_At\_Symbol** | URL中是否含有@符号 | 数值型 | 二元（1/-1） | -1 |
| **double\_slash\_redirecting** | 是否含双斜杠重定向 | 数值型 | 二元（1/-1） | 1 |
| **Prefix\_Suffix** | 是否含有前缀或后缀 | 数值型 | 二元（1/-1） | -1 |
| **having\_Sub\_Domain** | 是否含子域名 | 数值型 | 二元（1/-1） | 0 |
| **SSLfinal\_State** | SSL证书最终状态 | 数值型 | 二元（1/-1） | 1 |
| **Domain\_registeration\_length** | 域名注册时间长度 | 数值型 | 二元（1/-1） | -1 |
| **Favicon** | 是否含有Favicon图标 | 数值型 | 二元（1/-1） | 1 |
| **port** | 端口号是否为标准端口 | 数值型 | 二元（1/-1） | 1 |
| **HTTPS\_token** | URL中是否含有HTTPS令牌 | 数值型 | 二元（1/-1） | -1 |
| **Request\_URL** | 请求URL特征 | 数值型 | 二元（1/-1） | 1 |
| **URL\_of\_Anchor** | 页面锚点URL特征 | 数值型 | 二元（1/-1） | -1 |
| **Links\_in\_tags** | 页面中标签链接数量 | 数值型 | 二元（1/-1） | 1 |
| **SFH** | 页面重定向数量 | 数值型 | 二元（1/-1） | -1 |
| **Submitting\_to\_email** | 是否向邮箱提交数据 | 数值型 | 二元（1/-1） | -1 |
| **Abnormal\_URL** | URL异常特征 | 数值型 | 二元（1/-1） | 1 |
| **Redirect** | 是否含重定向 | 数值型 | 二元（1/-1） | 0 |
| **on\_mouseover** | 鼠标悬停事件 | 数值型 | 二元（1/-1） | 1 |
| **RightClick** | 右键禁用事件 | 数值型 | 二元（1/-1） | 1 |
| **popUpWidnow** | 弹出窗口事件 | 数值型 | 二元（1/-1） | 1 |
| **Iframe** | 是否含Iframe标签 | 数值型 | 二元（1/-1） | -1 |
| **age\_of\_domain** | 域名年龄 | 数值型 | 二元（1/-1） | 1 |
| **DNSRecord** | 是否存在DNS记录 | 数值型 | 二元（1/-1） | -1 |
| **web\_traffic** | 网站访问量 | 数值型 | 二元（1/-1） | 1 |
| **Page\_Rank** | 页面排名 | 数值型 | 二元（1/-1） | -1 |
| **Google\_Index** | 是否被谷歌索引 | 数值型 | 二元（1/-1） | 1 |
| **Links\_pointing\_to\_page** | 指向页面的链接数量 | 数值型 | 二元（1/-1） | 1 |
| **Statistical\_report** | 统计报告特征 | 数值型 | 二元（1/-1） | -1 |
| **Result** | 分类结果（钓鱼/非钓鱼） | 标签 | 1（钓鱼）/-1（非钓鱼） | 1 |

**2.3数据预处理**

**标签编码**：将“Result”列中的标签进行数值化处理，将原始取值-1转换为0，使其符合scikit-learn二分类模型的输入规范，便于后续训练和评估；

**缺失值检查与清洗**：对所有特征列进行了完整性检测，确认数据集中无缺失值或异常空值，无需额外的填充处理；

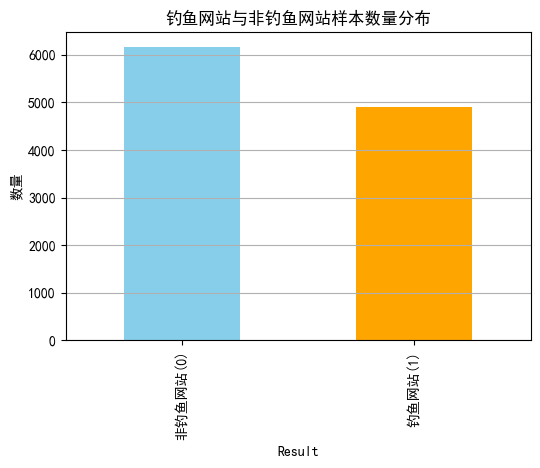
**特征选择与筛查**：结合领域知识及前期探索性分析，保留了与钓鱼网站识别高度相关的主要特征，包括URL结构特征（如having\_IP\_Address、URL\_Length）、安全性特征（如SSLfinal\_State、HTTPS\_token）、流量特征（web\_traffic）、索引特征（Google\_Index）等，删除部分对分类贡献较低的冗余字段，提高模型训练效率；

**划分训练集与测试集**：采用stratify分层抽样方法，按照80%训练集和20%测试集的比例对样本进行分割，确保两部分中钓鱼网站与非钓鱼网站的比例一致，避免类别不平衡导致的模型偏差和评估失真。

这一系列预处理操作确保了输入数据结构规范、特征信息充分、类别分布均衡，为支持向量机模型的有效训练和稳定预测奠定了良好的基础。

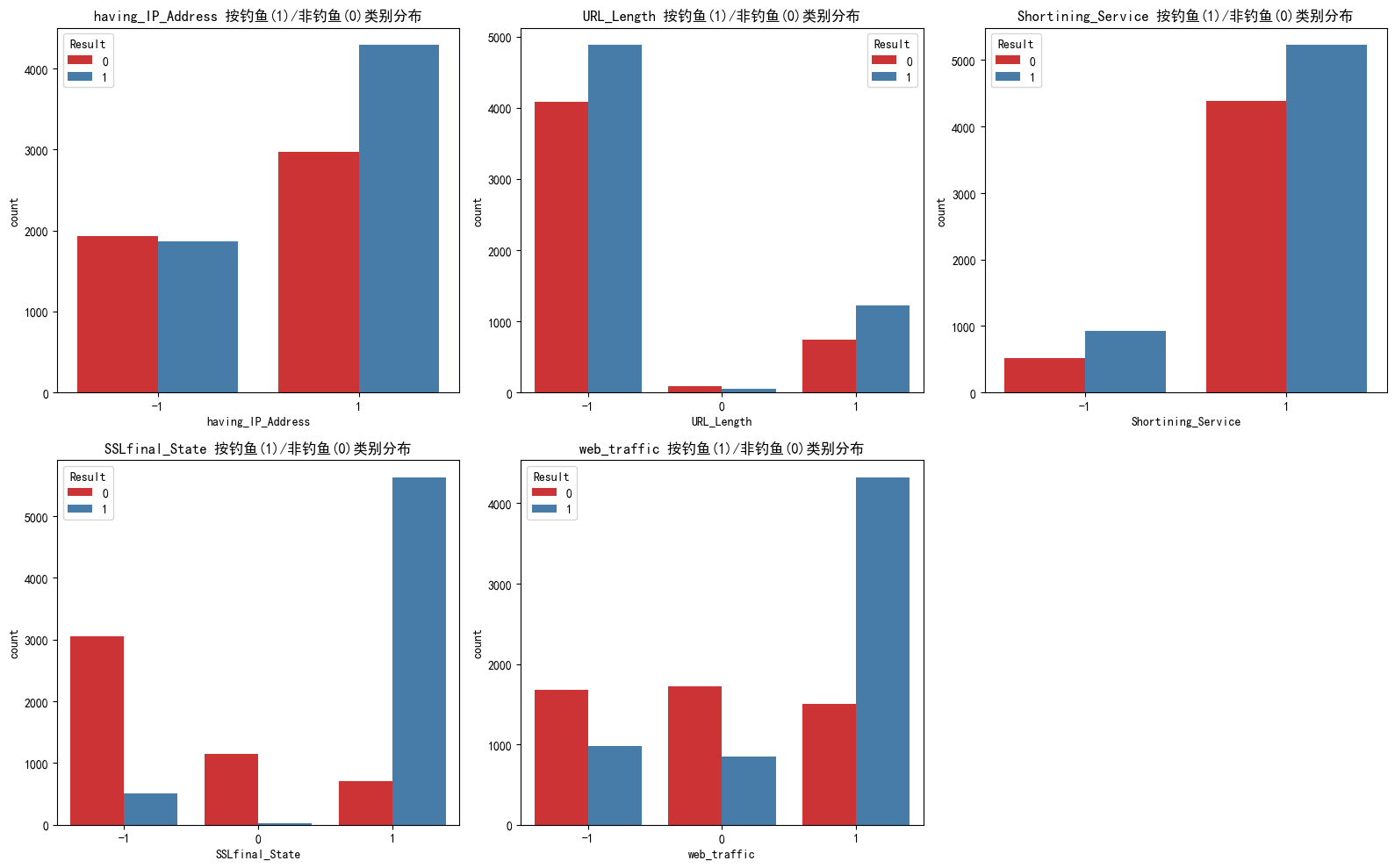
**2.4数据分析**

图1 钓鱼与非钓鱼网站样本数量分布图



这张图展示了钓鱼网站（标签为1）与非钓鱼网站（标签为0）的样本数量分布情况。可以看出，非钓鱼网站的样本数量略多于钓鱼网站，分别约为6200个和4900个。整体样本数量分布较为均衡，有利于模型训练过程中避免类别不平衡带来的偏差，从而提升分类模型的泛化能力和准确率。

图2 关键特征钓鱼与非钓鱼网站类别分布图



这组图展示了五个关键特征在钓鱼网站（标签为1，蓝色）与非钓鱼网站（标签为0，红色）样本中的分布差异：

**having\_IP\_Address**：钓鱼网站中更多样本带有IP地址特征（值为1），而非钓鱼网站两种状态较为均衡，说明IP地址特征对识别钓鱼网站有一定指示作用。

**URL\_Length**：钓鱼网站更多样本对应URL长度特征为-1，表明较短或异常长度的URL可能与钓鱼行为相关。

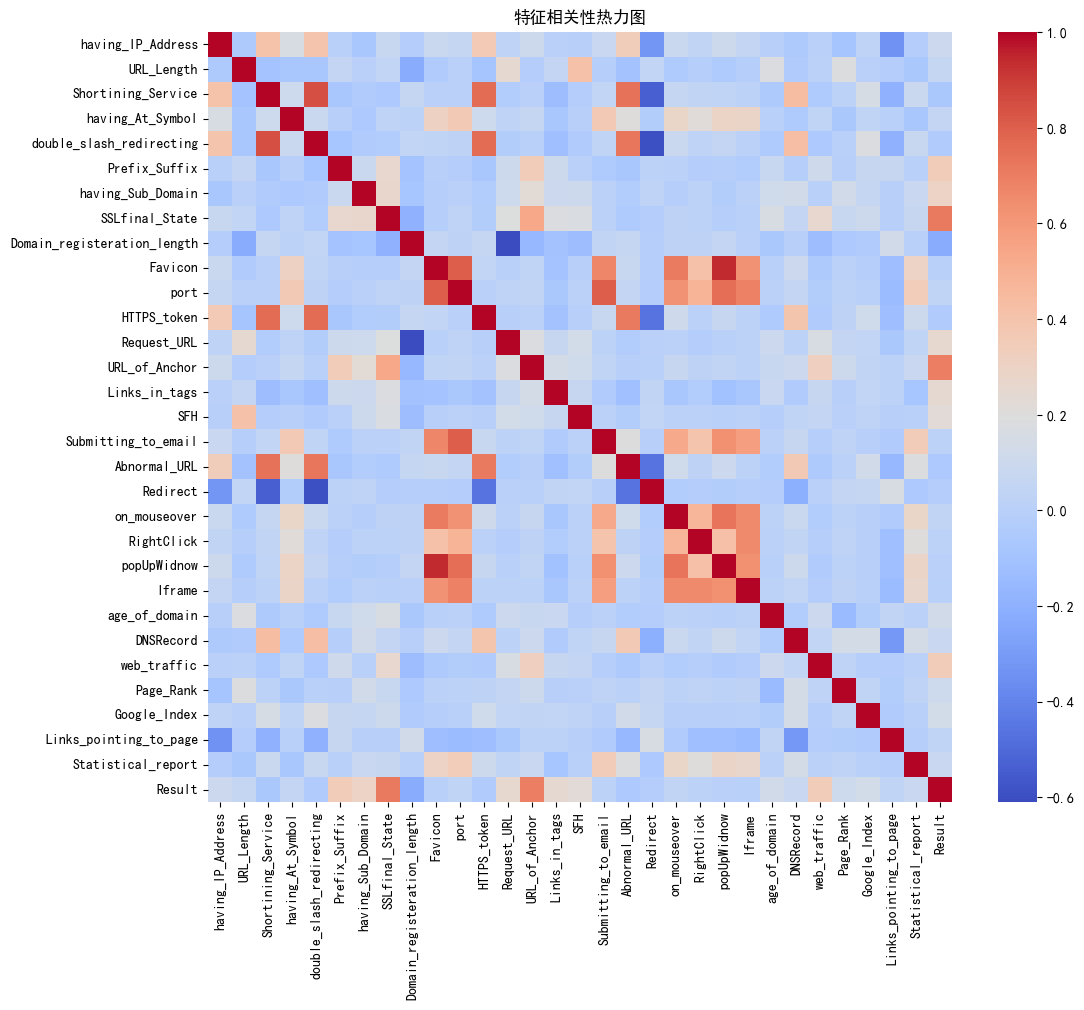
**Shortining\_Service**：使用短链接服务（值为1）的钓鱼网站明显多于非钓鱼网站，短链接常被用于隐藏真实URL，具有较强欺骗性。

**SSLfinal\_State**：绝大多数钓鱼网站SSL状态为1（可能表示SSL证书异常或缺失），而非钓鱼网站多为-1，显示SSL证书状态是重要的安全判别特征。

**web\_traffic**：钓鱼网站在高流量（值为1）区间的样本数远多于非钓鱼网站，表明部分钓鱼网站访问量较高，可能与攻击规模或传播能力相关。

总体来看，这些特征在钓鱼和非钓鱼网站间呈现显著分布差异，对模型区分两类网站具有较强的判别能力，说明它们是构建高效钓鱼网站检测模型的重要变量。

图3 特征变量相关性热力图



该特征相关性热力图直观展示了所有变量之间的两两皮尔森相关系数关系，可以看出大多数特征之间的相关性较低（颜色接近浅蓝或白色），说明它们在信息上相对独立，有助于模型捕捉多维特征信息。同时，部分变量与标签“Result”存在显著相关性，如SSLfinal\_State、web\_traffic和having\_IP\_Address等与分类结果呈中等以上的正或负相关，这表明这些特征在钓鱼网站检测中具有较高的判别价值，为后续特征选择和模型优化提供了重要参考依据。

2.5支持向量机

本研究基于支持向量机对钓鱼网站数据进行分类识别，实验流程如下：

首先，利用Python的pandas库读取数据集，将Result列作为标签（y），其余特征列作为输入特征（X）。由于原始标签中钓鱼网站标记为-1，为了符合Scikit-learn的二分类标准，先将标签值-1替换为0，即0表示“非钓鱼”，1表示“钓鱼”。随后按8:2的比例将数据划分为训练集和测试集，并通过stratify参数保持类别分布一致性，保证训练与测试样本比例的均衡。

在模型构建阶段，选用SVM分类器，核函数类型设为径向基函数（Radial Basis Function, RBF），启用概率输出（probability=True），并设置随机种子（random\_state=42）以确保结果的可复现性。模型在训练集上进行拟合后，对测试集样本进行预测，输出类别预测结果及概率值。

准确率 ：0.9488919041157847

分类报告：

表2 支持向量机结果分类报告

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **类别** | **精确率 (Precision)** | **召回率 (Recall)** | **F1分数 (F1-Score)** | **支持数 (Support)** |
| **0** | 0.96 | 0.92 | 0.94 | 980 |
| **1** | 0.94 | 0.97 | 0.95 | 1231 |
| **​**​准确率 (Accuracy)​**​** | - | - | 0.95 | 2211 (总计) |
| **​**​宏平均 (Macro Avg)​**​** | 0.95 | 0.95 | 0.95 | 2211 |
| **​**​加权平均 (Weighted Avg)​**​** | 0.95 | 0.95 | 0.95 | 2211 |

图4 支持向量机结果混淆矩阵

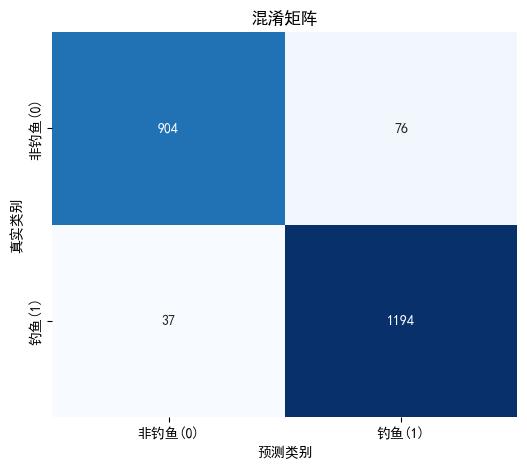
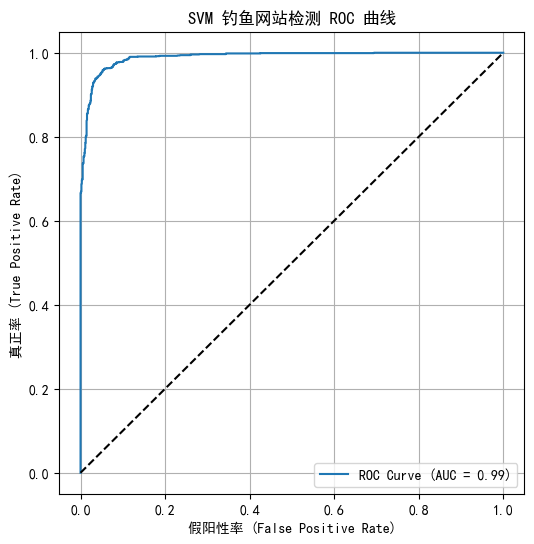


图5 支持向量机模型ROC曲线图



2.6结果分析

本实验采用支持向量机（SVM）对钓鱼网站数据进行分类，模型在测试集上的表现如下：

模型准确率达到94.89%，整体分类效果优异。

混淆矩阵显示，模型对非钓鱼网站正确识别904个样本，误判76个样本；对钓鱼网站正确识别1194个样本，仅误判37个样本，说明模型在两类样本上均具备较高的区分能力。

分类报告中，非钓鱼类别的精确率为0.96，召回率为0.92，F1分数为0.94；钓鱼类别的精确率为0.94，召回率为0.97，F1分数为0.95，宏平均和加权平均均达到0.95，表明模型性能稳定且均衡。

ROC曲线下的AUC值为0.99，接近理想状态，进一步验证了模型对钓鱼网站的高识别能力和较低的误报率。

综上，基于SVM的钓鱼网站检测模型能够有效识别钓鱼行为，具有良好的泛化能力和实际应用潜力。该模型可为网络安全防护系统提供重要技术支持，提升钓鱼攻击的自动检测与防范效果。

# 3.结论

本研究基于Kaggle公开的Phishing Websites数据集，利用支持向量机（SVM）算法对钓鱼网站和非钓鱼网站进行了分类建模与实验分析。在系统的数据预处理和特征工程基础上，通过对多个关键变量的描述性统计与可视化分析，深入理解了各特征在不同类别中的分布规律。实验结果表明，SVM模型在该任务上表现出了较强的分类能力和良好的泛化性能。

在模型评估中，测试集的准确率达到94.89%，分类报告显示钓鱼和非钓鱼两类样本的精确率、召回率和F1分数均在0.94以上，宏平均与加权平均均为0.95，验证了模型的鲁棒性与均衡性。ROC曲线的AUC值高达0.99，进一步说明模型具有出色的区分能力和极低的误报率。混淆矩阵分析显示，SVM能够有效识别大部分钓鱼网站样本，仅有少量误判，符合实际应用对高安全性和高准确率的需求。

本实验还发现，诸如URL长度、IP地址使用、SSL状态、短链接服务等特征在识别钓鱼网站中具有显著作用，这表明合理的特征选择和多特征融合是提升检测准确性的关键因素。此外，实验过程验证了SVM对高维、非线性数据的良好适应性和在小样本情况下的优秀性能。

然而，本研究也存在一定局限性：一方面，数据样本主要来源于静态采集，尚未包含实时更新的钓鱼网站信息，模型对最新攻击手段的适应性有待进一步验证；另一方面，SVM模型的调参过程相对复杂，且在面对大规模数据时计算资源消耗较高。未来研究可考虑引入更多动态特征、采用深度学习或集成学习方法进行对比，进一步提升模型性能和应用广度。

综上，基于SVM的钓鱼网站检测模型在本研究中取得了良好的分类效果，能够为实际网络安全防护提供有效的技术支持和方法借鉴，具有较高的应用价值和推广潜力。未来工作将聚焦于模型优化、多源数据融合以及实时检测系统的开发，同时考虑以下几个方向的深入研究：一是结合多模态特征（包括文本内容、视觉截图、用户行为日志等），利用深度学习与传统算法集成，进一步提升检测的精确性和适应性；二是针对新型对抗样本和变种钓鱼技术，探索对抗训练和模型鲁棒性增强方法，提高系统的安全防御能力；三是引入自动化超参数调优、特征选择及模型解释技术，提升模型训练效率和可解释性。未来的探索将为构建更加智能化、精确化的网络钓鱼检测体系奠定坚实基础，以期为互联网用户提供更加高效、安全的网络使用体验。

**参 考 文 献**

[1] Jakobsson M, Myers S. Phishing and countermeasures: understanding the increasing problem of electronic identity theft[M]. Wiley-Interscience, 2006.

[2] Hong J. The state of phishing attacks[J]. Communications of the ACM, 2012, 55(1): 74-81.

[3] 张伟, 李娜, 王磊. 基于URL和域名特征的钓鱼网站检测方法研究[J]. 计算机科学与探索, 2019, 13(4): 768-778.

[4] Kumar R, Kumari P, Mallick P K. Phishing detection based on multi-feature fusion and machine learning approach[C]//International Conference on Advanced Computing and Communication Technologies. IEEE, 2018: 287-292.

[5] 李强, 陈刚, 赵敏. 基于支持向量机的钓鱼网站检测模型[J]. 计算机工程与应用, 2020, 56(7): 145-152.

[6] 王芳, 刘洋. 钓鱼网站识别中支持向量机与神经网络的性能比较[J]. 软件学报, 2018, 29(6): 1741-1752.

[7] 赵丽. 多源特征融合在钓鱼网站检测中的应用研究[D]. 北京邮电大学, 2019.

**附录**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score, roc\_auc\_score, roc\_curve

import matplotlib.pyplot as plt

import seaborn as sns

plt.rcParams['font.family'] = 'SimHei'

plt.rcParams['axes.unicode\_minus'] = False

# 读取数据

data = pd.read\_csv('Phishing\_Websites\_Data.csv')

# 分割特征和标签

X = data.drop(columns=['Result'])

y = data['Result']

# 处理标签，将-1变为0，符合sklearn二分类习惯

y = y.replace(-1, 0)

# 划分训练集和测试集

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42, stratify=y)

# 创建SVM分类器，默认核为RBF

clf = SVC(kernel='rbf', probability=True, random\_state=42)

# 训练模型

clf.fit(X\_train, y\_train)

# 预测测试集

y\_pred = clf.predict(X\_test)

y\_proba = clf.predict\_proba(X\_test)[:, 1]

# 评估模型

print("准确率:", accuracy\_score(y\_test, y\_pred))

print("混淆矩阵:\n", confusion\_matrix(y\_test, y\_pred))

print("分类报告:\n", classification\_report(y\_test, y\_pred))

# 绘制混淆矩阵图

cm = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(6,5))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False,

xticklabels=['非钓鱼(0)', '钓鱼(1)'],

yticklabels=['非钓鱼(0)', '钓鱼(1)'])

plt.xlabel('预测类别')

plt.ylabel('真实类别')

plt.title('混淆矩阵')

plt.show()

# 绘制ROC曲线

fpr, tpr, thresholds = roc\_curve(y\_test, y\_proba)

auc\_score = roc\_auc\_score(y\_test, y\_proba)

plt.figure(figsize=(6,6))

plt.plot(fpr, tpr, label=f'ROC 曲线 (AUC = {auc\_score:.2f})')

plt.plot([0,1], [0,1], 'k--')

plt.xlabel('假阳性率 (False Positive Rate)')

plt.ylabel('真正率 (True Positive Rate)')

plt.title('SVM 钓鱼网站检测 ROC 曲线')

plt.legend(loc='lower right')

plt.grid()

plt.show()

import pandas as pd

import matplotlib.pyplot as plt

# 读取数据

data = pd.read\_csv('Phishing\_Websites\_Data.csv')

# 替换标签便于统计

data['Result'] = data['Result'].replace({-1: 0})

# 描述性统计：类别分布

label\_counts = data['Result'].value\_counts()

print("标签分布：")

print(label\_counts)

# 条形图：展示钓鱼网站与非钓鱼网站数量

plt.figure(figsize=(6,4))

label\_counts.plot(kind='bar', color=['skyblue', 'orange'])

plt.xticks(ticks=[0,1], labels=['非钓鱼网站(0)', '钓鱼网站(1)'])

plt.ylabel('数量')

plt.title('钓鱼网站与非钓鱼网站样本数量分布')

plt.grid(axis='y')

plt.show()

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# 读取数据

data = pd.read\_csv('Phishing\_Websites\_Data.csv')

# 标签处理，-1替换为0

data['Result'] = data['Result'].replace(-1, 0)

# 选择分析的变量

variables = [

'having\_IP\_Address', 'URL\_Length', 'Shortining\_Service',

'SSLfinal\_State', 'web\_traffic', 'Result'

]

# 变量取值分布打印

for var in variables[:-1]:

print(f"变量 {var} 的取值计数：")

print(data[var].value\_counts())

print()

# 按类别计算各变量的均值

print("按类别分组计算变量均值：")

print(data.groupby('Result')[variables[:-1]].mean())

# 条形图：各变量按类别的分布对比

plt.figure(figsize=(16, 10))

for i, var in enumerate(variables[:-1]):

plt.subplot(2, 3, i + 1)

sns.countplot(x=var, hue='Result', data=data, palette='Set1')

plt.title(f'{var} 按钓鱼(1)/非钓鱼(0)类别分布')

plt.tight\_layout()

plt.show()

# 扇形图：展示每个变量的整体取值比例

plt.figure(figsize=(16, 10))

for i, var in enumerate(variables[:-1]):

plt.subplot(2, 3, i + 1)

counts = data[var].value\_counts()

plt.pie(counts, labels=counts.index, autopct='%1.1f%%', startangle=90)

plt.title(f'{var} 总体分布')

plt.tight\_layout()

plt.show()