单位代码

学　　号 78066011

分 类 号

北京航空航天大學

BEIHANG UNIVERSITY

毕业设计(论文)

基于机器学习的钓鱼网站检测系统设计与实现

学　院　名　称　计算机学院

专　业　名　称　计算机科学与技术

学　生　姓　名　蔡佩津

指　导　教　师　傅翠娇

2022 年 6 月

基于机器学习的钓鱼网站检测系统设计与实现

蔡佩津

北京航空航天大学

北京航空航天大学

本科生毕业设计（论文）任务书

I、毕业设计（论文）题目：

基于机器学习的钓鱼网站检测系统设计与实现

II、毕业设计（论文）使用的原始资料（数据）及设计技术要求：

本文的技术要求是设计并实现一个基于机器学习的检测钓鱼网站系统。该系统主要目标是建立一个能够返回是否是钓鱼网站的分类结果，使得网民在网上冲浪更加安全。

III、毕业设计（论文）工作内容：

本文的主要内容如下：1）网络钓鱼的背景介绍和研究意义 2）对 url 数据预处理分析和理解；3）对监督机器学习算法分析和理解；4）实验结果分析与比较；5）实现了钓鱼网站检测系统部署

IV、主要参考资料：

[1]Sharifi, et al. A Phishing Sites Blacklist Generator[A]. IEEE/ACS International Conference on Computer Systems and Applications [C]. 2008:840-843.

[2] Shaikh, et al. A Literature Review on Phishing Crime, Prevention Review and Investigation of Gaps[A]. 10th International Conference on Software, Knowledge, Information Management & Applications IEEE [C]. 2016:9-15.

[3]Kalaharsha, et al. Detecting Phishing Sites—An Overview[D]. India: University of Hyderabad, 2021.

计算机 学院（系） 计算机科学与技术 专业类 182511 班

学生 蔡佩津

毕业设计（论文）时间： 2022 年 02 月 27 日至 2022 年 5 月 25 日

答辩时间： 2022 年 5 月 9 日

成 绩：

指导教师： 傅翠娇

兼职教师或答疑教师（并指出所负责部分）：

系（教研室） 主任（签字）：

注：任务书应该附在已完成的毕业设计（论文）的首页

本人声明

我声明，本论文及其研究工作是由本人在导师指导下独立完成的，在完成论文时所利用的一切资料均已在参考文献中列出。

作者：蔡佩津

签字：

时间：2022 年 5 月 25 日

基于机器学习的钓鱼网站检测系统的设计与实现

学　　生：蔡佩津
指导教师：傅翠娇

摘　　要

由于互联网的快速发展，用户的偏好从传统的购物转向了电子商务。如今，犯罪分子不再是抢劫银行，而是试图通过一些特定的技巧在网络空间中寻找犯罪机会。攻击者利用互联网的匿名结构，推出网络钓鱼等新技术，通过使用虚假网站来欺骗受害者，以收集账户 ID、用户名、密码等重要信息。了解网络地址是否合法或网络钓鱼是一个非常具有挑战性的问题，因为其基于语义的攻击结构主要是利用了网络用户的漏洞。

现在有很多应用程序可用于网络钓鱼检测。然而，与预测垃圾邮件不同，只有少数研究比较了机器学习技术在预测网络钓鱼方面的作用。在所有网络钓鱼预防技术中，基于 URL 的分类是主要的一种，在这方面，传统的黑白名单和基于网页图像相似度技术在快速变化的网络钓鱼中已经落后了。因此，需要用一种更智能的技术来保护用户避免受网络攻击，有效和准确地帮助网民判断网页是否合法，使得网络交易更加安全。

本文开发了一种高效的钓鱼 URL 检测机制，该检测机制的主要工作包括数据预处理，URL 特征分析及关键词可视化，确定适用于模型识别的特征，分别利用逻辑回归算法，多项式分布朴素贝叶斯算法及梯度增强树算法三种监督式机器学习算法进行训练，模型超参数优化和交叉验证。通过不同机器学习分类算法之间的实验对比，逻辑回归算法在钓鱼 URL 识别上效果最优，准确率为 96.8%，其次是多项式分布朴素贝叶斯算法，准确率为 96.7%，并且逻辑回归算法的 AUC-ROC 面积为 0.996。最后，本文使用逻辑回归模型部署到 Streamlit 网站上实现了钓鱼网站检测系统，测试结果表明该网站可以识别出钓鱼网站。

关键词：机器学习，URL，网络钓鱼，电子商务

Design and Implementation of a Phishing Website Detection System Based on Machine Learning

Author: Quinna Jodanti

Supervisor: 傅翠娇

Abstract

Due to the rapid growth of the Internet, users change their preference from traditional shopping to the e-commerce. Instead of bank robbery, nowadays, criminals try to find their victims in the cyberspace with some specific tricks. By using the anonymous structure of the internet, attackers set out new techniques, such as phishing, to deceive victims with the use of false websites to collect confidential information such as account IDs, usernames, passwords, etc. Understanding whether a web is legitimate or phishing is a very challenging problem, due to its semantic-based attack structure, which mainly exploits the computer user's vulnerabilities.

There are many applications available for phishing detection. However, unlike predicting spam, there are only a few studies that compare machine learning techniques in predicting phishing. Among all the technique of Phishing prevention, URL-based classification is the main one. In this area, most Anti-Phishing tools rely on blacklist or whitelist. These tools have some effect, but falling behind from new fast-changing phishing. The key to solving this is to add learning ability to these tools.

This paper develops an efficient phishing URL detection mechanism. The main work includes data preprocessing, URL feature analysis and keyword visualization, establishing features suitable for model recognition, training with three supervised machine learning algorithms: logistic regression algorithm, multinomial naive bayes algorithm and gradient boosting tree algorithm, model hyperparameter optimization and cross validation. Through the experimental comparison between different machine learning classification algorithms, logical regression algorithm has the best result on Phishing URL recognition, with an accuracy of 96.8%, followed by the multinomial naive bayesian algorithm, with an accuracy of 96.7%, and the AUC-ROC area of the logical regression algorithm is 0.996. Finally, this paper uses logistic regression model to deploy to streamlit website to implement the phishing website detection system.

Keywords: Machine Learning, URL , Phishing , E-commerce

目 录

1 绪论

1.1 课题背景与意义

近年来，互联网和云技术的快速发展导致电子交易显著增加，除了丰富了人们的生活之外，同时也吸引了大量犯罪分子进行网络钓鱼攻击。网络钓鱼目前还没有达成一致的定义。在大多数定义中网络钓鱼诈骗的目标是窃取个人的敏感信息。不同攻击的媒体可能会有不同的攻击设置。例如，Pharming 是一种网络钓鱼，攻击者将用户误导到欺诈站点或代理服务器，通常通过域名系统(DNS) 劫持或病毒。在这种情况下，攻击者可以通过获取网站域名并将该网站的流量重定向到钓鱼网站。

为了避免网络钓鱼，1) 用户应了解网络钓鱼网站[1]，2) 建立一个网络钓鱼网站黑名单，该黑名单要求了解网站是否被检测为网络钓鱼，或者 3) 使用机器学习和深度神经网络算法在其早期出现时进行检测。在以上三种方法中，基于机器学习的方法被证明是最有效的。即便如此，在线用户仍被困在网络钓鱼网站中泄露敏感信息。

1.2 网络钓鱼形式

网络钓鱼攻击早在互联网出现之初就已经存在。钓鱼者在 20 世纪 90 年代中期传播了第一次网络钓鱼攻击，利用美国在线 (AOL) 服务窃取密码和信用卡信息。虽然现代攻击使用类似的社会工程模型，但攻击者使用的是更先进的战术。一种常见的网络钓鱼攻击形式是网络钓鱼者建立一个与合法网页很相似的网页，并利用特殊的通信方法把指向虚假网页的链接发给目标。一些用户无法辨别网址的真假而输入重要信息，比如帐户密码。钓鱼者从网络后台收集用户的个人信息，盗取用户电子银行资产，并将贩卖用户的个人信息以获得不法收益。钓鱼攻击的主要方式有如下几类:

1）**发送电子邮件,** 也称为 “欺骗式网络钓鱼”，电子邮件网络钓鱼是最知名的攻击类型之一。钓鱼者平时利用用户的恐惧和紧迫感，例如从邮箱通知用户帐户将被限制或暂停的各种理由来勾引用户点击链接要求提交个人敏感信息。这些垃圾邮件也可以体现为中奖，模仿某些知名的品牌，等等。

2）**HTTPS 网络钓鱼,** 超文本传输协议安全 (HTTPS) 通常被认为是“安全”链接，因为它使用加密来提高安全性。大多数合法组织现在使用 HTTPS 而不是 HTTP，因为它建立了合法性。然而，钓鱼者现在正利用 HTTPS 链接来放入钓鱼电子邮件。

3）**钓鱼短信**，也被称为“Smishing”，是 SMS 短信和网路钓鱼术语的组合[2]。钓鱼者会通过短信发送假信息引导受害者点击链接至钓鱼者创建的网站。与邮件相比，使用短信形式发送信息更加个人化，使受害者不那么警觉。

4）**鲸鱼网络钓鱼**，也称为“CEO 欺诈”。攻击者使用社交媒体或者公司网站来找组织 CEO 或者其他高级领导成员的姓名。然后，他们使用类似的电子邮件地址冒充该人。电子邮件通常要求汇款或要求收件人查看文件。

1.3 网络钓鱼危害

根据互联网世界统计数据，2014 年全球互联网用户总数为 29.7 亿；也就是说，超过 38%的世界人口使用互联网。黑客利用不安全的互联网系统，可以愚弄不知情的用户，使其落入钓鱼欺诈的陷阱。网络钓鱼电子邮件用于在互联网上诈骗个人和金融组织。

根据全球反钓鱼工作组织(APWG)2014 年第一季度的报告，有记录以来第二高的网络钓鱼攻击数量发生在 2014 年 1 月至 3 月，支付服务是最受关注的行业。2014 年下半年，观察到 123,972 起独特的网络钓鱼攻击。2011 年，总财务损失为 12 亿美元，2013 年增至 59 亿美元。

根据 APWG(Anti-Phishing Working Group)的数据，用户对网络钓鱼网站的认识逐年提高，但网络钓鱼网站的数量及其造成的损失迅速增长。在 2016 年第四季度 APWG 报告中，2016 年 10 月的网络钓鱼活动趋势有 89,232 个站点被检测为网络钓鱼站点，而在 2016 年 11 月和 2016 年 12 月，分别有 118,928 个和 69,533 个网站被指示为网络钓鱼网站, 如图 1.1 所示。

在第二季度 2021，APWG 创始成员 OPSEC 安全发现网络钓鱼攻击针对金融机构的攻击最为普遍[3]，占所有攻击的 29.2%，高于 2020 年第 4 季度所有攻击的 22.5%，如图 1.2 所示。

图 1.1 2016 年第四季度网络钓鱼活动

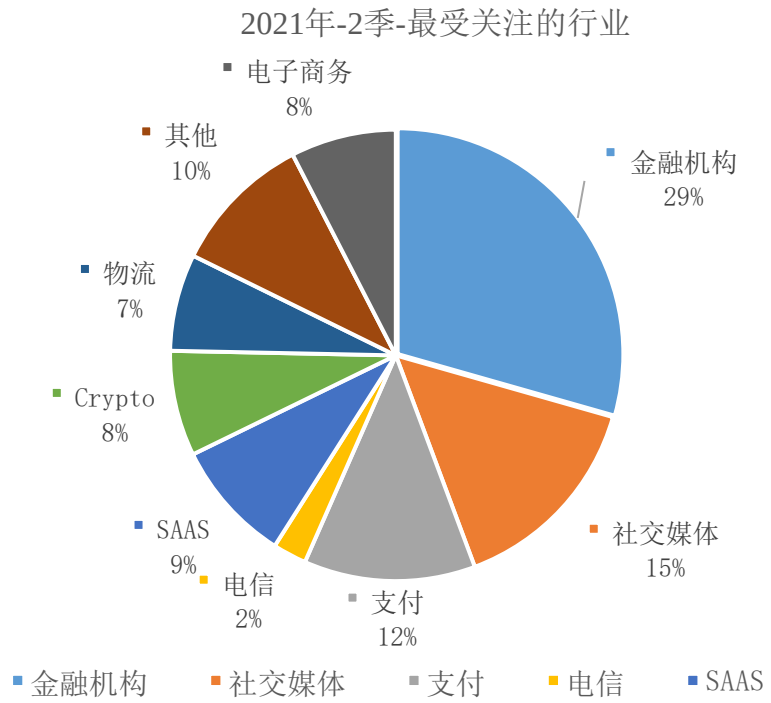

图 1.2 最受关注的行业

1.4 国内研究现状

社会工程学的角度和网络安全技术的角度是目前预防钓鱼的主要研究方向。简而言之，第一个旨在教网络用户如何识别和处理虚拟网站，第二个旨在使用机器能够自动快速地识别钓鱼网站。本文主要涉及第二方案，即网络安全预防的技术解决方案。在该领域，目前有不少科学家和组织做出了很好的工作，可以看出主要内容有下几方面：

1.4.1 基于列表的识别技术

基于列表的识别技术可以分别为黑白名单。这是传统方法或面向数据库的方法。该方法是使用最广泛的一种技术方案，反应时间和检测精度非常高。在黑名单技术中，有一个黑名单数据库来存储危险网站。当有用户查看网页时，并将 URL 于黑名单数据库当中的进行比较，如果匹配到则浏览器会提示用户或直接阻止用户访问，如图 1.3 所示。例如，谷歌浏览器可以通过自动更新黑名单功能来有效阻止这些恶意网站的网络钓鱼攻击，用户还可以使用谷歌安全浏览 API 检查他们访问的网站安全性。虽然这种基于黑白名单技术从易用性上是最佳的，但这种技术的局限性是需要等到出现受害者时才可以开始对黑名单数据库进行更新。

在白名单方法中，合法 URL 存储在数据库用于检查新的 URL[5]。当用户输入 URL 时，首先检查数据库的 URL，如果没有该 URL 的记录，则检查该 URL 的全部信息，例如域名，年龄，SSL 证书，链接到外网的超链接，确认是安全网站后将其存储在数据库中。常规白名单工具应该经常动态地更新整个白名单数据库以保持永久可用，这可能会对系统造成数据库造成冗余，带来了很大程度的维护成本。

图 1.3 Firefox 恶意网站提示

1.4.2 基于网页文本特征的检测

CANTINA 是一个利用分析网页内容来监测互联网钓鱼的工具，主要应用词频–逆文档频率的 TF-IDF 信息检索算法并创建一个词法签名[6]。词频-逆文档频率(TF-IDF)是

在文本挖掘领域中使用的一种算法，它可用与计算常用的每个单词的权重。简单来说，TF-IDF 方法用于找出一个单词在文档中出现的频率。然而，这项研究是有局限性的，因为它只对英语敏感。其增强模型被命名为 CANTINA + ，它包括基于 HTML 的 15 个属性，该系统达到了 92% 的准确率，但它可能会产生大量误报。

1.4.3 基于网页图像的相似度检测

为了避免文本检测，网络钓鱼者可能会使用图像而不是文本来创建网络钓鱼网页。因此，网络钓鱼网页变得更加复杂，并对检测基于文本的网络钓鱼任务提出了挑战。针对这个问题，Anthony Y.Fu，Liu Wen Yin 使用不同的方法计算网页的视觉相似性。该方法首先将 HTML 网页转换为图像，然后使用 EMD 方法对图像的签名进行相似性计算。

EMD[13]是一种在两个签名之间评估距离（相异性）的方法，签名是一组特征及其相应的权重。EMD 是基于众所周知运输问题的解决方案。假设有 m 个生产者，每个生产者都有一个相应的货物重量。将生产者集 P 表示为：

$$P = \{(p_1, w_{p1}), (p_2, w_{p2}), \dots, (p_m, w_{pm})\} \tag{1.1}$$

假设也有 n 个消费者，每个消费者带有一个权重，表明消费者所需要使用的产品数量。将消费者集 C 表示为：

$$C = \{(c_1, w_{c1}), (c_2, w_{c2}), \dots, (c_n, w_{cn})\} \tag{1.2}$$

计算 EMD 的时候第一个任务是找到流量矩阵(flow matrix):

$$F = [f_y | 1 \le i \le m, i \le j \le n] \tag{1.3}$$

该流量矩阵可以包含从特定的 P 到C 运输的产品量，使可以满足从 P 到 C 的运输产品数量尽可能多，总运输费应为最小化。总费用表示为$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{ij}$,则可以在以下约束条件下得到 F 的值：

$$s.t. \begin{cases} f_{ij} \ge 0 \\ \sum_{j=1}^{n} f_{ij} \le w_{pi} \\ \sum_{i=1}^{m} f_{ij} \le w_{cj} \\ \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = Min(\sum_{i=1}^{m} w_{pi}, \sum_{j=1}^{n} w_{cj}) \end{cases} \tag{1.4}$$

这是一个线性规划的问题。求解后得到 F，然后再计算出 EMD。EMD 距离表示如下：

$$EMD(P,C,D)=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(f_{ij}d_{ij})}{\sum_{i=1}^{m}\sum_{j=1}^{n}f_{ij}} \tag{1.5}$$

该方案还具有以下几个缺陷，第一，必须预先存储经常遭到入侵的目标网页的图像特征，并进行对比。如果目标页面数量过多，保存和检查这些数据会影响工作效率。由于在一个合法的页面布局中的颜色变更比较频繁或者同一个主题同时出现在不同的页面上，都会大大降低了 EMD 算法的准确性，进而影响了检测的准确性。

1.5 课题研究目标与内容

根据网络反钓鱼的研究现状以及互联网的快速发展，使用传统方法检测网络钓鱼是有限的。同时，网页内容检测在检测对象方面非常准确，但应用效率低，因此本文认为使用 URL 检测的研究具有重要意义，以下是主要原因：

1）适用性。无论是钓鱼邮件检测还是钓鱼网页检测，具有钓鱼特征 URL 都是主要的分析对象。URL 分类和检测还可以应对目前愈加严重的通过短信的钓鱼扩散。

2）统一标准。URL 是具有特定标准格式的字符，无论任何语言都会使用统一的 URL 来描述资源的地址。

3）节能高效。与传统的基于内容的钓鱼检测相比，使用 URL 从计算算法和磁盘空间消耗方面的空间要快和少得多，它根本不需要扫描、判定和学习整个邮件或网页。

本文的研究目标是在为预测钓鱼网站而创建的数据集上训练机器学习模型。研究工作主要体现在以下几个方面：

a）从开源平台收集网站的钓鱼和良性 URL 以形成数据集，并从中提取所需的 URL。

b) 研究监督机器学习检测算法在 URL 特征学习上的典型应用。

c) 使用 EDA (exploratory data analysis)技术对数据集进行分析和预处理。

d) 在数据集上运行选定的机器学习模型算法。

e) 对每个模型的性能水平进行测量和比较。

f) 实现检测钓鱼网址系统部署。

1.6 论文组织结构

本文共分五章，第 1 章对研究进行了概述。主要包括研究背景，国内外研究现状以

及研究目标。在第 2 章，有钓鱼 URL 的特征分析，其中包括对 URL 进行预处理和可视化。第 3 章主要解释机器学习的概念，在本章中，将列出经典的机器学习算法，并对每个算法进行详细描述。第 4 章包括对已训练好的模型进行比较并给出结果和准确率。第 5 章是模型线上部署，将描述实现方法。

2 钓鱼 URL 特征分析

2.1 统一资源定位地址(URL)分析

URL(Uniform Resource Locator)代表统一资源定位器，也称为是网址。它是对 web 资源的引用，用于指定远程服务器中的资源位置。资源可以是网页、文本文件、电子邮件、图像和数据库访问。网页有许多链接以提供更多信息，但链接到恶意网页的网页本身很可能是恶意的。URL 的结构如图 2.1 所示。

图 2.1 统一资源定位符组件

第一部分，http://，被称为协议。该协议告诉服务器所请求的文件类型。HTTP 代表超文本传输协议，主要用于请求 HTML(超文本标记语言)文件。除了 HTTP，还有很多协议，比如 FTP、SMTP 等。但是现在由于大多钓鱼网页都使用 https，本文把协议部分去掉。

第二部分，主机名只不过是一个 IP 地址，它有助于识别和记数字地址的互联网资源的名称。主机名由子域(www)，二级域(exampleurl)和顶级域(.com)组成。浏览器使用二级域名向 web 服务器检查此网站是否存在的名称。第三级域名被浏览器用于解析请求站点位置的名称服务器。最常见的 TLD 是 .com，.org，.net，等。一些网站在顶级域右边有另一个部分，例如 .uk，.us 这被称为国家代码顶级域(ccTLD)。二级域名(SLD)和三级域名(TLD)一起拥有网站的权限，称为域权限或域链接。主机名后面跟着端口号，网页的默认端口号是：80，但这可以省略掉。

第三部分，路径就像是将文件导航到个人计算机中的文件夹中。“/info/page1.html”是指向“page1.html”资源的路径，其中目录为“/info/”。目录为用户和搜索引擎提供了解资源部分的信息。“?param1&param2” 是提供给 Web 服务器的额外参数。这些参数是以问号“？”开头，并用 “&” 符号分隔。

2.2 数据采集与预处理

为了做出预测，首先本文从 PhishTank，Phishstorm，Kaggle 等开源平台收集网络钓鱼数据。本文收集的数据有 549,346 个唯一条目，共有 2 列。标签列是预测列，有两种类别，good 表示 URL 是合法的，bad 表示 URL 包含恶意链接，也就是网络钓鱼网站。在本文将 URL 数据提供给机器学习模型之前，需要先对数据进行预处理。

如图 2.2 所示，数据中有 392,924 个标记为合法，156,422 个标记为钓鱼。有了数据之后，本文需要对 URL 进行矢量化。首先，本文使用标记器(tokenizer)拆分 URL，然后创建一个名 text_tokenized 的新列。在得到标记化的单词后，本文使用 SnowballStemmer 库来获得根单词。Snowball 是一种返回根词的小字符串处理语言。本文将根单词保存到 text_stemmed 列，然后连接 text_stemmed 并将其保存到 text_sent 列。为了方便之后模型训练，本文使用 CountVectorizer 库把 text_sent 转换为标记计数向量。接下来，本文把 good 和 bad 标签做了标签编码，good 表示为 0，bad 表示为 1。

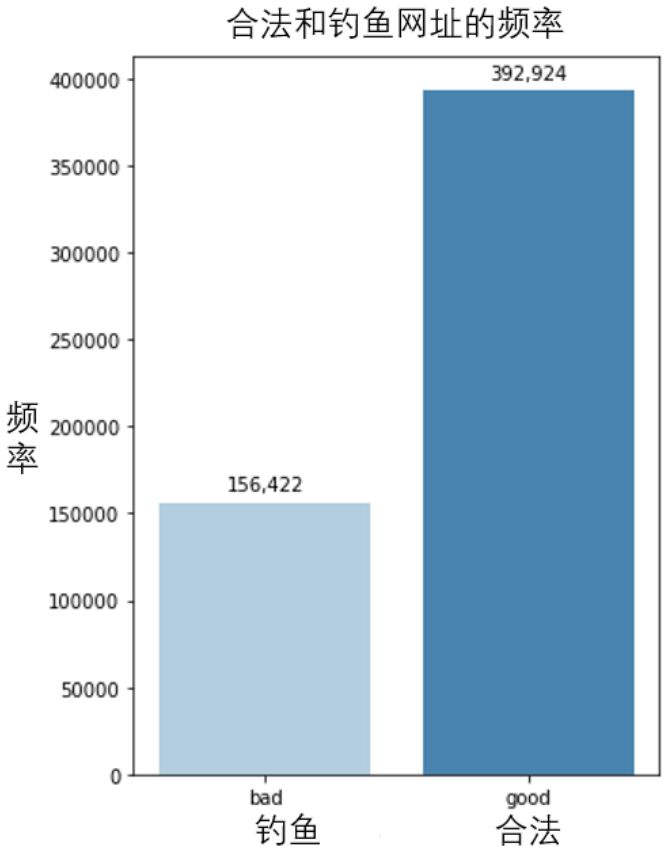

图 2.2 合法和钓鱼网址频率

2.3 关键词可视化

在进行可视化之前，为了得到更清晰的数据，本文先把停用词(Stop Words)删掉。停用词是在自然语言数据处理之前或之后过滤掉的停用列表中的任何词。然后，使用 WordCloud 库来对数据进行可视化，如图 2.3 所示，星形图显示良性 url 数据当中最常用的单词，字体大小表示出现频率，字体越大表示出现率越多。图 2.4 表示在钓鱼 url 最常用的单词。

图 2.3 WordCloud 显示合法 URL 最常用的词

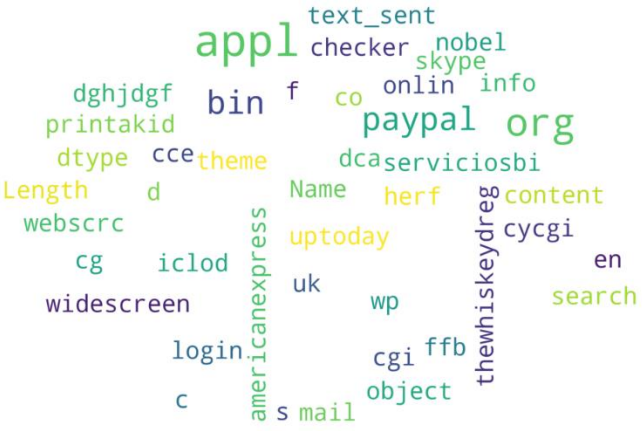

图 2.4 WordCloud 显示钓鱼 URL 最常用的词

2.4 测试数据与训练数据分开

模型训练之前，为了之后方便评估模型的泛化能力，首先本文把数据分为训练集和测试集。测试集是代表没有参与构建分类器的数据集，该数据是问题的代表性样本。在数据源充足的情况下，可以取大样本进行训练，也可以取其他不同但独立的大样本数据

进行测试。测试样本的误差将反映其真实的未来表现。虽然当训练样本超过一定限度时性能提升会变慢，但一般来说测试样本越大，误差估计越准确。本文对样本数据随机分为训练集和测试集。测试集占数据的 20%，训练集占数据的 80%。训练集用于训练模型，测试集用于验证模型的实际效果。数据分割时，本文使用 stratify 参数为了避免数据不平衡。接下来，对模型进行训练。

2.5 本章小结

本章首先对网络钓鱼的 URL 特征进行了分析，然后为了实现模型训练，本文进行数据采集及预处理。接着，本文使用 WordCloud 库把在合法和钓鱼网站经常用的 URL 进行可视化。最后，本文把数据分成数据集与训练集。下一章对机器学习算法进行研究与分析。

3 机器学习算法研究与分析

3.1 机器学习概述

机器学习是人工智能（AI）的一个子领域[8]，它为系统提供了在无需明确编写算法的情况下从数据和经验自动学习和改进的能力，从而做出分类、预测等行为。学习过程使用特定的机器学习算法，不同机器学习算法具有不同效率和规格。不仅仅是人类需要学习，而机器也需要通过学习来增加能力，帮助人类解决问题。

如今，随着科学和商业中数据驱动方法的增加，机器学习在许多领域得到了广泛的应用。从科学研究到计算机科学中的语音识别，以及银行业欺诈交易的检测。

目前，机器学习技术领域相关的研究工作方向主要集中在以下三个方面：

1）面向任务的研究。

2）认知模型。

3）理论分析。

3.1.1 机器学习框架

一个典型的机器学习系统主要由数据采取、数据预处理、特征选择和抽取、模型训练和分类与预测，如图 3.1 所示。

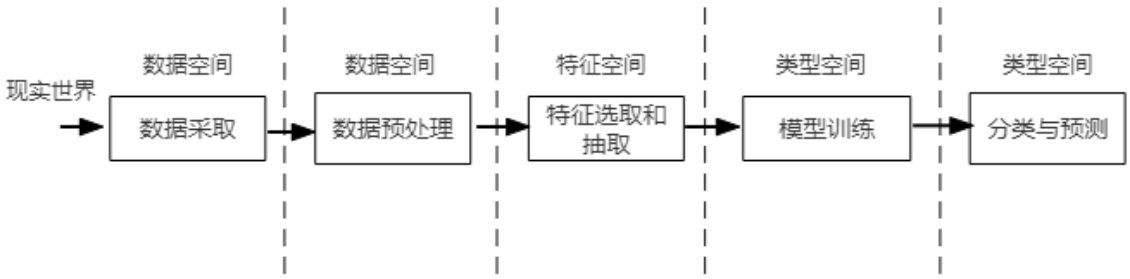

图 3.1 典型机器学习流水线

数据采取(Data Acquisition)。机器学习中，信息以数据集的形式提供给学习者。每个数据示例都是要探索的独立示例。

数据预处理(Data Preprocessing)。在实际生活中所收集到的数据目前还不可以直接应用到机器学习算法中，为了能够提高处理的效率，需要进一步引入数据预处理的环节。数据预处理是一种用于将原始数据转换为有用且高效的格式的技术。这一举措是必要的，因为原始数据通常不完整且格式不一致。数据本身的质量与任何涉及数据分析的项目的成功直接相关。预处理本身涉及数据验证和插补。验证目的是评估过滤数据的完整性和

准确性水平。

特征选择与和抽取(Feature Selection and Extraction)。选择特征时需要遵循一些基本原则。例如，选择的特征应该足以代表模式，而选择的特征的数量应该尽可能少，以便后续的模型训练和分类任务能够高效完成。

模型训练(Model Training)。模型训练是机器学习最核心的一个阶段，在此阶段尝试将权重和偏差的最佳组合拟合到机器学习算法种以得到最小化损失函数。假设将模型学习区分出来两种动物，如果只是用数学术语来看待模型，那么有两个输入，也称为特征，即颜色和形状。这两个特征有系数，这些系数称为是特征的权重，还会分别涉及到另一个常数或 y 截距，这就被称为模型的偏差。然后，使用训练数据集中的不同条目重复迭代，直到模型达到所需的准确度。

分类与预测(Classification and Prediction)。机器学习过程的最后一步是预测或分类。这是模型为实际应用做好准备的最后一步。未来该模型面临的主要挑战是，它能否在多个不同场景中匹配人类的判断。

3.2 学习算法分类

从功能与应用方式角度来看，机器学习算法还可以具体地分为以下几类：

1)**监督学习算法(Supervised learning)。**监督学习是最普遍的机器学习范式。这算法需要先提供已知的训练数据集及其相应的标签，算法并生成一个推断函数，以最终对一些新未知标签的数据进行预测。这与教孩子使用闪存卡非常相似。监督学习模型可以进一步分为涉及预测类别标签的分类和涉及预测数值的回归。

2)**非监督学习算法(Unsupervised learning)**。在非监督学习中，训练使用未标签和未分类的数据。因此，系统的算法在没有事先训练的情况下对数据起作用。它可以学习某种方式对数据进行分组，集群或组织，从而使人类能够介入并理解组织的数据。

3)**半监督学习算法(Semi-supervised learning)。**实际上，不是所有数据都带标签，有时候带标签的数据只有一半。为了解决这种情况，可以使用半监督学习算法。该算法是介于监督学习和非监督学习之间。算法的工作方式是使用非监督学习来发现结构，并将该数据重新插入监督学习算法，然后使用该模型做出新预测。

4)**强化学习(Reinforcement learning)**。强化学习是一种基于奖励期望行为和惩罚不期望行为的机器学习训练方法。一般来说，强化学习智能体能够感知和解释其环境，采取行动并通过反复试验进行学习。

基于目前钓鱼检测研究现状的特征，很容易发现钓鱼检测是属于监督学习的机器学习算法。下一章将继续介绍检测钓鱼的一些常用分类算法，也会进行详细对比及分析。

3.3 逻辑回归算法

逻辑回归(Logistic Regression, LR)是一种统计学方法，其通过将数据输入 logit 方程来得到某件事情的发生概率[9]，如是否会发生降雨。虽然被称为回归，但其实实际上是分类模型，并常用于二分类。LR 分析方法来源于 logistic 方程:

$$f(z)=\frac{e^{z}}{e^{z}+1}=\frac{1}{1+e^{-z}} \tag{3.1}$$

方程(3.1)的输入为 z，输出为 *f(z)*。Logistic 方程的优势在于它可以处理从负无穷到正无穷的任何输入，而将输出限制在 0 和 1 之间，如图 3.2 所示。

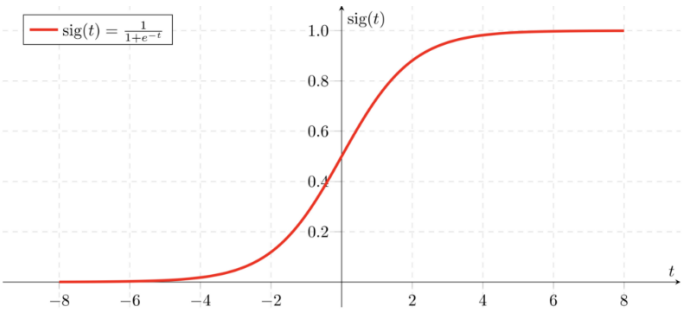

图 3.2 逻辑回归 sigmoid 函数

对于逻辑回归，如果用 x 表示训练数据，θ表示模型参数，h 表示预测的输出，y 表示训练数据的标签。

因为$h_\theta(\mathrm{x})=\mathrm{f}(\theta^{T}x)$，把他带入方程(3.1)得假设函数:

$$h_\theta(\mathrm{x})=\frac{1}{1+e^{(-\theta^{T}x)}} \tag{3.2}$$

假设函数返回是 y=1 的概率，给定 x，由 θ参数化，写为 $\mathrm{P(y=1|x;}\theta)=h_\theta(\mathrm{x})$，返回 y=0 的概率写为 $\mathrm{P(y=0|x;}\theta)=1-h_\theta(\mathrm{x})$。决策边界可以描述为：

如果 $\theta^{T}x\geq 0\rightarrow h(x)\geq 0.5$，预测为 1，表示预测为钓鱼网站。如果 $\theta^{T}x<0\rightarrow h(x)<0.5$，预测为 0，表示预测为合法网站。

3.3.1 逻辑损失函数

在逻辑回归不能用最小二乘误差作为损失函数因为使用最小二乘误差将导致具有局部最小值的非凸图。直观地说，当预测为 1 而实际为 0 时，以及当预测为 0 而实际为 1 时，希望分配很大的惩罚。逻辑回归的损失函数正是这样做的，称为逻辑损失。如

果 y=1，如图 3.3 左所示，预测为 1 时，代价为 0，预测为 0 时，学习算法惩罚的代价非常大。类似地，如果 y 为 0，如图 3.3 右所示，预测 0 没有惩罚，但预测 1 具有很大的惩罚。

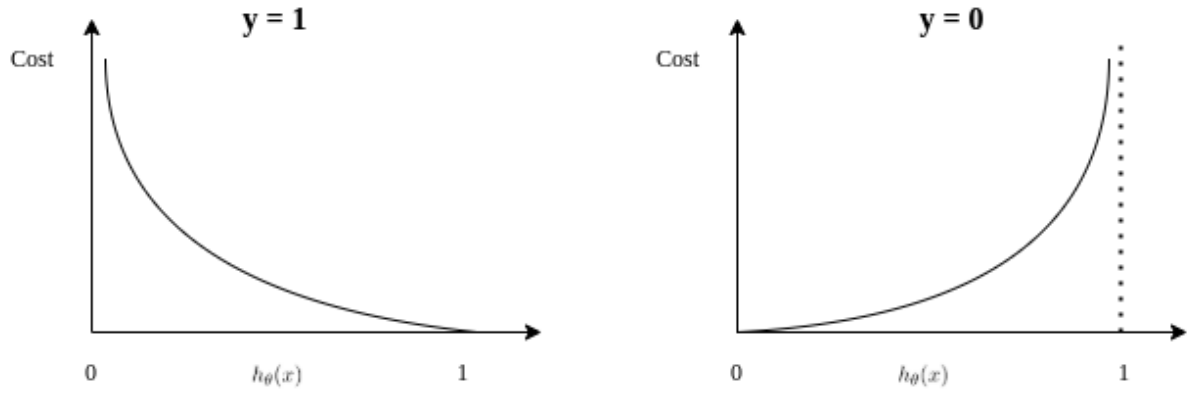

图 3.3 逻辑回归惩罚度

$$Cost(h_\theta(x),y)=\begin{cases}-\log(h_\theta(x))\text{，当 } y=1\\-\log(1-h_\theta(x))\text{，当 } y=0\end{cases} \tag{3.3}$$

当 y=1 时，$cost(h_\theta(x),1)=-log(h_\theta(x))$，$h_\theta(x)$越大，则损失函数越小

当 y=0 时，$cost(h_\theta(x),0)=-log\,(1-h_\theta(x))$，$h_\theta(x)$越小，则损失函数越小

这个损失函数的好处是，虽然是分别用 y=1 和 y=0 看的，但是为了方便计算可以写成一个公式：

$$Cost(h_\theta(x),y)=-(ylog(h_\theta(x))+(1-y)\log(1-h_\theta(x))) \tag{3.4}$$

所以模型的成本函数是所有训练数据样本的总和：

$$J(\theta)=\frac{1}{m}\sum\left(Cost\left(h_\theta\left(x^{(i)}\right),y^{(i)}\right)\right) \tag{3.5}$$

m 代表样本数，把方程(3.4)带入方程(3.5)得：

$$J(\theta)=-\frac{1}{m}\sum(y^{(i)}log(h_\theta(x^{(i)}))+\left(1-y^{(i)}\right)\log\left(1-h_\theta(x^{(i)})\right)) \tag{3.6}$$

3.4 多项式分布朴素贝叶斯算法

多项式分布朴素贝叶斯算法是自然语言处理(NLP)中流行的贝叶斯学习方法。该算法利用贝叶斯定理猜测文本的标签。它计算给定样本的每个标签的可能性，并以最大的机会输出标签。跟伯努利朴素贝叶斯的不同之处是多项式分布朴素贝叶斯将重复词语视为其重复次数，而伯努利朴素贝叶斯将重复词语视为其只出现一次。对于大规模数据，计算复杂度较低，比较适合本论文的数据。

朴素贝叶斯算法(Naïves Bayes Classifier, NBC)是垃圾邮件检测的一种经典分类算法，它建立在统计学中的贝叶斯概率理论之上，又被称为“独立特征模型”。贝叶斯规则指出，如果存在一个假说 A 和基于假说的例证 B，有如下关系：

$$P[A|B] = \frac{P[B|A]P[A]}{P[B]} \tag{3.7}$$

其中 P[A]指事件 A 发生的概率，也叫先验概率，P[A|B]是基于另一事件 B 发生，事件 A 发生的概率，也叫后验概率。URL 的特征是相互独立的，计算给定网站属于 m 类的概率(m1：非网络钓鱼，m2：网络钓鱼)，如下所示：

$$P[m1|A] = \frac{P[m1]P[A|mi]}{P[A]} \tag{3.8}$$

其中，P[A]均为常数，同时 P[A|mi]和 P[mi]通过训练可以很容易计算出。

朴素贝叶斯提供了一种简单且概念清晰的方式来表达、使用和学习知识，并且朴素贝叶斯的性能可与一些更复杂的分类器相比。这些分类器用于许多特征之间的弱依赖关系的数据集，可以达到很好的预测效果。然而，也发现朴素贝叶斯在许多其他数据集上表现不佳，因为朴素贝叶斯将属性视为完全独立，所以一些冗余属性会破坏机器学习过程。

3.5 XGBoost

XGBoost 是梯度增强树(Gradient Tree Boosting)算法的一种流行且高效的开源实现，使用 CART 树模型。梯度推进是一种监督式的学习算法，它试图通过组合一组更简单、较弱的模型的估计来准确预测目标变量[10]。Boosting 算法在处理偏差-方差权衡中起着至关重要的作用。Bagging 算法只控制模型中的高方差，与 Bagging 算法不同，Boosting 同时控制两个方面(偏差和方差)，被认为更有效。XGBoost 支持缓存感知和核心外计算、高效处理缺失数据、树构建并行、交叉验证功能、避免过度拟合的正则化、使用深度优先方法进行树修剪显著提高计算性能。

3.5.1 超参数优化

XGBoost 算法提供了大范围的超参数。超参数优化对模型性能有直接影响，为了改进和充分利用 XGBoost 模型，本文对这些超参数做调整。目前常见的超参数优化方式主要有两种：网格化寻优和随机寻优。网格化寻优技术可以说是最基本的超参数优化方法

[13]。在网格搜索中，建立了一个超参数值的网格，并为每个组合训练一个模型，然后测试数据上打分。在这种方法中，需要尝试超参数值的每一种组合，这可能非常低效。例如，为 4 个参数中的每一个搜索 20 个不同的参数值将需要 16 万次交叉验证试验。如果使用 10 倍交叉验证，这相当于 160 万次模型拟合和 160 万次预测。虽然 Scikit Learn 提供了 GridSearchCV 函数来简化过程，但无论是在计算能力还是在时间上，这都是一个极其昂贵的执行过程。

相比之下，随机搜索建立了超参数值的网格，并选择随机组合来训练模型和评分。这允许显式控制尝试的参数组合的数量。虽然 RandomizedSearchCV 可能无法找到与 GridSearchCV 一样准确的结果，但它可以更频繁地选择最佳结果，而且只花了 GridSearchCV 所需时间的一小部分。在相同的资源条件下，随机搜索甚至可以优于网格搜索。因此，本文选取使用了随机搜索。当使用连续参数时，网格布局与随机布局区别可以在下图 3.4 中见到。通过网格搜索，九次试验只测试了三个不同的地方。但通过随机搜索，所有九条路径都会探索不同的值。

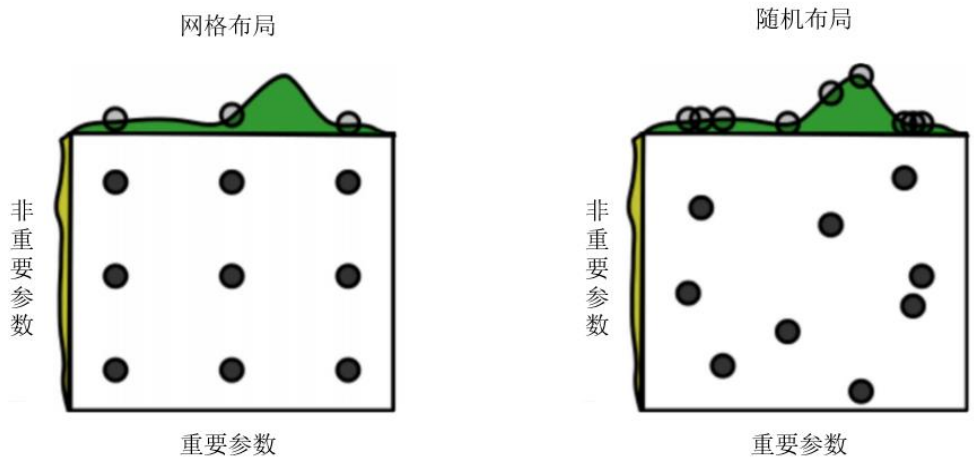

图 3.4 网格布局与随机布局

通常，XGBoost 超参数被分为 4 类，一般参数，助推器(Booster)参数, 学习任务参数, 命令行参数。最常被调整参数是助推器参数，分为两种：树助推器和线性助推器。树推助器总是优于线性助推器，因此后者很少被使用。经过随机搜索本文对这些超参数优化：

a)max_depth，是每棵树的最大深度[11]，用于控制过度的拟合，因为更深的树会允许模型学习非常特定于某种样本的关系，max_depth 值的范围(0, ∞)，默认值为 6。树越深可能会提高性能，但也会增加复杂性和过度拟合的机会。本文设置 max_depth 值为 15。

b)min_child_weight，定义了孩子节点所需的所有观察的最小权重总和，用于控制过拟合。如果树分支步骤导致叶节点的实例权重之和小于 min_child_weight，那么构建过程将放弃进一步的分支, min_child_weight 值的范围(0,∞)，默认值为 1。本文设置 min_child_weight 为 5。

c)learning_rate，学习率决定了每次迭代的步长，而该模型会朝着其目标进行优化。低学习率会使计算变慢，并且需要更多轮次才能实现与具有高学习率的模型相同的残差减少，但优化了达到最佳状态的机会。本文设置 learning_rate 值为 0.15。

d)gamma, 是一个伪正则化参数(拉格朗日乘子)，并且依赖于其他参数。只有当结果分割使损失函数正减少时，才会分割节点。Gamma 值越高，正则化程度越高，默认值为 0。本文设置 gamma 值为 0.3。

e)col_sample_bytree, 表示要为每棵树随机抽样的列的分数，它可能会改善过度拟合。对每棵构建的树进行一次抽样。本文设置 col_sample_bytree 值为 0.4，默认值为 1。

3.5.2 交叉验证

在模型训练过程中，当调整超参数时，需要一种评估模型性能的方法。然而，为此目的使用测试集将是一个错误方法，因为测试集应该代表新，之前没看过的数据，而这些数据只应作为模型能力的最终评估指标。因此本文使用交叉验证来评估超参数的模型性能，交叉验证是一种评估泛化的数据重采样预测模型的能力和避免过拟合方法。k-折叠交叉验证包括几个步骤。首先将训练集划分为大小相等的 k 个不相交子集。分区阶段使用来自训练集的随机采样执行。分层随机抽样通常用于确保单个类的类表示是正确的大致与整个训练环境成比例。在获得 k 个“折叠”后，对模型进行 k 次训练，每次都保留一个不同的子集作为验证集，在其上测量性能。所有迭代的平均性能是整体交叉验证性能。在这实验，本文使用 5 折交叉验证，如图 3.5 所示。

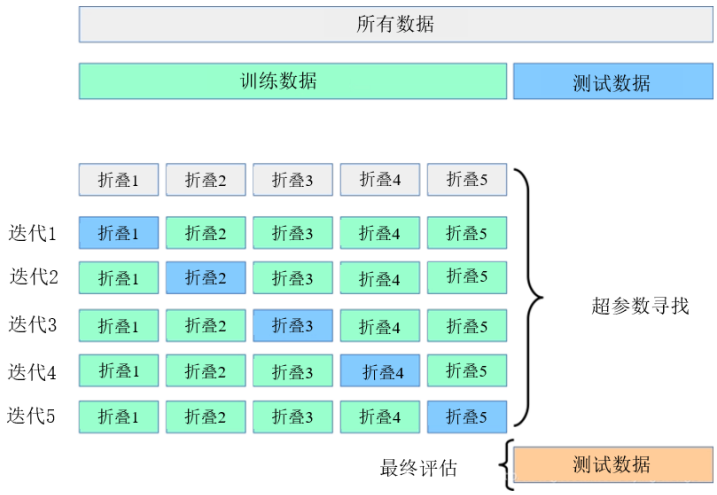

图 3.5 交叉验证

3.6 本章小结

本章介绍了机器学习的典型框架和分类，然后对典型监督式模型的机器学习算法介绍。之后，本文使用随机搜索对 XGBoost 超参数进行优化，采用 5 折交叉验证来评估模型性能。下一章将对实验结果进行分析。

4 实验结果与分析

4.1 混淆矩阵

为了分析监督学习算法的效率和准确率，本文使用混淆矩阵。混淆矩阵是一个列联表，可以可视化监督学习算法的效率和性能，这使得系统很容易理解是否混淆或错标记了两个类。二元分类问题的混淆矩阵是一个二乘二矩阵，用于评估分类算法的性能。对于具有 N 个类的多类问题，它也可以扩展到 N × N 矩阵。

在图 4.1 中，混淆矩阵可以分为四个部分，真阳性(True Positive)表示钓鱼网页被正确识别为钓鱼网页。真阴性(True Negative)表示安全网页正确标记为安全。假阳性(False Positive)意味着安全网页被错误地识别为钓鱼网页。假阴性(False Negative)表示钓鱼网页被错误地标记为安全。Scikit-learn 有一个混淆矩阵函数，本文使用它来获取混淆矩阵中的四个值。假设 y 是真实目标值，y_pred 是预测值，可以使用混淆矩阵作为 confusion_matrix(y,y_pred)。机器学习模型是使用开源 python 机器学习库 Scikit learn 进行训练的，需要注意的一件事是 Scikit-learn 反转混淆矩阵布局以首先显示假例。图 4.1 左是普通混淆矩阵，图 4.1 右是 Scikit-learn 的混淆矩阵布局。

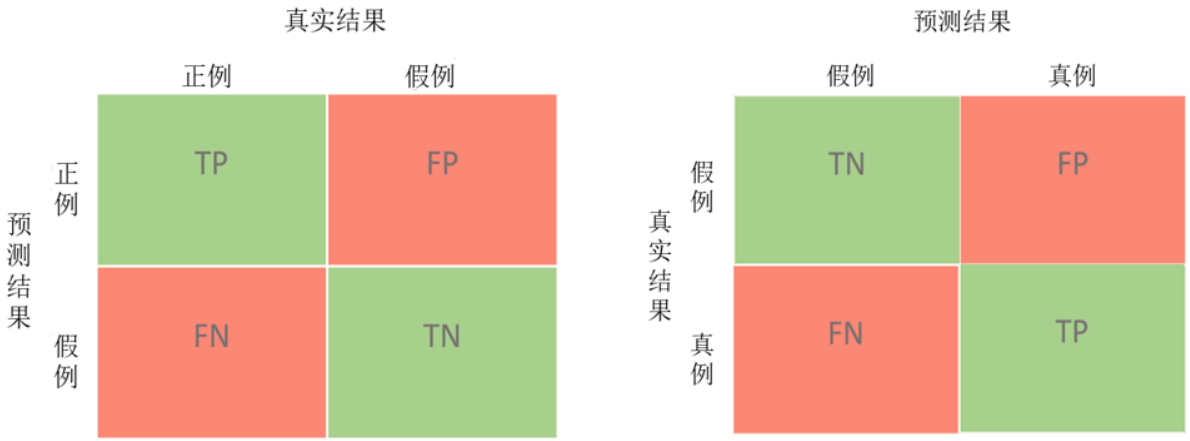

图 4.1 普通混淆矩阵(左)与 Scikit-learn 混淆矩阵布局(右)

表 4.1 分类错误测量

	公式	意义
准确率 ACC	$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$	分类模型所有判断正确的结果占总观测值的比重
精确率 PPV	$Precision = \frac{TP}{TP + FP}$	在模型预测是 Positive 的所有结果中，模型预测对的比重
灵敏度 TPR	$Sensitivity = Recall = \frac{TP}{TP + FN}$	在真实值是 Positive 的所有结果中，模型预测对的比重
特异度 TNR	$Specificity = \frac{TN}{TN + FP}$	在模型预测是 Negative 的所有结果中，模型预测对的比重
F-Score	$F - Score = \frac{2 * Recall * Precision}{Recall + Precision}$	通过取调和均值，将分类器的精确率和召回率组合成单个度量

其中, TPR(True Positive Rate)，真阳性，即称为灵敏度(Sensitivity)或者召回率(Recall)也就是在实际为正例的结果中，模型找出了多少。TNR(True Negative Rate)，真阴性，即称为特异度(Specificity), 也就是在实际为假例的结果中，模型找出了多少。灵敏度和精确率最高是最理想的情况。精确率是模型预测为正例的结果中，有多少确实是正例。准确率(Accuracy)就是模型预测正确的结果占总结果的比重。F-Score 是主要用于比较两个分类器的性能。假设分类器 A 具有更高的召回率，而分类器 B 具有更高的精确率。在这种情况下，两个分类器的 F-Score 可用于确定哪一个产生更好的结果。

4.1.1 权衡偏差与方差

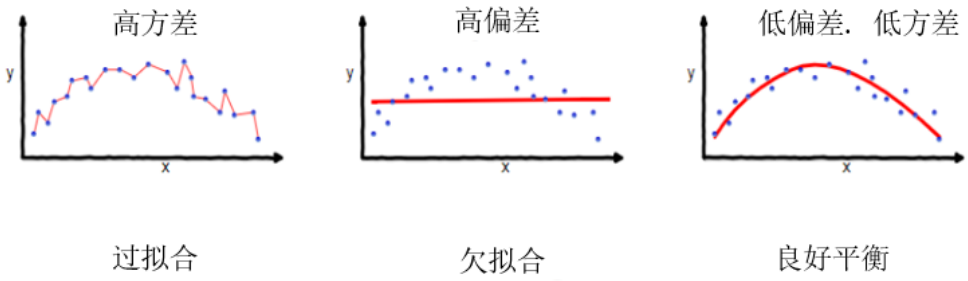

图 4.2 过拟合与欠拟合

每当模型预测时，理解预测误差，即偏差和方差，是很重要的。在模型最小化偏差和方差的能力之间存在权衡。偏差是模型的平均预测值与试图预测的正确值之间的差异[12]。具有高偏差的模型很少关注训练数据，并且过度简化了模型。它总是导致训练和测试数据的高误差，也成为欠拟合。方差是指模型不仅获取了数据集的信息还提取了噪声数据的信息使得此类模型在训练数据上表现良好，但在测试数据上具有较高的误差率。

因此，一个最佳的机器学习模型应在训练集和测试集都有较好的表现，良好平衡，如图 4.2 所示。

4.2 受试者工作特征曲线以及曲线下面积(AUC-ROC)

除了使用混淆矩阵，本文使用 AUC-ROC 曲线来测量模型的性能，也称为 AUROC (Area Under Receiver Operating Characteristic)作为评价指示。ROC，即受试者工作特征，总结了分类模型在所有分类阈值下的预测性能。ROC 用于表示真阳性比例(TP)与假阳性比例(FP)的关系图，分别 Y 轴代表真阳性率(TPR)，X 轴代表假阳性率(FPR)。点(0,1)是完美的分类器，因为代表正确地分类了所有正例和负例。因此，一个理想的系统将通过识别所有正例来启动，曲线将立即上升到 (0,1)，误报率为零，然后继续沿着(1,1)前进。对网络钓鱼数据集的检测率和误报进行评估，并将获得的结果用于形成 ROC 曲线。左上角的一个数据点对应于最优的高性能，即高检测率和低误报率。测试的准确性取决于测试将被测试组分类为 0 或 1 的程度。蓝色对角线表示随机预测来比较分类器性能能否至少超越随机预测。准确性通过 ROC 曲线下的面积(AUC)来衡量。AUC 体现着分类器的泛化能力，面积 1 代表完美的测试，面积 0.5 代表毫无价值的测试。在本文实验中，最高的 AUC 是逻辑回归模型，获得了 0.996，如图 4.3 所示。

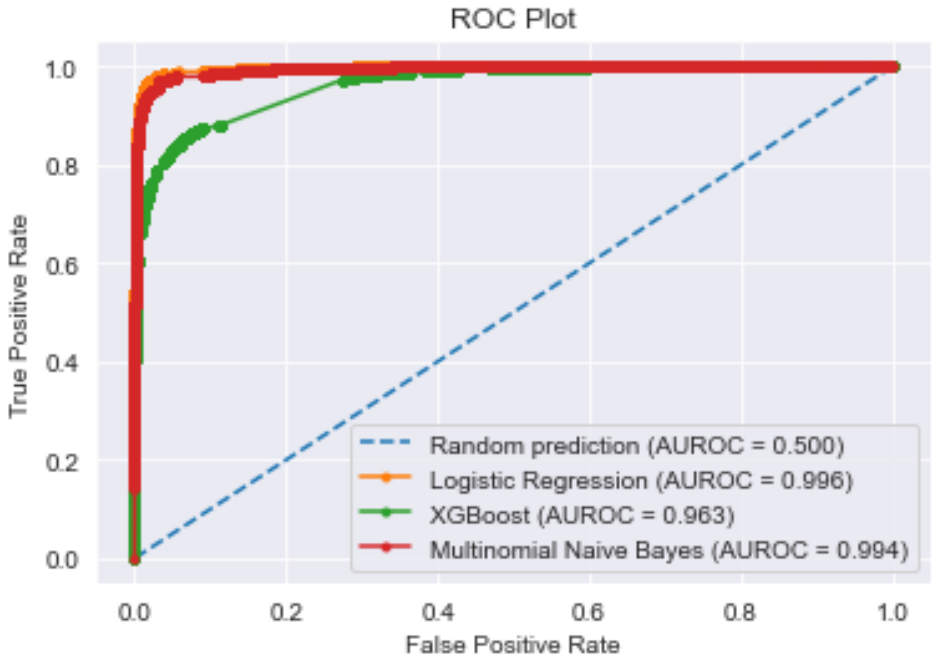

图 4.3 ROC 曲线的比较

4.3 测试环境

测试环境由硬件和软件组成。使用的硬件如下表所示：

表 4.2 硬件规格

名字	描述
笔记本电脑	HP Intel Core i7，硬盘 500GB，内存 8GB
鼠标	Logitech M330

除硬件外，本研究还使用了一些软件进行测试，如下表所示：

表 4.3 软件规格

名字	描述
操作系统	Windows 10
编程语言	Python 3.8.8
浏览器	Microsoft Edge
编程环境	Jupyter notebook

4.4 逻辑回归测试结果

本文对逻辑回归算法进行了试验，以使用经过数据预处理阶段的测试集来衡量逻辑回归算法的性能，并证明逻辑回归算法生成的预测是相当不错的。因为基本上逻辑回归算法可用于使用概率和统计方法根据现有数据进行预测。下面是测试逻辑回归算法的结果：

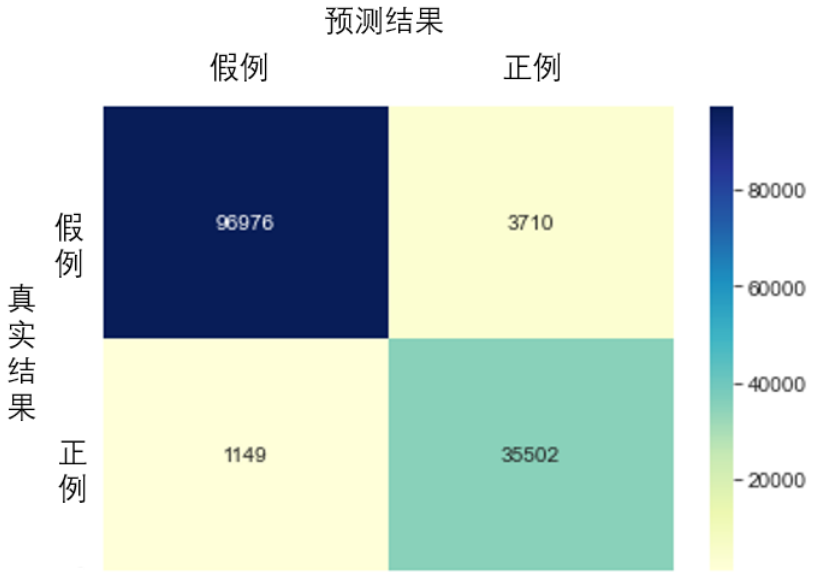

图 4.4 逻辑回归混淆矩阵结果

图 4.4 显示，逻辑回归算法将 3710 个非钓鱼网站预测为钓鱼网站(FP)，将 1149 个钓鱼网站预测为非钓鱼网站(FN)，则有 96,976 个针对非网络钓鱼网站的正确预测(TN)

和 35,502 个针对网络钓鱼网站的正确预测(TP)。此外，逻辑回归算法产生的准确率为96.41%，训练集和测试集准确性为 0.97 和 0.96。同时，基于 P(Precision)，R(Recall)，F(F-Score)以及准确率的逻辑回归算法性能如表 4.4 所示。

表 4.4 逻辑回归分类报告

类别	精确率(P)	召回率(R)	F-score
钓鱼	0.99	0.96	0.98
合法	0.91	0.97	0.94
准确率		0.96	

4.5 XGBoost 测试结果

下面是测试 XGBoost 算法的结果，图 4.5 显示 XGBoost 算法将 9655 个非钓鱼网站预测为钓鱼网站(FP)，将 2032 个钓鱼网站预测为非钓鱼网站(FN)，则有 96,093 个针对非网络钓鱼网站的正确预测(TN)和 29,557 个针对网络钓鱼网站的正确预测(TP)。此外，XGBoost 算法产生的准确率 91.41%，训练集和测试集准确性为 0.96 和 0.95。同时，基于 P(Precision)，R(Recall)，F(F-Score)以及准确率的 XGBoost 算法性能如表 4.5 所示。

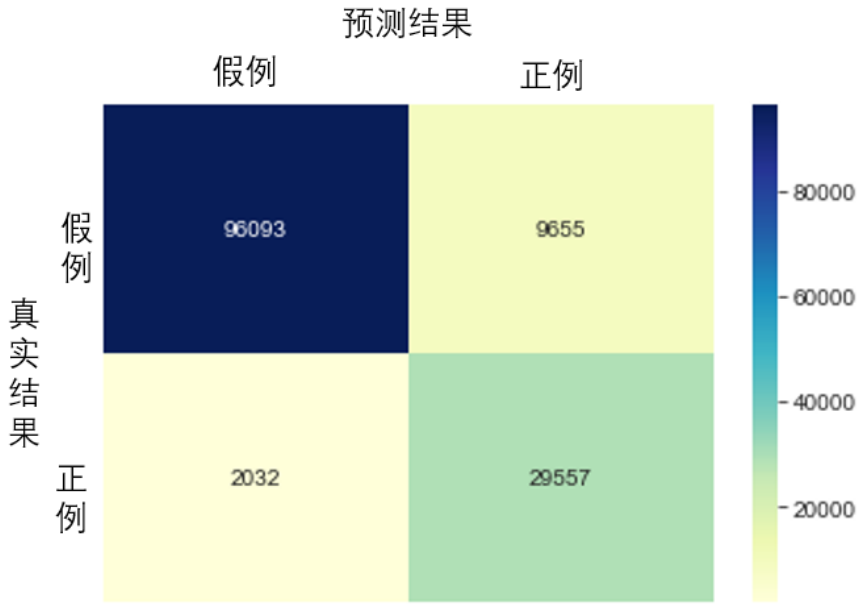

图 4.5 XGBoost 混淆矩阵

表 4.5 XGBoost 分类报告

类别	精确率(P)	召回率(R)	F-score
钓鱼	0.98	0.91	0.94
合法	0.75	0.94	0.83
准确率		0.91	

4.6 多项式分布朴素贝叶斯测试结果

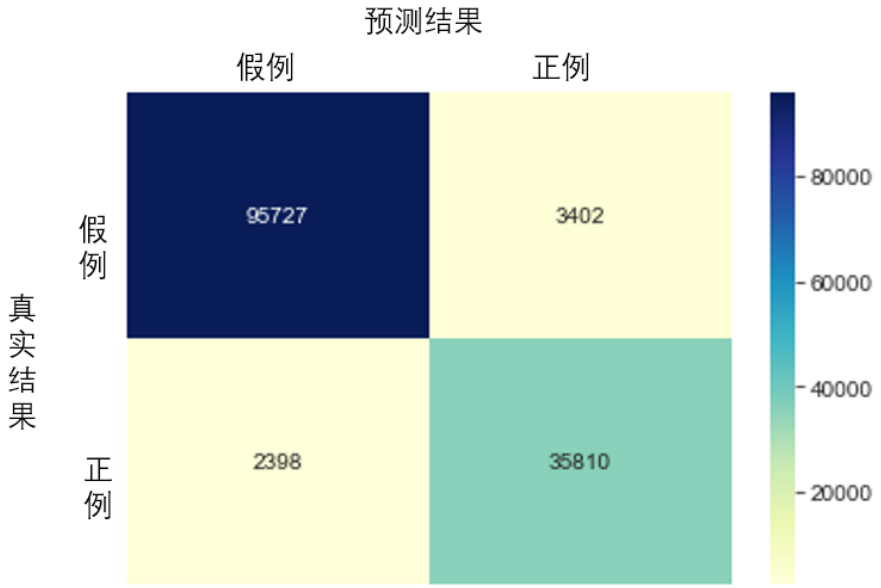

图 4.6 多项式分布朴素贝叶斯混淆矩阵

图 4.6 显示多项式分布朴素贝叶斯算法将 3402 个非钓鱼网站预测为钓鱼网站(FP)，将 2398 个钓鱼网站预测为非钓鱼网站(FN)，则有 95,727 个针对非网络钓鱼网站的正确预测(TN)和 35,810 个针对网络钓鱼网站的正确预测(TP)。此外，多项式分布朴素贝叶斯算法产生的准确率为 96%，训练集和测试集准确性为 0.97 和 0.95。同时，基于 P(Precision)，R(Recall)，F(F-Score)以及准确率的多项式分布朴素贝叶斯算法性能如表 4.6 所示。

表 4.6 多项式分布贝叶斯分类报告

类别	精确率(P)	召回率(R)	F-score
钓鱼	0.98	0.97	0.97
合法	0.91	0.94	0.93
准确率	0.96		

4.7 实验结果总结

本文所提出的机器学习技术的设计为优先检测真阳性的钓鱼网站。考虑到这一点，从高到低的优秀真阳性性顺序如下：逻辑回归(96.8%)，多项式分布朴素贝叶斯(96.7%)和 XGBoost(91%)。因此，本文在下一章会使用逻辑回归模型部署。从上面的实验结果数据结合得表 4.7。

表 4.7 实验结果总结

分类器	准确率	精确率	召回率	F-分数	ROC-AUC
逻辑回归	0.96	0.99	0.96	0.98	0.996
XGBoost	0.91	0.98	0.91	0.94	0.963
多项式分布朴素贝叶斯	0.96	0.98	0.97	0.97	0.994

5 系统部署实现

5.1 实际系统实现

本文使用上一章已经过机器学习训练过的模型保存到 Pickle 文件，Pickle 是一种 Python 中序列化对象的标准方法，然后部署到网站[14]。这网站希望可以帮助一些互联网用户决定一个网站是合法的还是网络钓鱼。在该网站中，本文提供了一些信息关于如何避免网络攻击的方法，以便用户能够更清楚地了解并远离钓鱼者。本文使用 Streamlit 来部署模型，Streamlit 是一个面向机器学习的开源应用程序框架。Streamlit 中的代码很容易部署、管理和与其他应用程序协作。

该网站将要求用户输入一个 URL 进行进一步检查。输入 url 后，用户需要按下“预测网站结果”按钮。如果用户输入的 URL 被检测为钓鱼网站，它将返回“This is a Phishing Website”的红色框，如图 5.2 所示。如果网站是安全的，那么它将返回“This is not a Phishing Website”的绿色框，如图 5.3 所示。本文把模型部署到本地 localhost，使用 Anaconda Prompt 的命令来执行，网站页面如图 5.1 所示。为了能让更多人访问，本文使用 Streamlit 提供的平台 Streamlit share 部署到远程，只要有链接的人都可以随时访问。

图 5.1 钓鱼网站检测系统

5.2 模型部署环境

模型部署环境由硬件和软件组成。本文使用的硬件如下表所示：

表 5.1 硬件规格

名字	描述
笔记本电脑	HP Intel Core i7，硬盘 500GB，内存 8GB
鼠标	Logitech M330

本文使用的软件如下表所示：

表 5.2 软件规格

名字	描述
操作系统	Windows 10
编程语言	Python 3.8.8
浏览器	Microsoft Edge
编程环境	Spyder(Anaconda)
模型部署	Streamlit Share

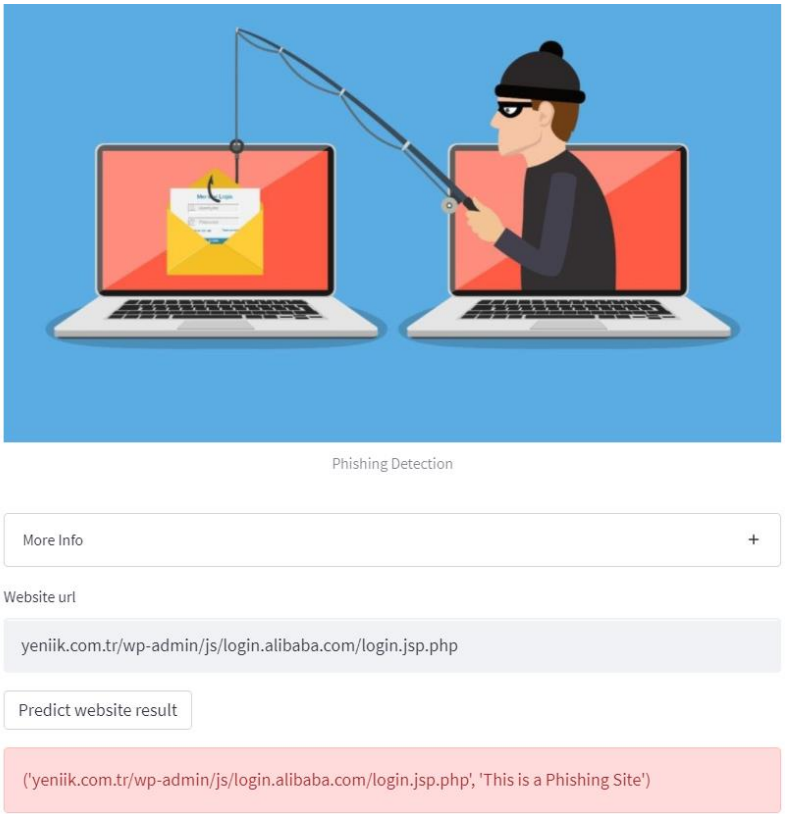

图 5.2 检测钓鱼网站

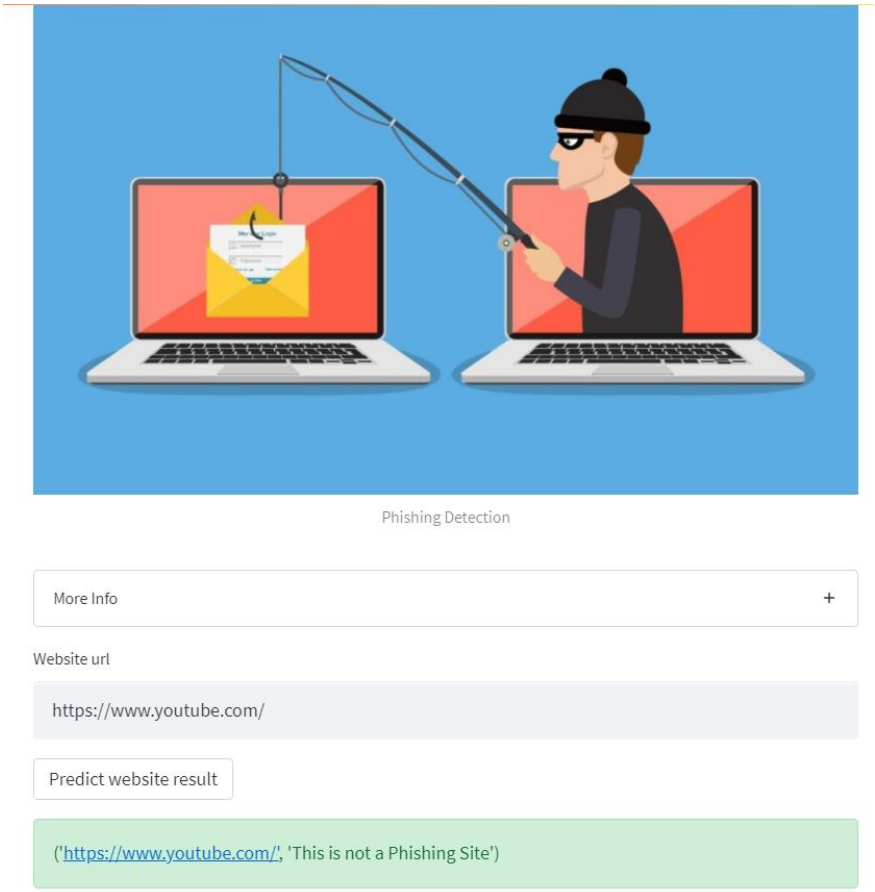

图 5.3 检测合法网站

5.3 本章小结

本章提出了基于机器学习的钓鱼网站检测系统。本章将利用上一次保存的模型部署到 Streamlit 平台实现了钓鱼检测系统，显示系统的用户页面。

总结与展望

论文总结

本文实现了利用机器学习算法进行钓鱼检测，介绍各种钓鱼的形式以及它们的危害。本文回顾了一些传统的网络钓鱼检测方法以及它们的优缺点。虽然网络钓鱼问题无法根除，但可以通过两种方式来减少，一种是改进网络钓鱼检测，另一种是告知公众如何检测和识别欺诈性网络钓鱼网站。为了对抗不断演变和复杂的网络钓鱼攻击，机器学习技术至关重要。本文使用逻辑回归算法，多项式分布朴素贝叶斯算法以及 XGBoost 三个机器学习算法进行模型训练，得到最好的准确率是逻辑回归算法，准确率达到 96.8%，并且 ROC 面积是 0.996。最后，本文使用 Streamlit 平台把最好模型，也就是逻辑回归模型部署到远程，实现了钓鱼网站检测系统。

展望

本文的工作可以通过创建浏览器插件来进一步扩展。另外，也可以结合在线学习，以便可以轻松学习新的网络钓鱼攻击模式，并通过更好的特征提高模型的准确率。

致谢

首先，我要感谢我的导师傅翠娇老师在整个研究过程中提供的指导。我还要感谢北航可以在疫情期间提供了线上学习资源，可以完成顺利毕业。

最重要的是，我要感谢自己从未放弃，相信自己即使面对巨大的挑战，困难和压力也能不懈地工作。

最后，在我生命中最重要的人，我的父母，感谢他们无私的爱是我用再多语言也感激不完的，没有他们，这一成就是无法实现的。

参考文献

[1] Shaikh, et al. A Literature Review on Phishing Crime, Prevention Review and Investigation of Gaps[A]. In 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)[C]. 2016:9-15.

[2] A. Kang, et al. Security Considerations for Smart Phone Smishing Attacks[J]. Advances in Computer Science and its Applications,Springer, 2014:467–473.

[3] Greg Aaron. APWG Phishing Activity Trends Report[R]. APWG, 2021.

[4] Sharifi, et al. A Phishing Sites Blacklist Generator[A]. IEEE/ACS International Conference on Computer Systems and Applications[C]. 2008:840-843.

[5] Cao Ye, et al. Anti-phishing based on automated individual white-list[A]. In Proceedings of the 4th ACM workshop on Digital identity management[C]. China: ACM workshop on Digital identity management, 2008: 51-60.

[6] Zhang, et al. Cantina: A Content-Based Approach to Detecting Phishing Web Sites[A]. In Proceedings of the 16th International Conference on World Wide Web[C]. New York:2007: 639-648.

[7] Anthony Y, et al. Detecting phishing web pages with visual similarity assessment based on earth mover's distance[J]. IEEE Transactions on Dependable and Secure Computing, 2006: V3(4) 301–311.

[8] Tom M.Mitchell. Machine Learning[Z]. McGraw-Hill,1999.

[9] Kleinbaum, David G, et al. Logistic Regression[Z]. New York: Springer-Verlag, 2002.

[10] Chen, Tianqi, et al. Xgboost: A scalable tree boosting system[A]. In Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining[C]. 2016:785-794.

[11] Wang, Yan, et al. A XGBoost risk model via feature selection and Bayesian hyper-parameter optimization[J]. arXiv preprint: 2019.

[12] Mehta, et al. A high-bias, low-variance introduction to machine learning for physicists[R].USA: Physics reports 810, 2019.

[13] Bergstra, James, and Yoshua Bengio. Random search for hyper-parameter optimization[J]. Journal of machine learning research 13(2): 2012.

[14] Singh P. Deploy Machine Learning Models to Production[Z]. India: Apress Publishing House, 2021.