[redacted]

本科生毕业设计[论文]

基于机器学习的恶意流量检测系统研究与实现

院　　系 [redacted]

专业班级 [redacted]

姓　　名 [redacted]

学　　号 [redacted]

指导教师 [redacted]

2022 年 5 月 22 日

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名：[redacted]

学位论文版权使用授权书

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于 1、保密口，在　　年解密后适用本授权书

2、不保密口　。

（请在以上相应方框内打“√”）

作者签名：[redacted]

导师签名：

摘　要

网络信息技术的快速发展和相关服务的广泛应用，在推动着社会经济的快速发展的同时，也暴露许多网络安全方面暗藏的漏洞，恶意流量检测成为了网络安全中的一个重要问题。因此，一种基于机器学习的恶意流量检测系统被提出，该系统利用机器学习算法对网络流量进行分类，可以有效地识别恶意流量。

系统包括三大功能：不同算法不同种类流量的训练功能、本地测试集测试功能、用户输入数据测试功能。系统选择了 KNN 算法和逻辑回归算法作为系统的机器学习核心算法，因为不同种类流量的特征向量维度很难保持一致，所以根据五种不同的数据类型和两种不同的算法得到了共十种二分类训练模型。系统首先在 github 上收集到了相关的恶意流量 payloads，其中恶意流量包括 SQL 注入攻击、XSS 攻击、目录遍历攻击和 CRLF 注入攻击；在 HTTP DATASET CSIC 2010 数据集中选取部分作为系统正常流量数据集，并从中的 http 请求中提取出其 URL 作为数据对象。然后根据不同种类流量的特点分别进行特征提取工作，根据得到的特征向量分别利用 KNN 算法和逻辑回归算法进行不同模块的训练得到相应的模块并保存。系统可以根据用户不同的选择接受不同的测试集来进行测试，最终根据测试结构生成一份测试报告，其中包括了本次测试的准确率、回归率等信息。除此之外，系统还设计了与用户的交互功能，当用户输入单个 URL 让系统进行判断时，系统会将测试结果输出给用户并让用户进行是否准确的反馈，若用户判断本次测试准确系统会将此次测试的 URL 加入的训练集中并重新训练。

系统最终测试结果显示，无论是训练数据，本地测试数据集还是用户测试数据集，KNN 算法的精确率、召回率和 F1 值总体上都比逻辑回归算法高，并且两个算法的精确率、召回率和 F1 值整体都在 90%以上，表明了系统良好的实现效果。

关键词：机器学习；恶意流量；数据预处理；特征提取；分类算法

Abstract

The rapid development of network information technology and the wide application of related services, while promoting the rapid development of social economy, also exposed many hidden loopholes in network security, malicious traffic detection has become an important issue in network security. Therefore, a machine learning-based malicious traffic detection system is proposed, which uses machine learning algorithms to classify network traffic and can effectively identify malicious traffic.

The system includes three functions: the training function of different algorithms and different types of traffic, the local test set test function, and the user input data test function. The system chooses the KNN algorithm and the logistic regression algorithm as the core algorithm of the system's machine learning, because it is difficult to keep the feature vector dimensions of different types of traffic consistent, so according to five different data types and two different algorithms, a total of ten kinds of two kinds of data are obtained. Classification training model. The system first collected related malicious traffic payloads on github, among which malicious traffic included SQL injection attacks, XSS attacks, directory traversal attacks and CRLF injection attacks; selected part of the HTTP DATASET CSIC 2010 data set as the normal traffic data set of the system, and from it Extract its URL from the http request as a data object. Then, feature extraction is carried out according to the characteristics of different types of traffic, and the KNN algorithm and logistic regression algorithm are used to train different modules according to the obtained feature vectors to obtain corresponding modules and save them. The system can accept different test sets for testing according to different choices of users, and finally generate a test report according to the test structure, which includes information such as the accuracy rate and regression rate of this test. In addition, the system has also designed an interactive function with the user. When the user enters a single URL for the system to judge, the system will output the

test result to the user and let the user give feedback on whether it is accurate. If the user judges that the test is accurate The system will add the URL of this test to the training set and retrain.

The final test results of the system show that whether it is training data, local test data sets or user test data sets, the precision rate, recall rate and F1 value of the KNN algorithm are generally higher than those of the logistic regression algorithm, and the precision rate and recall rate of the two algorithms The overall rate and F1 value are above 90%, indicating that the system has a good implementation effect.

Key Words： Machine Learning; Malicious traffic; Data preprocessing; Feature extraction; Classification algorithm

目 录

1 绪论

1.1 选题背景

根据中国互联网络信息中心(China Internet Network Information Center，CNNIC)发布的报告[1]中展示，截止 2022 年 12 月，中国网民规模达 10.67 亿。随着计算机网络技术的高速发展，互联网已经覆盖了社会生活的各个方面。如今，网络技术在经济、军事、教育等各社会领域都有着非常广泛而深入的应用[2]。

人们在社会中的需求也趋于多元化，而发展迅猛的网络信息技术及时地弥补了这一方面的缺憾，尤其是移动设备、物联网设备等种种遍历更是促使了网络资源在人类生活中的所占比重日渐增长[3]。我们可以看到不同的群体对网络服务的涉入领域也有所区分。如有社交需求的人群经常使用微信、qq 等社交软件；同时有着购物需求的人群则通过淘宝、京东、拼多多等电商软件迅速的达到自己的目的；而更多在日常工作中办公、学习的人民需要多种方面的软件等帮助，从而更便利地进行交流、研究以及其他商业活动。

网络信息技术的快速发展和相关服务的广泛应用，在推动着社会经济的快速发展的同时，也暴露许多网络安全方面暗藏的漏洞[4]。得益于 TCP/IP 协议[5]的广泛应用，一台计算机通过互联网几乎可以将信息传递至世界的每个角落，但另一面，计算机病毒也凭借于此扩散和进化，愈加难以被固有方法检测拦截[6]。先进计算机网络技术的研发本是为了人们生活幸福指数的提高，却被大量不法分子认作牟利或破坏的工具，这种行为不仅会影响网络环境的平稳运行，更渐渐地在现实生活中造成人们的经济损失，对社会整体产生不良影响[7]。

近几年中，网络的应用迅速深入了我们生活中的方方面面，大量具有重要价值的信息在网络空间中以网络流量为载体进行交互和传输，因此网络空间中的信息安全要求也随之提高。但相比于发达国家已形成的较为完整的安全战略体系，我国的网络安全产业依旧处于发展阶段。人们习惯于将一些个人信息和重要文件保存在计算机软盘或硬盘中。黑客利用熟练的技术和丰富的知识，结合计算机中的漏洞和病毒程序[8]，对用户的重要文件进行恶意攻击甚至恶意篡改。携带病毒的文件会继续恶意快速地感染用户计算机中的其他文件[9]，导致数据丢失和系统崩溃。木马病毒程序通过非法入侵和窃取用户重要信息和账号密码信息[10]，使个人

信息安全受到严重威胁。为防止计算机被不法分子和病毒程序入侵，少数用户会在计算机中安装安全查杀软件，但大部分用户网络安全意识薄弱，未能及时查杀，人为地给计算机带来了严重的负担。此外，因缺乏对网络欺诈的恶意信息防范意识，导致人为地泄露自己的身份信息，这些问题使得网络安全无法得到保障。

众多安全隐患中，信息泄漏由于与日常生活息息相关，一直以来都是人们的关注热点。过去已发生的信息泄漏事件所处行业涉及各个领域，如全球化公司、大型工商企业这样的大规模企业，也可能是小规模的城市医保系统，如已公开使用的 12306 网站、医疗行业患者信息、邮寄服务、出行共聚、社交工具等。由此可以看出在网络安全的技术发展方面，我国仍然走在风险与挑战并行的路上。

1.2 选题目的及意义

网络安全问题一直是人们关注的焦点，而恶意流量攻击作为一种常见的网络安全威胁，已经给网络系统的安全性带来了很大的挑战。传统的恶意流量检测方法往往需要事先定义特征和规则，但是这些方法对于未知的恶意流量难以进行准确的检测，同时对于动态变化的网络环境也存在一定的局限性。机器学习技术以其对数据的学习和识别能力，已经成为了恶意流量检测领域中的重要技术手段，能够通过对恶意流量行为模式的学习和分析，实现对恶意流量的高效、准确的检测和防范。本文将通过对机器学习的相关理论及恶意流量检测技术的分析和研究，设计和实现一个基于机器学习的恶意流量检测系统，从而提高网络系统的安全性和稳定性，为网络安全领域的发展做出一定的贡献。同时希望通过本次系统的设计和论文的书写达到以下目的：

（1）了解恶意网络攻击流量，如 SQL 注入攻击，XSS 攻击，目录遍历攻击，CRLF 注入攻击等攻击流量；

（2）从负例样本中筛选出最具代表的特征，如关键词，短语等，使用机器学习分类算法识别恶意流量；

（3）编程实现设计的分类算法及检测系统，使用样本验证改进算法；

（4）熟练运用编程开发技能，独立完成本课题的设计、开发和测试工作；

（5）撰写符合规范要求的论文；

1.3 国内外研究现状

恶意流量检测是网络安全领域中一个重要的研究方向。近年来，随着机器学习技术的不断发展，基于机器学习的恶意流量检测系统逐渐成为主流。上个世纪80年代，Anderson[11]首次提出了网络安全和入侵检测的概念入侵检测可以分为外部入侵、内部入侵和不法行为[12]。随后Denning[13]等人提出了入侵检测系统（IDS）的概念，该模型的提出，为后来的相关研究打下了坚实的基础。从1990年开始，网络攻击变得越来越多，因此第一套入侵检测系统得以产生，第一套对于网络的安全监控系统(Network Security Monitor，NSM)由 Heberlein[14]所研发。之后不久，第一个网络入侵检测系统(Network Instruction Detection System，NIDS)正式产生，它的模型非常完整，对入侵发生前、发生时以及发生后都有比较妥当的处理办法。至此，两种不同恶意流量检测体系结构诞生了分别是：基于主机和基于网络。在1998年发生的莫里斯蠕虫事件对入侵检测的研究起到了进一步的推动作用[15]。

在机器学习中，特征提取扮演了十分重要的作用。Smitha[16]使用最小冗余最大相关性(mRMR)的方法，筛选出与分类变量具有最高相关性的的特征，采用了支持向量机（SVM）和逻辑回归(LR）算法检测异常。Sahin[17]使用了基于请求的长度和类似于误用检测模式匹配方法来提取特征，使用了C4.5决策树模型，达到了较高的准确率。Zhang、Ju和Kamel[18]对服务器日志进行会话分割，并采使用了贝叶斯分类方法来检测日志中的异常数据，但缺少对 Web日志中请求参数特征的提取。Cho[19]则综合了无监督学习方法与合法规则，从浏览器访问服务器的流程与结构等特点生成了一个基于Web站点架构的规则库，依据该规则库进行异常检测。

近几年，针对网络中异常访问流量检测的方法大多侧重下面所述三个方向：基于人工规则的流量检测方法、基于机器学习的流量检测方法以及基于深度学习的流量检测方法[20]。

迄今为止，国内外的很多学者已经提出了很多不同类型的网络流量异常检测方法。根据Ahmed等的研究成果，网络流量异常检测方法可分为基于分类、基于统计[21]、基于聚类、基于信息论等四类方法。其中，基于网络流量分类的方法是其中很重要的一类。网络流量分类，是指将网络流量归类至特定的应用类型，是网络流量分析领域的项基本任务。例如，可以将流量归类至某一类应用，例如聊天类、视频流类、邮件类等，也可以根据具体业务需求分类，例如将流量划分为

加密流量和非加密流量[22]。网络流量分类在网络管理和网络安全领域都起着基础作用例如，在网络管理领域，可以将流量分类为不同的优先级，以实现更好的服务质量控制；在网络安全领域，可以将流量分为正常流量和恶意流量，以实现网络异常检测并采取防护措施。

1.4 本文结构安排

文章一共分为六章。其中第一章主要对选题背景和国内外研究现状进行一定的介绍。第二章将介绍网络安全基础知识、恶意流量的概述、恶意流量的分类以及机器学习的相关知识，为后续恶意流量检测系统的研究和实现提供必要的基础知识。第三章围绕数据处理和特征提取展开讨论，并探究其中的相关技术和方法。首先将介绍数据预处理的基本概念和技术。接着，文章将重点探讨特征提取的相关技术。最后，文章将结合实际案例对数据处理和特征提取的重要性进行探究。第四章通过设计与实现一个基于机器学习的恶意流量检测系统来探讨机器学习算法在恶意流量检测中的应用。第五章将对实验所用的环境和软件进行一个简单的说明，并对测试的所有结果进行一个分析和评估，并分析产生这个结果的原因。第六章将对整个文章进行总结。

2 相关技术背景

随着互联网技术的快速发展和普及，网络已经成为人们生活和工作中不可或缺的一部分。然而，网络安全问题也随之而来，包括恶意攻击、网络病毒、僵尸网络等，严重危及网络系统和用户的安全。恶意流量作为网络安全领域的一个重要组成部分，对于网络攻击的检测和防范至关重要。本章将介绍网络安全基础知识、恶意流量的概述、恶意流量的分类以及机器学习的相关知识，为后续恶意流量检测系统的研究和实现提供必要的基础知识。

2.1 网络安全基础知识

网络安全是指保护网络系统、数据和应用程序免受各种威胁和攻击[23]，包括未授权访问、数据泄露、病毒感染、DoS 攻击等。随着互联网的快速发展和普及，网络安全问题已经成为公共安全的一部分，需要得到高度关注。在这里，我们将对网络安全基础知识进行更深入的探讨[24]。

网络安全的基本要素包括机密性、完整性和可用性。机密性指的是保护数据不被未授权访问或泄露，确保数据只被授权的用户访问[25]。完整性指的是保护数据不被恶意篡改或破坏，确保数据的准确性和完整性。可用性指的是确保网络系统和应用程序在需要时可用和可靠，避免因网络攻击和其他原因导致服务不可用的情况发生[26]。

要保障网络安全，首先需要建立一个完善的网络安全体系。这个体系包括对网络系统进行合理的设计和配置、对数据进行加密保护、对恶意攻击进行实时检测和防范等措施[27]。其中，对网络系统进行合理的设计和配置是最基础的一步。网络系统的设计应该根据安全要求和实际需求进行合理的设计和配置，例如设置合适的网络拓扑结构、部署防火墙和入侵检测系统等，以提高网络的安全性和稳定性。此外，对数据进行加密保护也是网络安全体系的重要组成部分[28]。加密技术可以有效地保护数据的机密性和完整性，例如采用 SSL/TLS 协议加密网站通信、采用文件加密技术加密重要数据等[29]。

恶意攻击是网络安全领域的一个重要威胁。为了对恶意攻击进行有效的检测和防范，需要采用一些实时检测和防范措施。入侵检测系统[30]、入侵防御系统、网络安全监控系统等是常用的防范和检测手段。其中，入侵检测系统可以对网络

流量进行实时监控和分析，及时发现网络攻击并进行响应和处理；入侵防御系统可以根据安全策略对网络流量进行过滤和防御，有效地防止网络攻击的发生；网络安全监控系统可以对网络设备、主机和应用程序进行监控和分析，及时发现和处理安全事件。

为了保障网络安全，需要采取一系列措施，包括但不限于强化网络管理和监控、加强系统安全配置和管理、使用高强度加密技术、建立有效的访问控制和身份认证机制、加强安全审计和日志记录、进行定期的安全检测和漏洞修复等。此外，对于企业和组织而言，建立健全的安全管理体系和组织机构也非常重要。

2.2 恶意流量概述

恶意流量是指在网络中传输的具有恶意目的的数据包或流量[31]。恶意流量通常包括病毒、木马、蠕虫、僵尸网络等恶意代码，用于攻击和破坏网络系统和数据安全。恶意流量的出现给网络安全带来了巨大的挑战，因此研究和发展恶意流量检测系统成为了当前网络安全领域的热点之一。

恶意流量可以通过多种方式传播，包括但不限于邮件、Web、文件共享等。其中，Web 恶意流量是最为常见的一种方式。攻击者通常会在受害者不知情的情况下，将恶意代码隐藏在看似合法的 Web 页面中，一旦受害者访问该页面，恶意代码就会被激活并对受害者的计算机造成攻击和破坏。

恶意流量分类繁多，常见的恶意流量包括病毒、蠕虫、木马、僵尸网络等。其中，病毒是指一种可以自我复制并感染其他程序的恶意代码，通常会破坏受害者的文件、系统或数据。蠕虫是指一种自我复制并通过网络传播的恶意代码，通常会破坏受害者的计算机和网络系统。木马是指一种悄悄地安装在受害者计算机上的恶意软件，可以被攻击者用来获取受害者计算机上的敏感信息。僵尸网络是指攻击者利用大量的僵尸主机构建的网络，用于进行 DDoS 攻击等破坏活动。

在网络安全领域，研究恶意流量可以帮助我们了解黑客的攻击方式和手段，从而提高网络的安全性。然而，恶意流量本身并不能直接揭示黑客的攻击目标和手法，因此我们需要进一步分析恶意流量中的 URL 信息。通过对恶意流量中的 URL 进行分析，我们可以发现这些 URL 与黑客的攻击行为之间的关联，并深入了解攻击的方式和目标。因此，将研究重心从恶意流量转移到恶意 URL 可以更加准确地分析黑客的攻击方式，从而为网络安全提供更有效的保障。

本篇文章的数据集采用 URL 的形式来表示。恶意 URL 是指用于欺骗用户访问的 URL，通常指具有欺骗性或恶意目的的 URL。恶意 URL 常常是网络钓鱼攻击的一种手段，攻击者会在 URL 中隐藏恶意代码，一旦用户点击该 URL，就会被重定向到恶意网站或下载恶意软件，导致用户的计算机和数据遭受攻击和破坏。

恶意 URL 的制作和传播方式多种多样，包括但不限于电子邮件、社交媒体、即时通讯工具、恶意广告等。攻击者通常会在诱导用户点击 URL 的过程中，伪装成合法的网站或者伪造看似可信的信息，以达到欺骗用户的目的。恶意 URL 的攻击方式对用户和企业的安全造成了极大的威胁，可以导致个人隐私泄露、金融损失、企业机密泄露等严重后果。

总之，恶意流量是网络安全领域中的一个重要问题，对于网络系统和数据的安全造成了巨大威胁。研究和发展恶意流量检测系统，采用机器学习等先进技术，可以有效提高网络安全的水平，保护网络系统和用户的安全。

2.3 常见的恶意攻击

此系统一共选取了五种不同的流量进行训练和检测，分别是正常流量，SQL 注入攻击流量，XSS 攻击流量，目录遍历攻击流量，CRLF 注入攻击流量，下面将对四种恶意攻击分别进行介绍。

（1）SQL 注入攻击

SQL 注入攻击（SQL Injection）[32]是一种针对 Web 应用程序的安全漏洞攻击方式。攻击者通过在 Web 应用程序中注入恶意的 SQL 代码，从而实现对数据库的非法访问和操作，导致数据泄露、数据损毁或服务拒绝等安全问题[33]。

SQL 注入攻击的原理是利用 Web 应用程序没有对用户输入进行严格的校验和过滤，攻击者通过构造特定的输入数据，欺骗应用程序将恶意的 SQL 代码作为正常的查询语句执行[34]。这些恶意的 SQL 代码可能包括但不限于删除数据、修改数据、获取敏感数据、升级权限等。

SQL 注入攻击常常针对使用 Web 应用程序与数据库进行交互的页面，如登录页面、搜索页面、用户信息页面等。攻击者通过在 URL 中加入恶意代码，或者在输入框中输入特定字符序列，从而执行恶意的 SQL 语句。一旦攻击成功，数据库中的敏感信息将面临泄露和损坏的风险。

（2）XSS 攻击

XSS 攻击（Cross-Site Scripting）是一种常见的 Web 应用程序安全漏洞攻击方式[35]，攻击者通过在 Web 页面中嵌入恶意脚本代码，从而实现对用户的非法控制，盗取用户的敏感信息，或者实现对 Web 应用程序的拒绝服务攻击等行为。

XSS 攻击通常通过向 Web 应用程序输入特定的恶意脚本代码实现。攻击者可能通过各种方式将这些恶意代码注入到 Web 页面中，比如在评论框中插入特殊字符、在表单中输入恶意代码等。一旦用户在浏览器中打开带有恶意脚本代码的 Web 页面，恶意代码将自动执行，并以攻击者预设的方式获取用户信息或者实现其他的攻击行为。

（3）目录遍历攻击

目录遍历攻击（Directory Traversal Attack）是一种常见的 Web 应用程序安全漏洞攻击方式，攻击者通过构造特定的 URL 或者请求，访问系统或者应用程序的目录或者文件，从而实现非法的文件读取、文件上传、文件删除等行为。

目录遍历攻击通常利用应用程序没有对用户输入的路径进行有效的过滤和验证，从而导致攻击者可以通过修改请求中的路径参数，访问任意文件或者目录。攻击者利用这种漏洞可以访问敏感文件或者数据，或者通过上传恶意文件等方式，实现对目标系统的攻击和控制。

（4）CRLF 注入攻击

CRLF 注入攻击是一种针对 Web 应用程序的安全漏洞攻击方式，攻击者利用这种漏洞，向应用程序发送包含特定控制字符的请求，从而实现对应用程序的攻击和控制。

CRLF 注入攻击通常利用应用程序对用户输入的数据没有进行有效的过滤和验证，从而导致攻击者可以通过发送包含 CRLF 字符的请求，实现 HTTP 报头注入、HTTP 响应截断、HTTP 重定向等攻击方式。攻击者利用这种漏洞可以伪造 HTTP 请求，修改系统的响应，甚至实现对系统的完全控制。

由于传统的基于规则的恶意流量检测方法难以应对不断增加的新型攻击，因此，机器学习技术在恶意流量检测中的应用日益受到关注。下面将主要介绍机器学习技术在恶意流量检测中的应用。首先，将简要介绍机器学习基础知识。其次，将详细阐述机器学习在恶意流量检测中的应用，包括数据预处理、特征提取和模型训练等。最后，将介绍常见的机器学习算法，并分析其优缺点。

2.4 机器学习基础知识

机器学习是人工智能的分支，是一种通过数据训练模型实现自主学习的算法。机器学习基于统计学、概率论和线性代数等数学基础，旨在让计算机从数据中学习，并通过学习结果来进行预测、分类、聚类等任务。

机器学习大致可分为监督学习、无监督学习和半监督学习[36]三种类型。

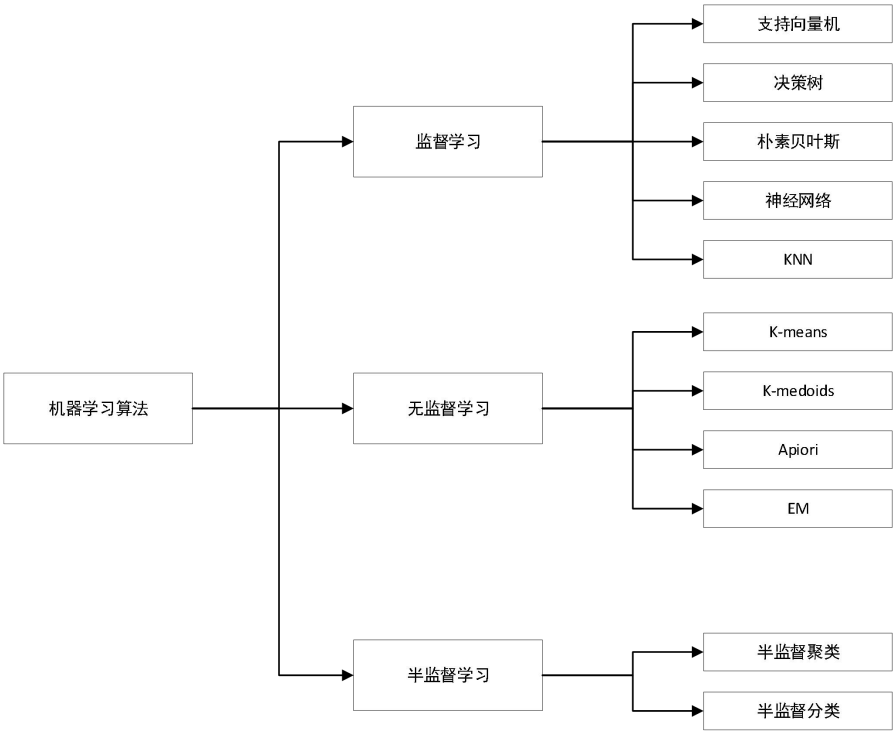

图 2-1 机器学习算法分类

（1）监督学习

监督学习是指从已知输入和输出的数据中学习并建立一个预测模型的过程。例如，给定一组已知的肿瘤患者数据，包括病人的年龄、性别、肿瘤大小等信息和他们是否患有癌症的标签，通过这些数据建立一个模型，预测患者是否患有癌症。

监督学习包括回归和分类两种类型。回归模型用于预测连续值，例如预测股票价格、房价等；而分类模型则用于预测离散值，例如预测病人是否患有癌症、邮件是否为垃圾邮件等。

（2）无监督学习

无监督学习是指在没有标签的情况下，从数据中发现潜在的结构、关系和模式的过程。例如，对于一组不带标签的图像数据，无监督学习可以帮助我们发现

它们之间的共性和差异。

无监督学习包括聚类和降维两种类型[37]。聚类模型用于将数据分组，例如将顾客分为不同的消费群体；而降维模型则用于将高维数据转化为低维数据，以便进行可视化或后续分析。

（3）半监督学习

半监督学习是指在部分数据带有标签的情况下，从这些带标签数据中学习并推广到未标记数据的过程。例如，在进行图像分类时，我们只对部分图像进行了标注，但是我们希望能够对未标注的图像进行分类。

半监督学习的核心思想是，通过带标签数据和未标签数据之间的关系，将未标签数据分为不同的类别。因此，半监督学习在数据量有限的情况下，可以显著提高分类准确率。

2.5 机器学习在恶意流量检测中的应用

机器学习在恶意流量检测中的应用可以大大提高检测的准确性和效率。传统的恶意流量检测方法主要是基于规则和签名的，需要事先确定规则或者签名，然后使用这些规则或者签名对流量进行检测。这种方法在对已知恶意流量的检测效果较好，但是对未知的恶意流量却无法有效检测，而且需要不断地更新规则和签名，需要耗费大量的人力和时间。

机器学习的出现解决了这一问题。机器学习可以通过学习大量的恶意流量样本，自动提取出这些流量的特征，并且可以通过对这些特征进行分类，判断流量是否为恶意流量。相比于传统方法，机器学习可以自动识别未知的恶意流量，并且可以不断地学习和适应新的恶意攻击方式。

机器学习在恶意流量检测中的应用可以分为有监督学习和无监督学习两种方式。有监督学习需要使用已知的恶意流量样本进行训练，然后利用训练得到的模型对新的流量进行分类。无监督学习则不需要使用已知的恶意流量样本进行训练，而是通过对流量数据进行聚类，将相似的流量归为一类，从而识别恶意流量。

在实际应用中，机器学习在恶意流量检测中的应用已经被广泛采用，例如IDS/IPS 系统、网络防火墙等。通过将机器学习技术应用到恶意流量检测中，可以大大提高网络安全性，减少恶意攻击对网络带来的危害。

2.6 常见的机器学习算法及其优缺点

在机器学习中，有许多常见的算法可以用于恶意流量检测。这里我们简单介绍几种常见的机器学习算法及其优缺点：

（1）支持向量机（Support Vector Machine，SVM）

支持向量机（Support Vector Machine，SVM）[38]是一种经典的监督学习算法，常用于二分类和多分类任务。它的基本思想是找到一个最优的超平面，将不同类别的样本分开，并且使得不同类别的样本到该超平面的距离最大化。SVM 可以用于分类和回归任务，其中分类任务更为常见。如图 2-2 所示，超平面 B 是正确的超平面。

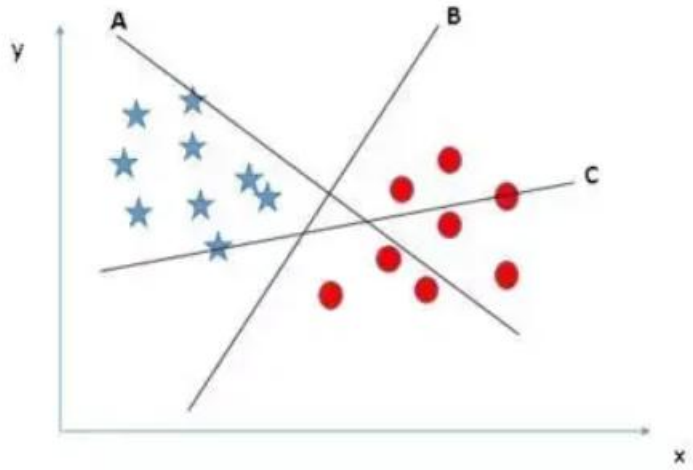

图 2-2 SVM 算法超平面示例

SVM 的优点包括：

1) 可以解决高维空间和非线性分类问题：SVM 通过使用核函数将低维数据映射到高维空间中，从而能够解决非线性问题。

2) 模型泛化能力强：在构建分类器时，SVM 考虑了样本空间的全局特征，可以更好地适应未知的测试数据。

SVM 的缺点包括：

1) 对数据的敏感度：SVM 对异常值和噪声数据非常敏感，这可能导致模型性能的下降。

2) 训练时间较长：对于大规模数据集，SVM 训练时间可能非常长。

3) 参数调整困难：SVM 的性能受到模型参数的影响，参数调整困难可能导致

性能下降。

（2）K-NN 算法

K-NN 算法[39]的原理是：存在一个训练样本集合，该集合中每行数据包含多个特征和分类标签，输入没有标签但有多个特征的新数据，将新数据的每个特征与样本中每条数据对应的特征进行比较，然后提取出样本中与新数据最相似的 K 条数据，统计该 K 条数据中各类标签出现的次数，那么出现次数最多的标签即为新数据的分类标签。

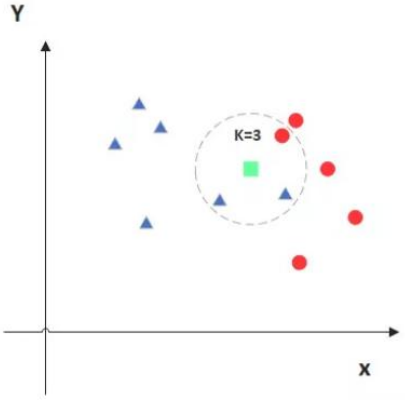

图 2-3 K-NN 算法原理图 1

如图 2-3 所示，当 K 值为 3 时，由于距离待检测样本最近的 3 个数据包括两个蓝色三角形和一个红色圆形，因此待检测样本的标签被预测为蓝色三角形。

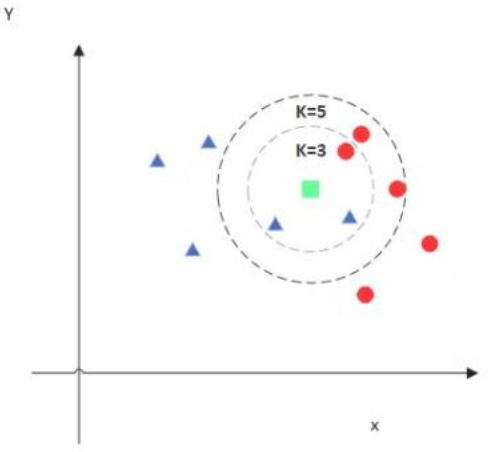

图 2-4 K-NN 算法原理图 2

如图 2-4 所示，由于此时 K 值由 3 更改成为了 5，此时距离待检测样本最近的 5 个数据包括两个蓝色三角形和三个红色圆形，因此待检测样本的标签被预测为红色圆形。

KNN 算法的优点包括：

1）KNN 算法对数据没有假设，因此可以适用于各种数据类型。

2) KNN 算法适用于多分类问题。

KNN 算法的缺点包括：

1) KNN 算法对于高维数据的处理效果较差。

2) KNN 算法在处理大规模数据时计算量较大，需要较长的时间。

（3）逻辑回归算法

逻辑回归（Logistic Regression，LR）[40]是一种常见的分类算法，它主要用于二分类问题，即将数据分为两个不同的类别。该算法基于统计学方法，通过对数据进行拟合，得到一个逻辑回归模型，该模型可以用于预测新数据的分类。通常，逻辑回归使用某种函数将概率值压缩到某一特定范围。例如，Sigmoid 函数（S 函数）是一种具有 S 形曲线、用于二元分类的函数。它将发生某事件的概率值转换为 0，1 的范围表示。Sigmoid 函数如下所示：

$$S(x) = \frac{1}{1 + exp(-x)} \tag{2-1}$$

Sigmoid 函数图像如图 2-5 所示：

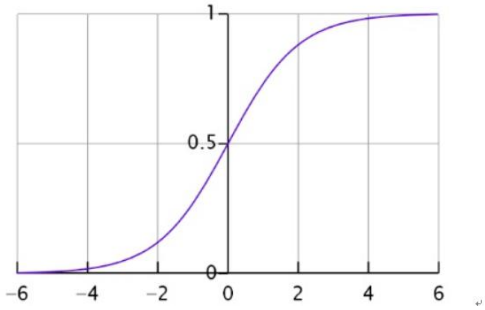

图 2-5 sigmoid 函数的图像

逻辑回归算法的优点包括：

1) 计算效率高，可用于大规模数据集。

2) 输出结果可以直观地表示为概率值。

逻辑回归算法的缺点包括：

1) 只能处理二分类问题，无法处理多分类问题。

2) 对于数据集中的异常值和噪声比较敏感。

在上述相关技术的基础上，本文将通过设计与实现一个基于机器学习的恶意流量检测系统来探讨机器学习算法在恶意流量检测中的应用，并通过测试结果判断 KNN 算法和逻辑回归算法在本系统上的检测效果更好。

而在机器学习领域，数据处理和特征提取是至关重要的环节。随着信息技术

的不断发展和应用，我们已经进入了一个大数据时代，海量的数据对机器学习模型的性能提出了更高的要求。在机器学习模型的训练过程中，数据的质量和特征的有效性直接影响到模型的准确性。因此，数据处理和特征提取成为了机器学习中至关重要的步骤。

3 数据处理与特征提取

本章将围绕数据处理和特征提取展开讨论，并探究其中的相关技术和方法。首先，我们将介绍数据预处理的基本概念和技术，包括数据清洗、归一化、去噪等。接着，我们将重点探讨特征提取的相关技术，如基于统计方法、基于机器学习的特征选择、卷积神经网络等。最后，我们将结合实际案例对数据处理和特征提取的重要性进行探究，并讨论如何优化数据处理和特征提取过程，提高机器学习模型的性能。

3.1 数据集收集与预处理

3.1.1 数据集收集

此系统选择的数据集是在 Github 上收集的一些 payloads 和 HTTP DATASET CSIC 2010 数据集，其中 HTTP DATASET CSIC 2010 数据集是由西班牙萨拉戈萨大学计算机科学与工程系收集的公共 HTTP 流量数据集。该数据集收集了一些常见的 Web 应用程序，例如 SQL 注入、跨站点脚本攻击（XSS）和远程文件包含（RFI）等攻击类型的 HTTP 请求。

在 github 上采集到的恶意流量数据集绝大多数都是已经打好标签的，一共包含五种不同种类的网络流量，分别是正常流量，SQL 注入攻击流量，XSS 攻击流量，目录遍历攻击流量，CRLF 注入攻击流量。由于数据集的格式并不统一，为了方便后序实验的进行，利用一个 python 脚本将所有的 txt 格式的数据集都统一转换为 CSV 格式的数据集并把采集到的数据集手动的分为训练集和测试集保存为本地 CSV 文件。

恶意流量训练模型的数据集由正常流量数据集充当负例，该类攻击流量数据集充当正例，而正常流量训练模型的数据集由正常流量数据集充当正例，其他四种恶意流量数据集充当负例。每种类型训练集的数据量如表 3-1 所示：

表 3-1 每种模型训练集的数据量

	正常流量	SQL 注入	XSS 攻击	目录遍历	CRLF 注入
正例	18504	5035	5022	5082	2083
负例	17985	5803	5359	5158	2098

3.1.2 数据集预处理

预处理是对数据进行必要的清洗、筛选、转换和标准化等操作，以使数据更适合进行后续分析和建模。在恶意流量数据分析中，预处理非常重要，因为原始数据通常包含噪声、重复记录、缺失值和不一致性等问题。为了确保分析的可靠性和准确性，必须对数据进行预处理。

原始数据集中有些数据是由 web 日志的形式组成，为了提取其中的 http 头，使用了 python 中的函数进行处理，步骤如下：

1) 打开 Web 日志文件，使用 Python 内置的 open 函数读取文件内容。

2) 遍历日志文件中的每一行，使用字符串函数判断该行是否包含 HTTP 请求。

3) 如果该行包含 HTTP 请求，可以使用正则表达式或字符串分割操作从该行中提取出 HTTP 请求部分，通常是包括 HTTP 方法、URL、协议版本和一些 HTTP 头部信息等。

4) 将提取出来的 HTTP 请求部分保存到另一个文件或数据结构中，以便后续处理和分析。

数据集中很多数据都是包含了大量的 URL 转义符，URL 转义符是一种用于将某些字符编码成 URL 安全字符的机制。在 URL 中，某些字符具有特殊含义，例如问号、等号、斜杠、空格等，如果直接在 URL 中使用这些字符可能会导致 URL 解析错误。为了避免这种情况，需要使用 URL 转义符将这些字符编码成安全的 URL 字符，使其能够正确地在网络上传输。为了可以在后续特征提取的过程中更加准确和直观的生成特征向量，使用了 python 脚本自动实现 URL 的解码，其中关键代码为 unescaped_line = urllib.parse.unquote(line，encoding='utf-8'，errors='replace')，这行代码使用 Python 的 urllib.parse 模块中的 unquote 函数将 URL 字符串进行解码，并将结果赋值给变量 unescaped_line。unquote 函数用于将 URL 中的特殊字符转义为其原始字符。

在对 URL 进行解码后发现有部分 URL 中出现了乱码的情况，查找资料后发现是因为有些恶意攻击是在利用不同操作系统的不同版本针对性的进行攻击，例如在中文版本的 Win2000 中，Unicode 编码就存在着 bug，在 Unicode 编码中会将%c1%1c 解释为'/'，而黑客正是利用这个 bug 绕开了安全检测进行目录遍历攻击，但%c1%1c 这样的 URL 编码在解码后并无重要意义，因此面对这样的攻击采取的

预处理方法是手动对这样的 URL 编码进行解码。

3.2 特征提取

恶意流量检测系统的特征提取部分是整个系统中最为关键的一部分。该部分主要通过分析网络流量数据的特征来识别潜在的恶意行为。在本文中，我们将采用机器学习的方法来实现特征提取，并利用 KNN 算法和逻辑回归算法进行恶意流量的分类。

根据不同种类恶意流量的不同特征，本文采取对不同的恶意流量分别进行特征提取的方法，不同种类恶意流量的特征向量维数并不统一，利用 python 脚本将每种恶意流量的特征向量提取出后，需要对提取出来的特征进行预处理和归一化。这是因为不同的特征具有不同的尺度和分布，如果不进行处理，可能会对机器学习算法的效果产生很大的影响。在本文中，我们将采用 Z-score 的方法来对特征进行归一化处理，即将每个特征值减去其均值，再除以其标准差。

3.2.1 SQL 注入攻击的特征向量

表 3-2 是一些常见的用于机器学习检测 SQL 注入攻击的特征向量：

表 3-2 SQL 注入攻击的特征向量

序号	SQL 注入攻击的特征向量
1	字符个数
2	SQL 注入关键词数量
3	大写字符百分比
4	数字字符百分比
5	空格百分比
6	特殊字符百分比
7	前缀百分比

（1）字符个数：恶意 SQL 代码通常比合法输入更长，因此输入长度可以作为 SQL 注入攻击的一个特征向量。

（2）SQL 注入关键词数量：恶意 SQL 代码通常包含 SQL 关键词，如 SELECT、FROM、WHERE 等。因此，SQL 关键词数量可以作为 SQL 注入攻击的一个特征

向量。

（3）大写字符百分比：恶意 SQL 代码通常包含大写字母，因此大写字符百分比可以作为 SQL 注入攻击的一个特征向量。

（4）数字字符百分比：恶意 SQL 代码通常包含数字字符，因此数字字符百分比可以作为 SQL 注入攻击的一个特征向量。

（5）空格百分比：恶意 SQL 代码通常不包含空格，因此空格百分比可以作为 SQL 注入攻击的一个特征向量。

（6）特殊字符百分比：SQL 注入攻击通常涉及特殊字符，如单引号、分号、括号、反斜杠等。因此，特殊字符百分比可以作为 SQL 注入攻击的一个特征向量。

（7）前缀百分比：恶意 SQL 代码通常以特定的前缀开始，例如 OR、AND 等。因此，前缀百分比可以作为 SQL 注入攻击的一个特征向量。

其中 SQL 注入关键词如表 3-3 所示：

表 3-3 SQL 注入关键字

序号	SQL 注入关键字	序号	SQL 注入关键字
1	and	13	aes
2	or	14	xp_cmdshell
3	xor	15	exec
4	version	16	union
5	substr	17	order
6	substring	18	information schema
7	len	19	sleep
8	length	20	md5
9	benchmark	21	database
10	shutdown	22	load_file
11	mid	23	load data infile
12	into outfile	24	into dumpfile

用于提取 SQL 注入攻击特征向量的函数逻辑流程如下：

（1）定义函数 generate_url_sql，输入参数为 url，即待检测的 URL。

（2）初始化 num_len（字符数量）为 0，capital_len（大写字母数量）为 0，key_num 为 0（关键字数量），capital_f 为 0（大写字符所占比例），num_f 为 0（关键字占比

例），space_f 为 0（空格所占比例），special_f 为 0（特殊字符所占比例），prefix_f（特定前缀所占比例）为 0。

（3）利用正则表达式 re.compile(r'\d')统计数字字符的数量，存储在 num_len 中。

（4）如果 url 的长度不为 0，则计算数字字符的频率，存储在 num_f 中。

（5）利用正则表达式 re.compile(r'[A-Z]') 统计大写字母的数量，存储在 capital_len 中。

（6）如果 url 的长度不为 0，则计算大写字母的频率，存储在 capital_f 中。

（7）利用 lower()函数将 url 转换为小写字母。

（8）利用 count()函数统计 url 中出现的 SQL 注入关键词数量，存储在 key_num 中。

（9）如果 url 的长度不为 0，利用 count()函数则计算空格的百分比，存储在 space_f 中。

（10）如果 url 的长度不为 0，则利用 count()函数前缀字符（'\x'或'\u'）的百分比，存储在 prefix_f 中。

（11）统计 url 中出现的特殊字符数量，并计算特殊字符的百分比，存储在 special_f 中。

（12）将特征值以列表的形式存储在 feature 中。

（13）返回 feature。

3.2.2 XSS 攻击的特征向量

表 3-4 XSS 攻击特征向量

序号	XSS 攻击特征向量
1	字符个数
2	特殊字符百分比
3	大写字符百分比
4	前缀百分比
5	XSS 攻击关键字数量

表 3-4 是一些常见的用于机器学习检测 SQL 注入攻击的特征向量。

（1）攻击载荷的长度：XSS 攻击常常使用长载荷，以便注入足够的恶意代码，因此长度是一项重要的特征。

（2）攻击载荷的特殊字符：XSS 攻击常常使用一些特殊字符来欺骗目标网站的输入检查，这些特殊字符包括<、>、/、'、"等等。

（3）大写字符的比例：XSS 攻击载荷中大写字母的比例较高，这是因为攻击者通常会使用大小写混合的字符串来欺骗输入检查。

（4）前缀的比例：攻击载荷中是否包含特定的前缀，如“\x”或“\u”，这些前缀通常被用于注入恶意代码。

（5）关键词数量：XSS 攻击载荷中关键词的数量，例如“script”、“alert”等等，这些关键词通常被用于注入恶意代码。

其中 XSS 攻击关键字如表 3-5 所示：

表 3-5 XSS 攻击关键字

序号	XSS 攻击关键字
1	script
2	alert
3	src
4	href
5	import
6	eval
7	document
8	javascript
9	iframe
10	onerror
11	sleep

用于提取 XSS 攻击特征向量的函数逻辑流程如下：

（1）定义函数 generate_url_xss，输入参数为 url，即待检测的 URL。

（2）定义并初始化 num_len（字符数量）为 0，capital_len（大写字母数量）为 0，key_num 为 0（关键字数量），capital_f 为 0（大写字符所占比例），num_f 为 0（关键字占比例），special_f 为 0（特殊字符所占比例），prefix_f（特定前缀所占比例）为 0。

（3）利用正则表达式 re.compile(r'[A-Z]') 统计大写字母的数量，存储在 capital_len 中。

（4）如果 url 的长度不为 0，则计算大写字母的频率，存储在 capital_f 中。

（5）利用 lower()函数将 url 转换为小写字母。

（6）利用 count()函数统计 url 中出现的 XSS 攻击关键词数量，存储在 key_num 中。

（7）如果 url 的长度不为 0，则利用 count()函数前缀字符（'\x'或'\u'）的百分比，存储在 prefix_f 中。

（8）统计 url 中出现的特殊字符数量，并计算特殊字符的百分比，存储在 special_f 中。

（9）将特征值以列表的形式存储在 feature 中。

（10）返回 feature。

3.2.3 目录遍历攻击的特征向量

表 3-6 是一些常见的用于机器学习检测目录遍历攻击的特征向量：

表 3-6 目录遍历攻击特征向量

序号	目录遍历攻击特征向量
1	关键字出现次数
2	..出现次数
3	//出现次数
4	/出现次数
5	.出现次数
6	特殊字符百分比

（1）关键词出现次数：检测 URL 中是否包含常见的目录遍历攻击关键词，出现次数越多表示越有可能是目录遍历攻击。

（2）..出现次数：检测 URL 中..的出现次数。

（3）//出现次数：检测 URL 中//的出现次数。

（4）/出现次数：检测 URL 中/的出现次数。

（5）.出现次数：检测 URL 中.的出现次数。

（6）特殊字符百分比：计算 URL 中特殊字符（如$、*、%等）的出现频率，出现频率越高表示越有可能是目录遍历攻击。

其中目录遍历攻击关键字如表 3-7 所示：

表 3-7 目录遍历攻击特征值

序号	目录遍历攻击特征值
1	etc
2	passwd
3	system

用于提取目录遍历攻击特征向量的函数逻辑流程如下：

（1）定义函数 generate_url_traversal，输入参数为 url，即待检测的 URL。

（2）初始化 len1（..数量）为 0，len2（//数量）为 0，len3 为 0（.数量），len4 为 0（/数量），num_f 为 0（关键字占比例），special_f 为 0（特殊字符所占比例）。

（3）利用 lower()函数将 url 转换为小写字母。

（4）利用 count()函数分别统计\\，..，\，.，出现的次数并存储在 len1，len2，len3，len4 中。

（5）利用 count()函数统计 url 中出现的目录遍历攻击关键词数量，存储在 key_num 中。

（6）统计 url 中出现的特殊字符数量，并计算特殊字符的百分比，存储在 special_f 中。

（7）将特征值以列表的形式存储在 feature 中。

（8）返回 feature。

3.2.4 CRLF 注入攻击特征向量

表 3-8 是一些常见的用于机器学习检测 CRLF 注入攻击的特征向量：

表 3-8 CRLF 注入攻击的特征向量

序号	CRLF 注入攻击的特征向量
1	%a 出现的次数
2	%d 出现的次数
3	%a%d 出现的次数

这些特征值是为了检测 URL 中是否存在 CRLF 注入而设计的。CRLF 注入攻击利用换行符（\r\n）来修改 HTTP 响应，插入恶意代码或执行其他恶意操作。攻击者通过在 URL 参数中插入特殊字符和标记来实现 CRLF 注入。

因此，检测 CRLF 注入的特征向量主要是计算 URL 中出现的特殊字符的数量，

特别是 CRLF 注入中常用的特殊字符，如%a、%d。这些字符可以被用来插入换行符、空行或其他不可见的字符，从而导致 HTTP 响应受到攻击。

用于提取 crlf 注入攻击特征向量的函数逻辑流程如下：

（1）定义函数 generate_url_crlf，输入参数为 url，即待检测的 URL。

（2）初始化 len1（%a 数量）为 0。

（3）初始化 len2（%d 数量）为 0。

（4）初始化 len3（%a%d 数量）为 0。

（5）利用 lower()函数将 url 转换为小写字母。

（6）利用 count()函数统计%a 出现的次数并存储在 len1 变量中。

（7）利用 count()函数统计%d 出现的次数并存储在 len2 变量中。

（8）利用 count()函数统计%a%d 出现的次数并存储在 len3 变量中。

（9）将特征值以列表的形式存储在 feature 中。

（10）返回 feature。

3.2.5 正常流量的特征向量

因为正常流量的形式过于复杂和多样，特征值难以提取，因此本文采用提取上述四种恶意攻击流量的特征值用来检测异常流量并表示为负例，将正常流量表示为正例，而提取的特征向量就包含上述四种攻击特征向量的全部。

表 3-9 异常流量的特征向量

序号	异常流量特征向量	序号	异常流量特征向量	序号	异常流量特征向量
1	..出现的次数	7	%a%d 出现的次数	13	关键词出现的次数
2	.出现的次数	8	字符数量	14	空格百分比
3	//出现的次数	9	特殊字符百分比		
4	/出现的次数	10	大写字符百分比		
5	%a 出现的次数	11	前缀百分比		
6	%d 出现的次数	12	数字字符百分比		

表 3-9 为包含了 SQL 注入攻击、XSS 攻击、目录遍历攻击和 CRLF 注入攻击四种异常流量的特征向量。其中包括了..出现的次数、字符数量、特殊字数百分比、大写字符百分比、前缀百分比、数字字符百分比、关键字出现的次数和空格百分比等。

其中关键词如表 3-10 所示：

表 3-10 恶意攻击关键词

序号	恶意攻击关键词	序号	恶意攻击关键词
1	and	20	database
2	or	21	load_file
3	xor	22	into outfile
4	version	23	into dumpfile
5	substr	24	script
6	substring	25	javascript
7	len	26	alert
8	length	27	src
9	benchmark	28	href
10	shutdown	29	import
11	mid	30	eval
12	aes	31	document
13	cmdshell	32	javascript
14	exec	33	iframe
15	union	34	onerror
16	order	35	sleep
17	information schema	36	etc
18	sleep	37	passwd
19	md5	38	system

用于提取异常流量特征向量的函数逻辑流程如下：

（1）定义函数 generate_url_abnormal，输入参数为 url，即待检测的 URL。

（2）初始化 num_len（字符数量）为 0，capital_len（大写字母数量）为 0，key_num 为 0（关键字数量），capital_f 为 0（大写字符所占比例），num_f 为 0（关键字占比例），space_f 为 0（空格所占比例），special_f 为 0（特殊字符所占比例），prefix_f（特定前缀所占比例）为 0 等。

（3）利用正则表达式 re.compile(r'\d')统计数字字符的数量，存储在 num_len 中。

（4）如果 url 的长度不为 0，则计算数字字符的频率，存储在 num_f 中。

（5）利用正则表达式 re.compile(r'[A-Z]') 统计大写字母的数量，存储在 capital_len 中。

（6）如果 url 的长度不为 0，则计算大写字母的频率，存储在 capital_f 中。

（7）利用 lower()函数将 url 转换为小写字母。

（8）利用 count()函数统计 url 中出现的关键词数量，存储在 key_num 中。

（9）利用 count()函数分别统计\\，..，\，.，出现的次数并存储在 len1，len2，len3，len4 中。

（10）利用 count()函数分别统计%a，%d，%a%d 出现的次数并存储在 len1，len2，len3 中。

（11）如果 url 的长度不为 0，利用 count()函数则计算空格的百分比，存储在 space_f 中。

（12）如果 url 的长度不为 0，则利用 count()函数前缀字符（'\x'或'\u'）的百分比，存储在 prefix_f 中。

（13）统计 url 中出现的特殊字符数量，并计算特殊字符的百分比，存储在 special_f 中。

（14）将特征值以列表的形式存储在 feature 中。

（15）返回 feature。

3.2.6 提取 csv 文件中恶意流量的特征向量

上述功能都是提取单个 URL 的函数实现，提取 csv 文件中恶意流量的特征向量的函数类似，只是在入口参数和返回值上有所区分。入口参数有三个，分别是 odir，wdir，lable，其中 odir 表示要打开的 csv 文件的路径，wdir 表示要写入特征向量的 csv 文件的路径，label 表达要打开的文件是正例还是负例；而在返回值上不再将特征向量作为返回值返回，而是写入 csv 文件中，并在特征向量列表的添加一列 label 表示正负例。

4 恶意流量检测系统的设计与实现

本章将通过设计与实现一个基于机器学习的恶意流量检测系统来探讨机器学习算法在恶意流量检测中的应用，旨在提高对网络安全的保护能力。本章主要包括三个部分：首先，介绍基于机器学习的恶意流量检测系统的设计与实现流程；其次，实现机器学习算法在恶意流量检测中的应用；最后，针对设计实现的系统进行实验验证，探究其性能表现。通过本章的研究，可以为恶意流量检测技术的研究提供借鉴，同时提高网络安全的保护能力。

4.1 系统架构设计

本文选择了 KNN 算法和逻辑回归算法作为系统的机器学习核心算法，且 KNN 算法的 K 值选定为默认值 5，根据五种不同的数据集分别得到共十种训练模型，分别是基于 KNN 算法的正常流量模型、基于 KNN 算法的 SQL 注入攻击流量模型、基于 KNN 算法的 XSS 攻击流量模型、基于 KNN 算法的目录遍历攻击流量模型、基于 KNN 算法的 CRLF 注入攻击流量模型、基于逻辑回归算法的正常流量模型、基于逻辑回归算法的 SQL 注入攻击流量模型、基于逻辑回归算法的 XSS 攻击流量模型、基于逻辑回归算法的目录遍历攻击流量模型、基于逻辑回归算法的 CRLF 注入攻击流量模型。考虑到系统实现的功能是一个多分类的判断，如果直接使用多分类的训练和测试会要求每个种类的流量提取出的特征向量维数要相同，但是实际上每种不同的流量的特征值和特征点都不相同，很难统一出一个相同的维数，因此系统针对每种不同的流量分别进行模型训练，恶意流量训练模型的数据集由正常流量数据集充当负例和该类攻击流量数据集充当正例组成，而正常流量训练模型的数据集由正常流量数据集充当正例和其他四种恶意流量数据集充当负例组成。系统训练模型框架图如图 4-1 所示：

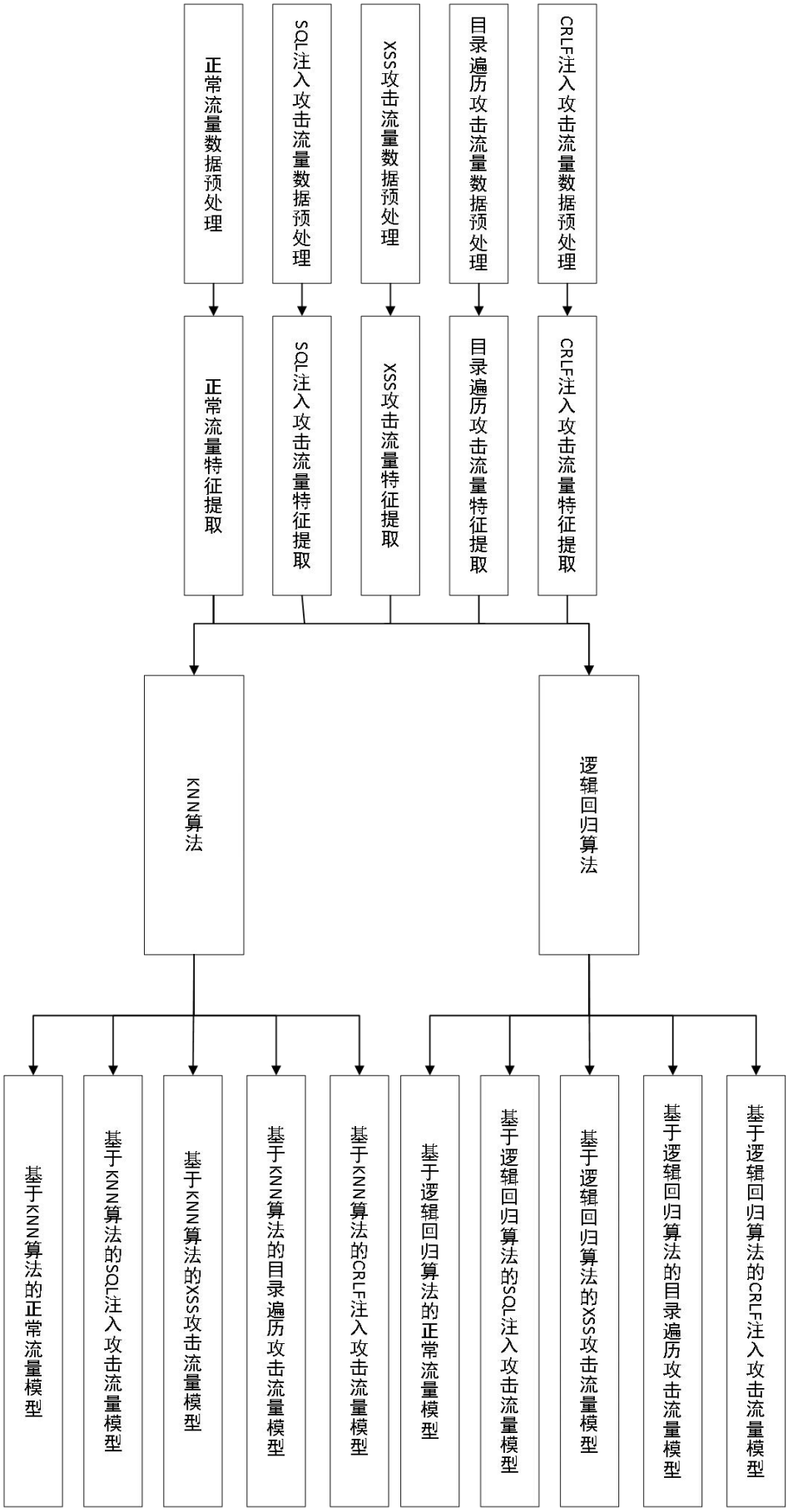
正常流量数据预处理
SQL注入攻击流量数据预处理
XSS攻击流量数据预处理
目录遍历攻击流量数据预处理
CRLF注入攻击流量数据预处理
正常流量特征提取
SQL注入攻击流量特征提取
XSS攻击流量特征提取
目录遍历攻击流量特征提取
CRLF注入攻击流量特征提取
KNN算法
逻辑回归算法
基于KNN算法的正常流量模型
基于KNN算法的SQL注入攻击流量模型
基于KNN算法的XSS攻击流量模型
基于KNN算法的目录遍历攻击流量模型
基于KNN算法的CRLF注入攻击流量模型
基于逻辑回归算法的正常流量模型
基于逻辑回归算法的SQL注入攻击流量模型
基于逻辑回归算法的XSS攻击流量模型
基于逻辑回归算法的目录遍历攻击流量模型
基于逻辑回归算法的CRLF注入攻击流量模型

图 4-1 训练模型图

系统由三大功能组成，第一个是根据用户的输入选择不同的算法和不同的数据集进行训练，训练完成后生成一份训练报告，其中包括此次训练的准确率，回归率，F1 值和混淆矩阵；第二个功能是根据用户的输入选择不同类型的测试数据集进行模型的测试功能，测试完成后生成一份测试报告，其中包括此次训练的准确率，回归率，F1 值和混淆矩阵；第三个功能是用户选择输入单个 URL 或一个 csv 数据集进行测试，测试完成后系统会生成一个报告，若输入为单个 URL，则报告为十个模型对该 URL 的检测结果（1 为真 0 为假），当且仅当一个算法中只有一个模型测试结果为真且其他模型测试结果为假时，系统判断该算法的最终结果为真的那个结果，否则判断为其他种类恶意攻击流量；若输入为一个 csv 数据集，则最终的报告包括每种攻击流量的数量以及所占比例。图 4-2 为系统结构框架图。

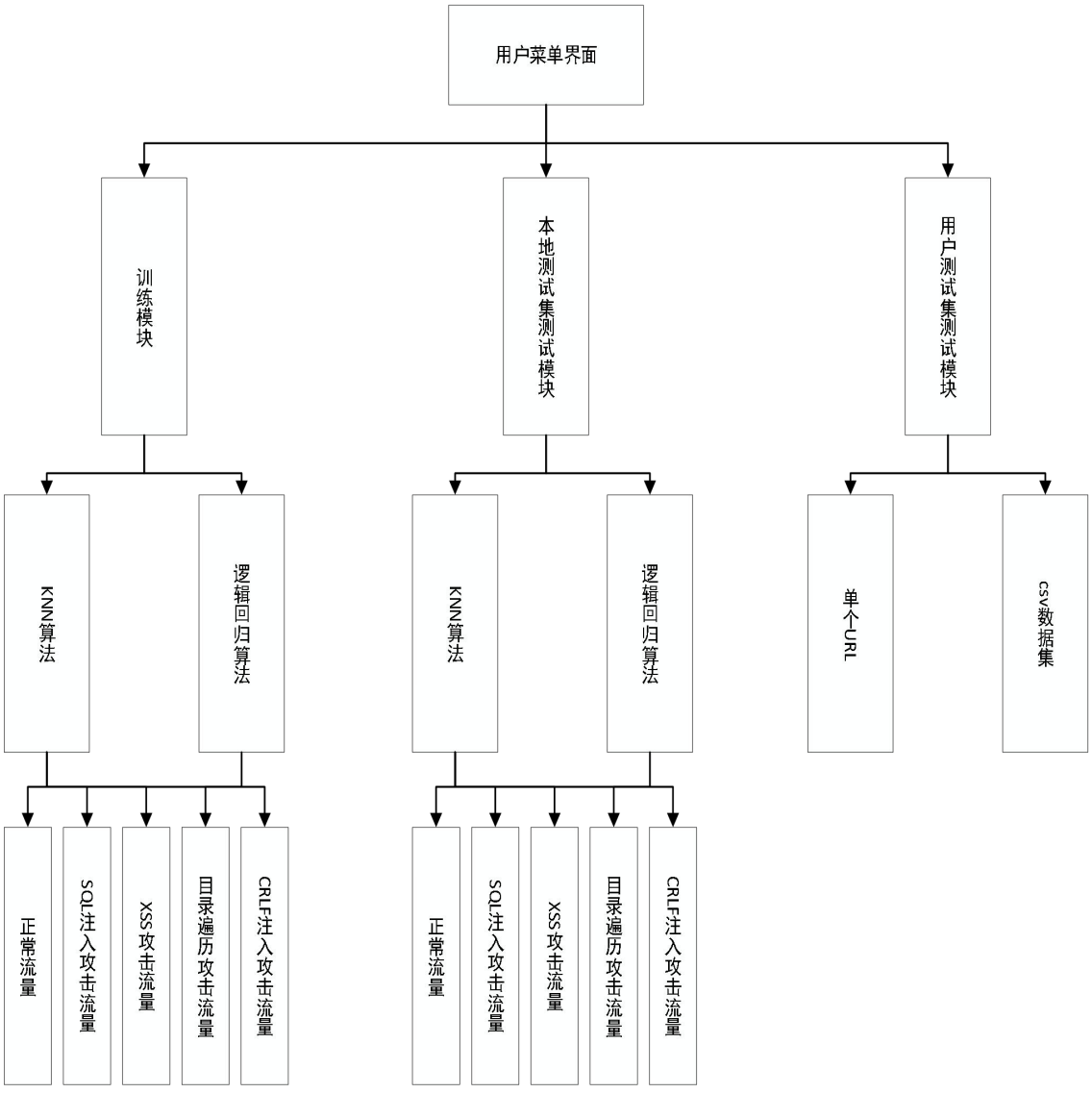

图 4-2 系统结构框架图

4.2 模型训练

因为不同种类流量的训练过程只是在打开特征向量文件和最终生成的模型上有所不同，其余框架都一样，因此本文只介绍 KNN 算法和逻辑回归算法训练实现的框架。

4.2.1 KNN 算法模型训练框架

KNN 算法是一种基于相似度度量的分类和回归算法，在机器学习领域得到了广泛应用。在训练 KNN 模型时，需要使用一个训练框架来组织和管理整个过程。在整个训练过程中，需要注意数据的质量和数量、K 值的选择、距离度量方法的选择等问题，以确保训练出一个高效、准确的模型。同时，由于 KNN 算法是一种“懒惰学习”方法，因此在实际应用中需要注意其计算成本较高的问题。

（1）首先根据具体的恶意流量种类定义了一个函数，该函数有一个输入参数 signal。

（2）在该函数中调用了 generate 函数，分别生成了恶意攻击和正常请求的样本特征矩阵，并保存到了对应的 csv 文件中。

（3）接着使用 pandas 读取了特征矩阵文件，并分别将两个矩阵合并保存到了同一个 csv 文件中，并把 label 列保存进 target 数组中用于检测，删除了最后一列 label，得到特征数据和目标数据。

（4）通过 sklearn 中的 train_test_split 函数将数据集划分为训练集和测试集，划分比例为测试集占总数据集的 30%。

（5）创建一个名为 KNN 的分类器对象，设置最大迭代次数为 10000，然后使用训练集进行训练。

（6）使用 joblib.dump 函数将训练好的分类器保存到文件中，文件路径为 './model/logistic_sql.model'。

（7）使用训练好的分类器对测试集进行预测，得到预测结果 y_pred。

（8）计算分类器预测结果的准确度、召回率和 F1 值，并输出混淆矩阵。

（9）将结果以字符串的形式保存到控制台的输出信息中，并通过 emit 函数将结果发射出去。

图 4-3 为 KNN 算法模型训练框架图。

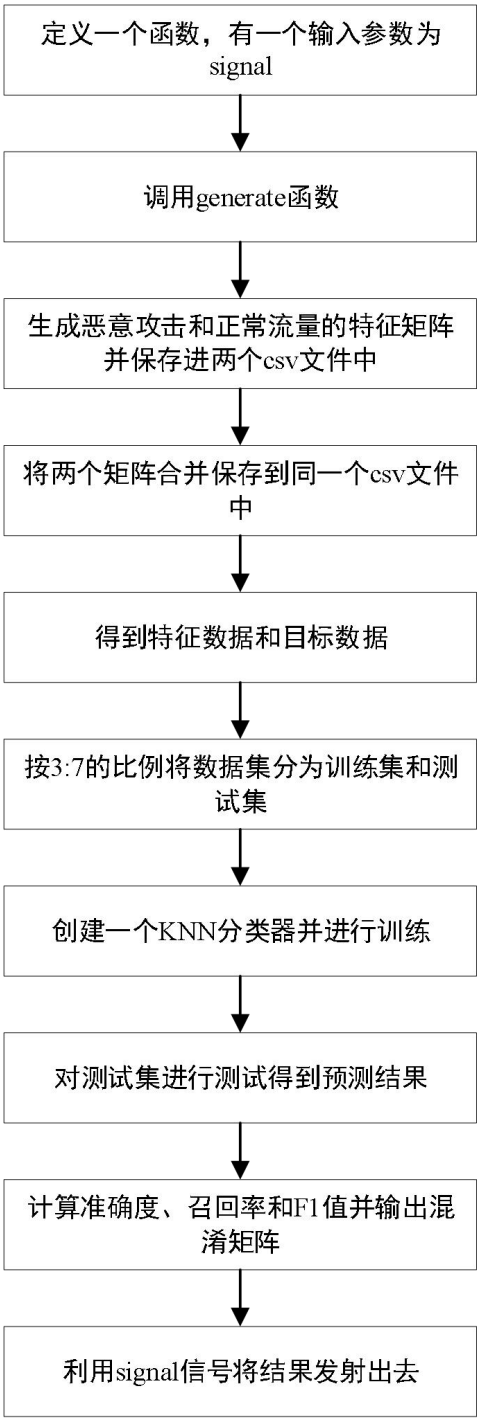

图 4-3 KNN 算法模型训练框架

一般来说，训练集和测试集的比例划分应该保证测试集的数量足够进行模型的评估，但是又不要太大，否则会导致训练集的数量不足以训练模型。3:7 是一个常用的取值，通常被认为可以平衡测试集和训练集之间的比例关系。

4.2.2 逻辑回归算法模型训练框架

逻辑回归是一种用于分类问题的经典算法，在机器学习领域得到了广泛应用。在训练逻辑回归模型时，通常需要使用一个训练框架来组织和管理整个过程。该框架通常包括以下几个步骤：首先，需要准备训练数据集和测试数据集，通常需要将数据集进行预处理和特征工程。接下来，需要选择适当的损失函数和优化算法来训练模型。在训练过程中，还需要对模型进行评估和调优，以确保其在测试集上的性能达到最佳。最后，需要对模型进行验证和部署，以便将其应用于实际场景中。在整个训练过程中，需要注意数据的质量和数量、超参数的选择、过拟合和欠拟合等问题，以确保训练出一个高效、准确的模型。以下是针对本系统设计的逻辑回归算法模型训练流程。

（1）导入需要的库和模块，包括 numpy、pandas、sklearn 等，以及自定义的函数和类。

（2）读取两个 csv 文件，其中一个包含正常数据的特征矩阵，一个包含 SQL 注入攻击的特征矩阵，将这两个文件合并成一个 csv 文件，并将其存储在指定路径下。

（3）从合并后的 csv 文件中读取数据，并将其转换为 NumPy 数组，将其中的最后一列删除作为数据特征，将第 7 列作为数据标签。

（4）将数据集随机划分为训练集和测试集，划分比例为 70%的训练集和 30%的测试集。

（5）创建逻辑回归分类器对象，使用 ball_tree 算法，k 值默认为 5，对训练集进行训练。

（6）将训练好的逻辑回归分类器模型保存到指定路径下的文件中，以便之后可以快速地加载模型进行预测。

（7）使用测试集对模型进行验证，计算模型的准确率、召回率、F1 值以及混淆矩阵，并将结果输出到控制台。

（8）发射信号，将输出结果传递给控制界面，以便在界面上显示。

逻辑回归算法模型训练框架和上述 KNN 算法模型训练框架类似，只是在创建分类器时 KNN 算法模型创建的是 KNN 分类器来进行模型的训练，而逻辑回归算法模型在创建分类器时创建的是逻辑回归分类器来进行模型的训练。

4.3 重点系统功能函数介绍

本小节将针对系统的三个功能函数做详细的介绍。三个功能函数分别是 train、test、user_test，分别对应功能为不同算法不同种类流量的训练功能、本地测试集测试功能、用户输入数据测试功能。

4.3.1 train 函数介绍

train 函数实现了一个菜单式的模型训练功能，可以训练 KNN 模型或逻辑回归模型，并选择不同类型的数据进行训练。该函数包含了一个无限循环，直到用户选择退出才停止。在循环中，首先输出菜单供用户选择训练的模型类型（KNN 或逻辑回归），如果用户选择退出，则跳出循环；如果选择训练 KNN 模型，则再次输出菜单供用户选择训练的数据类型，并根据用户的选择调用相应的训练函数；如果选择训练逻辑回归模型，则与训练 KNN 模型相似，也是根据用户的选择调用相应的训练函数。在菜单输出过程中，通过 emit 函数将输出的信息传递给信号对象。输入信息则从 input_queue 队列中获取。整个函数的逻辑是：不断输出菜单，获取用户的输入，并根据用户的选择调用相应的训练函数。该函数的函数逻辑如下：

（1）进入一个 while 循环，不断询问用户要进行训练的模型和数据类型。

（2）输出训练模型的选项菜单，等待用户输入。

（3）如果用户输入的是 6，即退出选项，跳出循环。

（4）如果用户输入的是 1，即选择 KNN 模型，则输出数据类型的选项菜单。

（5）如果用户输入的是 1 到 5 的整数，通过 function_knn_dic 字典调用相应的训练函数。

（6）如果用户输入的不是 1 到 5 的整数，跳出循环

（7）如果用户输入的是 2，即选择逻辑回归模型，则输出数据类型的选项菜单。

（8）如果用户输入的是 1 到 5 的整数，通过 function_logistic_dic 字典调用相应的训练函数。

（9）如果用户输入的不是 1 到 5 的整数，跳出循环。

（10）如果用户输入的不是 1、2 或 3，跳出循环。

图 4-4 为 train 函数流程图。

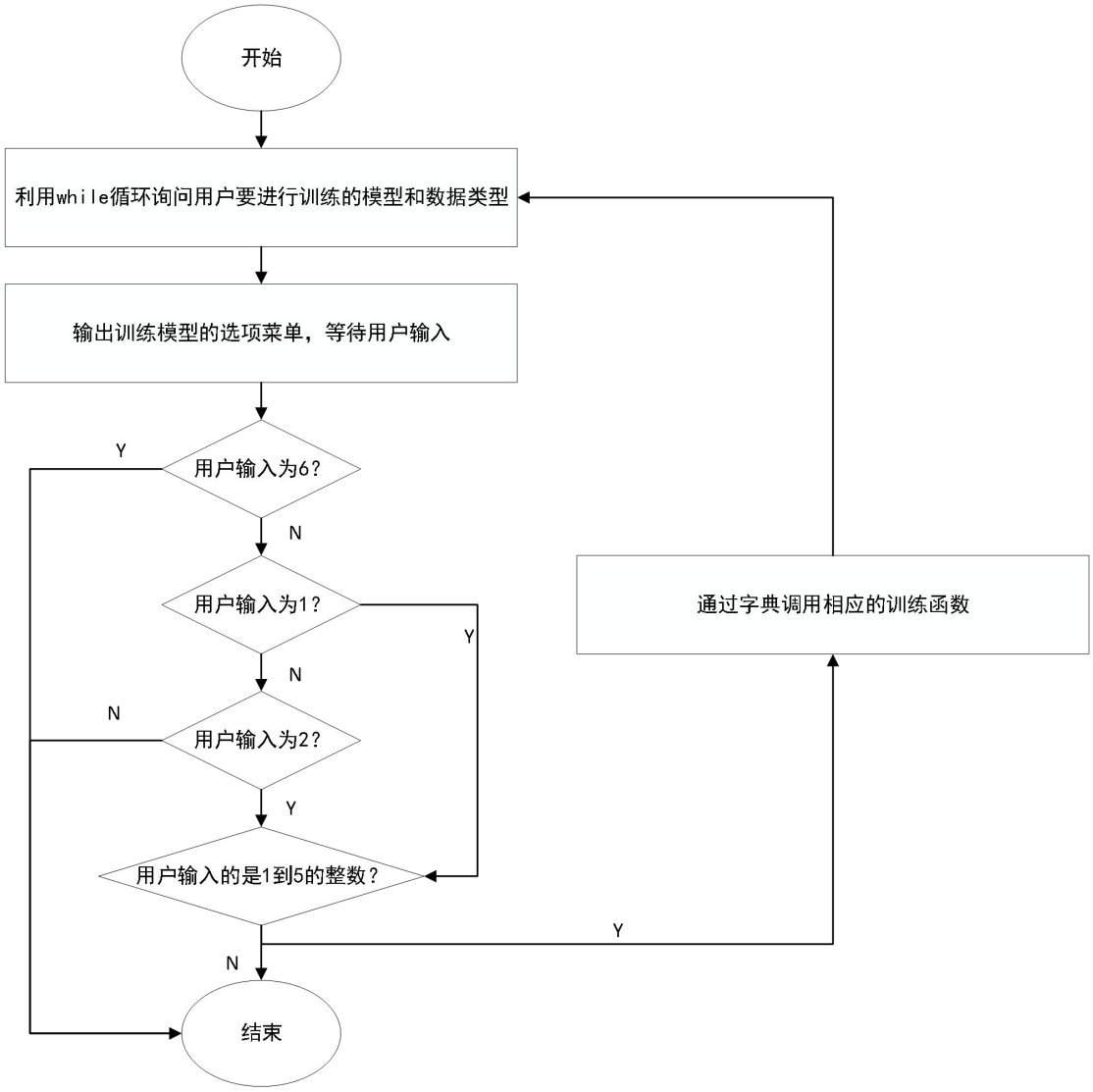

图 4-4 train 函数流程图

4.3.2 test 模块介绍

test 模块定义了一些函数，用于读取、处理和检测不同类型的网络攻击流量数据。主要包括以下函数：

（1）gen_data 函数：从文件中读取测试数据，将其分成特征矩阵和目标向量两部分，并返回这两个部分。

（2）gen_data_crlf 函数、gen_data_xss 函数、gen_data_normal 函数：这三个函数与 gen_data 函数类似，但是分别用于不同类型的网络攻击流量数据。

（3）Normal 函数、Sql 函数、Xss 函数、Traversal 函数、Crlf 函数：这些函数用于对不同类型的网络攻击流量进行检测，并返回检测结果。

（4）choose_model 函数：该函数用于选择要进行哪种类型的攻击检测，可以

选择 KNN 模型或逻辑回归模型。

（5）switch 字典：该字典将不同的攻击类型与相应的检测函数对应起来。

（6）test 函数：该函数是整个程序的入口函数，通过调用其他函数来实现不同类型的攻击检测。

test 模块中函数之间的调用关系流程图如图 4-5 所示。

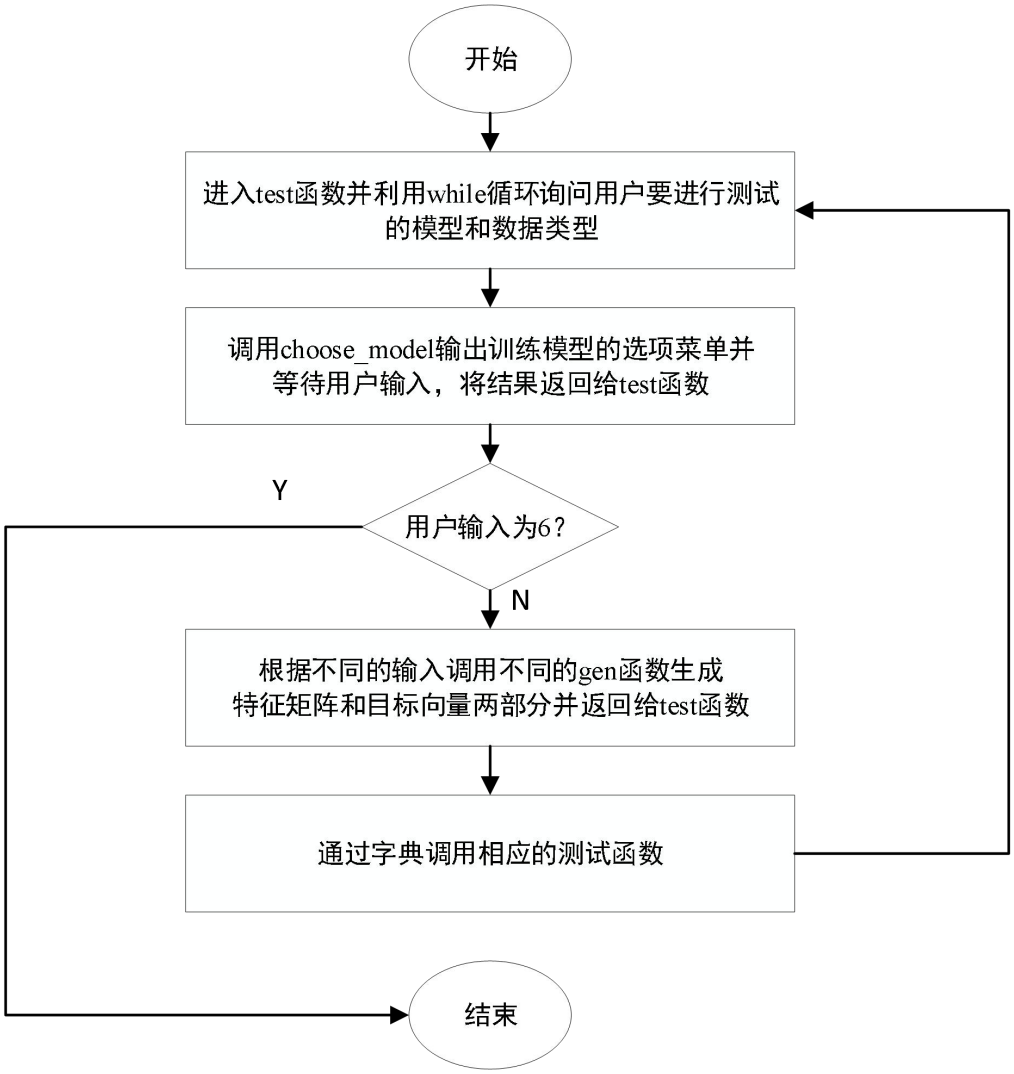

图 4-5 test 模块之间函数调用流程图

4.3.3 user_test 模块介绍

user_test 函数接受两个参数 signal1 和 signal2。函数的主要任务是在一个无限循环中，根据用户的输入执行不同的操作。

在每次循环迭代时，程序会在输出窗口中打印一些信息，包括选择进行单个 URL 流量检测或 csv 文件流量检测，以及退出程序。如果用户选择进行单个 URL 流量检测，则程序会要求用户输入要检测的 URL，然后使用两个模型对该 URL 进

行分类，最后将结果打印到输出窗口中。如果结果是不确定的，程序会要求用户确认该结果是否正确，并将输入的 URL 添加到训练集中。

如果用户选择进行 csv 文件流量检测，则程序会要求用户输入要检测的 CSV 文件路径，并对该文件中的每个 URL 进行分类，最后将结果打印到输出窗口中。

如果用户选择退出程序，则程序将跳出循环并结束执行。user_test 函数逻辑流程如下：

（1）接收两个参数 signal1，signal2 并进入一个 while 无限循环。

（2）输出菜单选项到 control.Menu.output2 中

（3）从输入队列 input_quene 中获取用户选择

（4）如果用户选择了 1，则输出要检测的 URL 的提示信息到 control.Menu.output2 中，并从输入队列中获取用户输入的 URL

（5）调用 url_knn_test 和 url_logistic_test 函数对 URL 进行检测，获取检测结果

（6）如果检测结果为 -1，则输出该流量不属于五种中的任何一种的提示信息到 control.Menu.output1 中，否则根据检测结果输出该流量属于哪种恶意流量的提示信息到 control.Menu.output1 中

（7）输出系统反馈调查表的提示信息到 control.Menu.output2 中，并从输入队列中获取用户反馈

（8）如果用户反馈为 1 且检测结果为准确，则将该 URL 写入对应的训练集文件中

（9）如果用户选择了 2，则输出要检测的 CSV 文件路径的提示信息到 control.Menu.output2 中，并从输入队列中获取用户输入的路径

（10）分别调用 csv_knn_test 和 csv_logistic_test 函数对 CSV 文件进行检测

（11）如果用户选择了 3，则跳出循环

（12）如果用户选择了无效选项，则继续循环

其中用来进行 CSV 数据集检测的函数 csv_test 是一个用于进行基于逻辑回归模型的恶意 URL 检测的 Python 函数。函数首先使用生成函数生成测试数据，并使用 joblib 库加载已经训练好的逻辑回归模型。然后将特征矩阵从 CSV 文件中读取到 numpy 数组中，然后删除最后一列。然后预测特征数据的结果，并统计结果的数量，最后生成恶意 URL 检测报告。此函数适用于检测不同类型的恶意 URL，如

SQL 注入攻击、XSS 注入攻击、遍历攻击、CRLF 注入攻击等。

用来进行单个 URL 检测的函数 URL_test，这个函数的目的是通过逻辑回归模型检测一个 URL 的流量类型。它接受三个参数：URL、signal1 和 signal2（信号传递器，用于在 GUI 界面中输出检测结果）。函数内部使用了五个逻辑回归模型来对 URL 进行分类，这些模型分别用于检测正常流量、SQL 注入攻击流量、XSS 攻击流量、目录遍历攻击流量和 CRLF 注入攻击流量。在执行这些分类预测之前，函数通过调用其他函数来生成与 URL 相关的特征。预测结果将以 1 或 0 的形式输出，其中 1 表示预测为该类型的流量，0 表示预测不是该类型的流量。最后，函数会根据预测结果在 GUI 界面中输出相应的分类报告，如果预测结果不属于上述五种攻击类型之一，则会输出一个怀疑信息。函数将返回一个整数，它表示 URL 流量的类型编号，如果无法确定 URL 的流量类型，则返回-1。

4.4 GUI 的设计

在现代软件开发中，图形用户界面(GUI)的使用越来越普遍，因为它们可以提高用户交互性和用户体验。Qt Creator 是一种流行的跨平台 GUI 开发工具，可以用于创建各种 GUI 应用程序。同时，Python 语言是一种流行的编程语言，拥有丰富的库和工具，可以方便地创建 GUI 应用程序。在本文中使用 Qt Creator 和 Python 创建 GUI 应用程序。

4.4.1 利用 Qt creator 设计用户界面

系统使用 qt creator 中的 QgroupBox，Qframe，QpushBotton 等插件搭建了一个简易的用户界面，如图 4-6 所示：

图 4-6 qt creator 搭建的用户界面

因为系统有三个功能，所以使用了三个按钮来实现不同功能的显示，点击不同的功能选择按钮可以进行功能页面的切换，用户选择了对应功能的选择后点击确认按钮就可以开始运行系统。检测报告生成区用来显示用户进行测试的检测报告，下面的清除按钮可以将检测报告生成区中所有的内容清除。

因为 qt creator 创建的文件是 ui 文件，为了能利用 python 进行下一步的设计，这里使用 PyUIC 将其转换为 Python 文件。首先利用 cd 命令切换到 ui 文件所在文件夹，然后利用命令 pyuic GUI.ui -o GUI.py 即可得到 ui 文件对应的 python 文件。

4.4.2 利用多线程实现 GUI

系统将 GUI 界面的运行作为主线程，系统菜单程序的运行作为从线程。将系统菜单程序 Menu 定义为一个单独的类，并继承 QThread 父类，在主线程中使用 self.print_thread = Menu()开启系统菜单程序从线程。

主线程和从线程信息的传递使用的是 Python 标准库中的一个线程安全的队列 Queue，当从线程中需要读入用户输入数据时使用 get 函数请求队列中的数据，若此时队列中没有数据则等待；当主线程接收到用户的输入时，会使用 put 函数将数据送入到队列中。

将系统菜单中的信息输出到文本框中则使用了信号量来实现，在从线程中使

用 pyqtSignal(str)定义了两个信号量分别对应两个不同的输出文本框，系统菜单程序会把输出的内容都赋值给变量 output，而需要输出到文本框中时则调用 emit 函数发送信号给主线程，主线程中因为使用了 connect 函数将信号量与文本框连接起来，所以在收到从线程的信号量时会把 output 中的内容全部输出到对应的文本框中。

5 测试与结果分析

本章将对测试所用的环境和软件进行一个简单的说明，并对测试的所有结果进行一个分析和评估，并分析产生这个结果的原因。

5.1 测试环境介绍

本系统是采用的 python 编程技术，在 Window10 操作系统环境下，采用 pycharm 开发工具和 Qt creator 软件进行设计与开发的恶意流量检测分类系统。

5.1.1 python 编程技术

Python 是一种高级的、解释型的、面向对象的编程语言，具有简单易学、可读性强、功能丰富的特点，被广泛应用于 Web 开发、数据科学、人工智能、自动化测试等领域。Python 语言的核心哲学是“优雅、明确、简单”，因此它被称为“胶水语言”和“最好的初学者语言”，并且在近年来逐渐成为编程教育和编程入门的首选语言之一。

系统采用的 python 版本为 python 3.9.0，使用的 pycharm 版本为 2020.2.3。

5.1.2 Qt creator 软件

Qt Creator 是一款跨平台的集成开发环境（IDE），专门用于开发基于 Qt 框架的应用程序。Qt Creator 集成了代码编辑器、调试器、可视化设计器、版本控制工具等多个工具，支持多种编程语言，包括 C++、QML、JavaScript 等。Qt Creator 提供了丰富的功能和工具，如自动代码补全、代码高亮、语法检查、图形化调试、版本控制集成等，可以大大提高 Qt 应用程序的开发效率和质量。

系统采用的 Qt creator 版本为 QT creator 10.0.0 和 QT 6.4.2。

5.2 系统测试

下面将从系统的三大功能分别进行测试：

（1）训练功能

图 5-1 为选择正常流量 KNN 模型进行训练功能的 GUI 测试图。

图 5-1 正常流量 KNN 模型进行训练功能

首先点击开始训练数据集并测试的按钮选择系统功能，在模型训练选择中选择 KNN 模型，在数据类型选择中选择正常流量，点击确认按钮开始进行检测，最终生成一个检测报告。检测报告如图 5-2 所示。

图 5-2 正常流量 KNN 模型训练检测报告

从检测报告中可以看出，使用 KNN 模型进行正常流量进行训练时，一共有 10949 条数据用于训练，其中的 30%数据用来模型训练完成后的测试，其中准确度达到了 99.9%，召回率达到了 99.7%，F1 值达到了 99.8%。

（2）本地数据集测试功能

图 5-3 为利用逻辑回归模型对 SQL 注入攻击本地测试集进行攻击的 GUI 测试图。

图 5-3 逻辑回归模型 SQL 注入攻击测试图

首先点击开始使用本地数据集进行测试的按钮选择系统功能，在模型训练选择中选择逻辑回归模型，在数据类型选择中选择 SQL 注入攻击，点击确认按钮开始进行检测，最终生成一个检测报告。检测报告如图 5-4 所示。

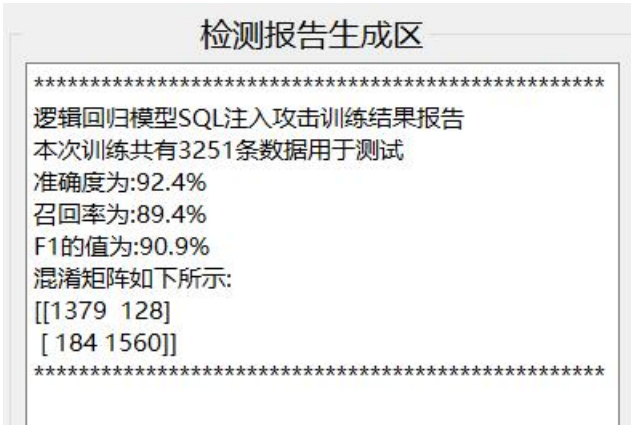

图 5-4 逻辑回归模型 SQL 注入攻击检测报告

从检测报告中可以看出，使用 KNN 模型进行 SQL 注入攻击流量进行训练时，一共有 1439 条数据用于训练，其中准确度达到了 100%，召回率达到了 92.5%，F1 值达到了 96.1%。

（3）用户数据进行测试

图 5-5 为用户输入单个 URL 进行测试的 GUI 测试图。

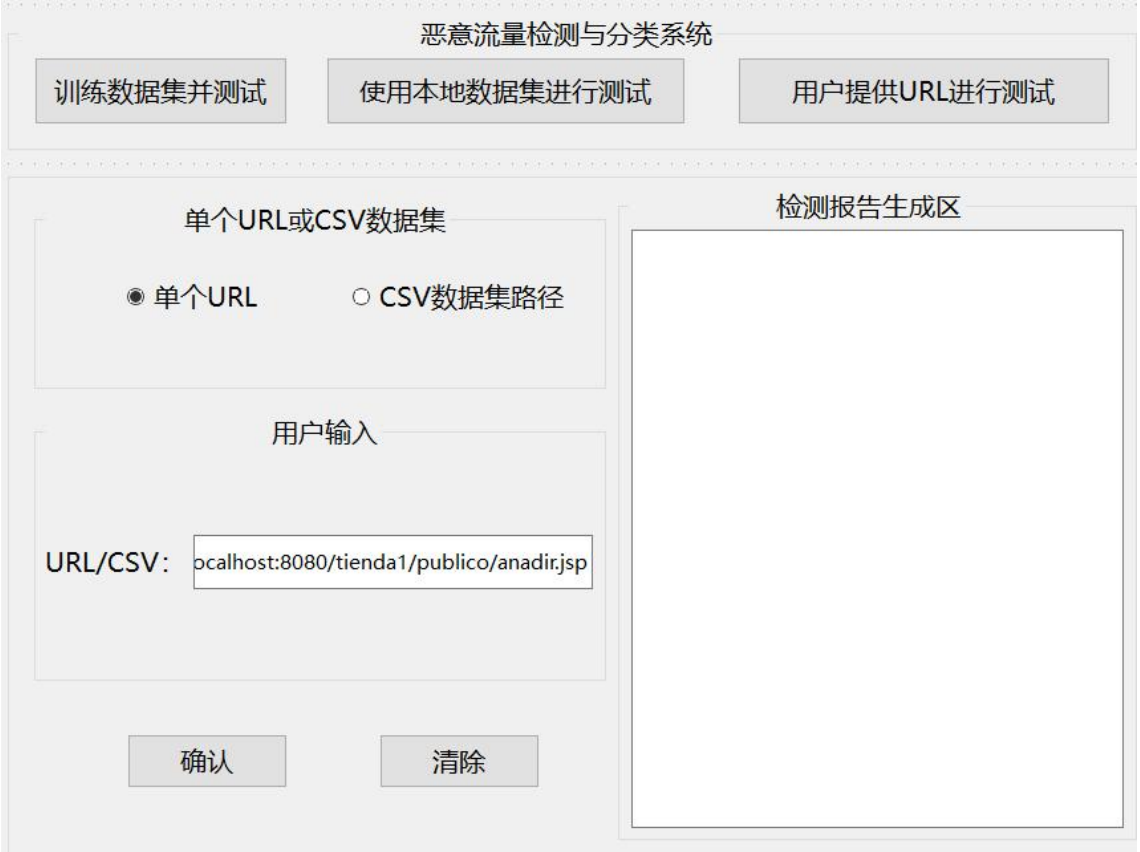

图 5-5 用户测试单个 URL

首先点击开始用户提供 URL 进行测试的按钮选择系统功能，然后选择单个 URL，在输入框中输入待检测的 URL，点击确认按钮开始进行检测，最终生成一个检测报告。检测报告如图 5-6 所示。

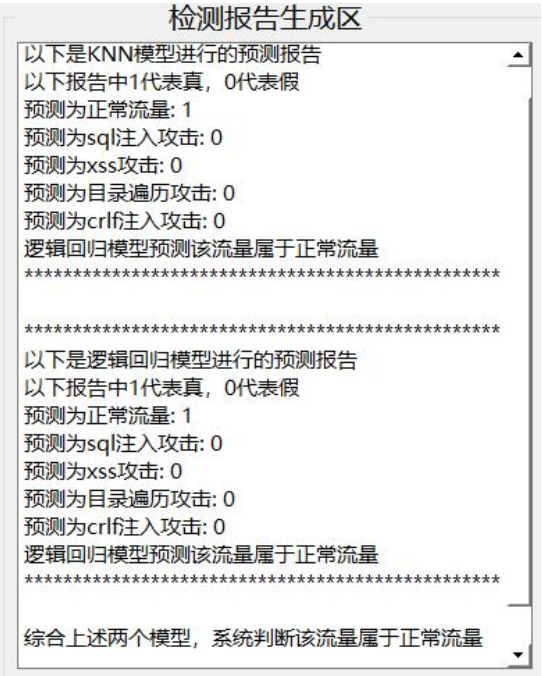

图 5-6 用户测试单个 URL 测试报告

从检测报告中可以看出，当用户输入单个 URL 进行测试时，两个模型分别给出了自己的检测数据，KNN 模型检测中只有正常流量检测为正例，其他种类流量检测都为负例，因此 KNN 模型判断此 URL 为正常流量；逻辑回归判断和 KNN 模型相同，因此系统综合两个模型判断该 URL 为正常流量。

（4）用户提供 CSV 数据集进行检测、

图 5-7 为用户输入 CSV 数据集进行测试的 GUI 测试图。

图 5-7 用户提供 CSV 数据集

首先点击开始用户提供 URL 进行测试的按钮选择系统功能，然后选择单个 URL，在输入框中输入待检测的 URL，点击确认按钮开始进行检测，最终生成一个检测报告。检测报告如图 5-8 所示。

图 5-8 用户提供 CSV 数据集检测结果

从检测报告中可以看出，当用户输入 CSV 数据集进行测试时，两个模型分别给出了自己的检测数据，两个模型对 CSV 数据集中的每个数据进行了预测分类，给出了不同的流量数量，当且仅当五个模型中只有一个预测为正例且其他四个预测为负例时将该数据预测为正例种类，否则将其预测为其他种类恶意流量。

5.3 测试结果分析和比较

最终的测试结果采用准确率、召回率、F1 值和混淆矩阵来表示。

5.3.1 测试结果介绍

（1）精确度

精确度（Precision）是指分类器预测为正例的样本中，实际为正例的样本数占比，也称为查准率。它是分类模型性能评估中的一个重要指标之一，特别是在样本不平衡（imbalanced）的情况下更加重要。

精确度的计算公式为：

$$Precision = TP / (TP + FP) \quad (5\text{-}1)$$

其中，TP 表示真正例（True Positive），即实际为正例且被预测为正例的样本数；FP 表示假正例（False Positive），即实际为负例但被预测为正例的样本数。

（2）召回率

召回率（Recall）是指实际为正例的样本中，被分类器预测为正例的样本数占比，也称为查全率。它是分类模型性能评估中的一个重要指标之一，特别是在需要尽量避免漏检的场景中更加重要。

召回率的计算公式为：

$$Recall = TP / (TP + FN) \quad (5\text{-}2)$$

其中，TP 表示真正例（True Positive），即实际为正例且被预测为正例的样本数；FN 表示假反例（False Negative），即实际为正例但被预测为负例的样本数。

（3）F1 值

F1 值是机器学习中综合评价分类器性能的指标之一，它综合考虑了分类器的精确度和召回率，是精确度和召回率的调和平均数。

F1 值的计算公式为：

$$F1 = 2 * (Precision * Recall) / (Precision + Recall) \quad (5\text{-}3)$$

其中，Precision 是分类器的精确度，Recall 是分类器的召回率。F1 值的范围在 0 到 1 之间，越接近 1 说明分类器的性能越好。与精确度和召回率不同的是，F1 值更加稳健，能够在处理类别不平衡的情况下更好地评估分类器的性能。

（4）混淆矩阵

混淆矩阵（Confusion Matrix）是用来评估分类模型性能的一种表格形式的工具。它可以将模型的分类结果与真实标签进行比较，展示模型在分类上的表现。

混淆矩阵的表格形式如下：

表 5-1 混淆矩阵的表格形式

	实际类别为正例	实际类别为负例
预测类别为正例	True Positive	False Positive
预测类别为负例	False Negative	True Negative

其中，True Positive 表示真正例，即实际为正例且被预测为正例的样本数；False Positive 表示假正例，即实际为负例但被预测为正例的样本数；False Negative 表示

假反例，即实际为正例但被预测为负例的样本数；True Negative 表示真负例，即实际为负例且被预测为负例的样本数。

5.3.2 测试结果展示

（1）训练功能测试结果

表 5-2 KNN 算法训练功能测试结果

KNN 算法										
	正常流量		SQL 注入		XSS 攻击		目录遍历攻击		CRLF 注入攻击	
数据量	36497		10837		10387		10240		4180	
精确度	99.8%		93.4%		99.7%		99.3%		97.7%	
召回率	99.8%		96.4%		98.7%		99.8%		100%	
F1 值	99.8%		94.9%		99.2%		99.5%		98.9%	
混淆矩阵 (30%数据用于测试)	5382	7	1378	120	1609	5	1541	11	632	14
	15	5545	63	1690	19	1483	3	1520	0	608

从表 5-2 中可以看出，当使用 KNN 模型进行训练时，不管是五种流量中的哪一种，精确度，召回率，F1 值都在 90%以上，并且正常流量、XSS 攻击流量、目录遍历攻击流量和 CRLF 注入攻击流量这四种类型的测试结果都十分出色，到达了 97%以上。

表 5-3 逻辑回归算法训练功能测试结果

逻辑回归算法										
	正常流量		SQL 注入		XSS 攻击		目录遍历攻击		CRLF 注入攻击	
数据量	36497		10837		10387		10240		4180	
精确度	99.0%		92.4%		99.6%		99.4%		96.8%	
召回率	99.6%		89.4%		93.9%		99.8%		99.5%	
F1 值	99.3%		90.9%		96.7%		99.6%		98.1%	
混淆矩阵 (30%数据用于测试)	5274	57	1379	128	1616	6	1551	9	631	20
	22	5596	184	1560	91	1403	2	1512	3	600

从表 5-3 中可以看出，当使用逻辑回归模型进行训练时，除了 SQL 注入攻击的召回率只有 89.4%，其他的精确度，召回率，F1 值都在 90%以上，虽然整体测试结果数据都比较出色，但较之前的 KNN 模型，逻辑回归模型的测试数据各个方面都略低于 KNN 模型。

（2）本地 csv 数据集测试结果

表 5-4 KNN 算法本地 csv 数据集测试结果

KNN 算法										
	正常流量		SQL 注入		XSS 攻击		目录遍历攻击		CRLF 注入攻击	
数据量	3991		1439		1250		1062		2081	
精确度	99.7%		99.5%		99.7%		100%		100%	
召回率	98.1%		92.5%		97.3%		100%		100%	
F1 值	98.9%		96.0%		98.5%		100%		100%	
混淆矩阵	0	0	0	0	0	0	0	0	0	0
	77	3914	108	1331	34	1216	0	1062	0	2081

从表 5-4 中可以看出，当使用 KNN 模型进行测试时，不管是五种流量中的哪一种，精确度，召回率，F1 值都在 90%以上，并且目录遍历攻击流量和 CRLF 注入攻击流量这两种类型的精确度、召回率和 F1 值都到达了 100%。

表 5-5 逻辑回归算法本次 csv 数据集测试结果

逻辑回归算法										
	正常流量		SQL 注入		XSS 攻击		目录遍历攻击		CRLF 注入攻击	
数据量	3991		1439		1250		1062		2081	
精确度	99.6%		99.3%		99.6%		100%		100%	
召回率	98.0%		95.1%		95.4%		98.7%		98.9%	
F1 值	98.7%		97.2%		97.5%		99.3%		99.5%	
混淆矩阵	0	0	0	0	0	0	0	0	0	0
	79	3912	70	1369	58	1192	14	1048	22	2059

从表 5-5 中可以看出，当使用 KNN 模型进行测试时，不管是五种流量中的哪一种，精确度，召回率，F1 值都在 90%以上，虽然总体的测试结果都在 95%以上，甚至有个别的测试结果达到了 100%，但和 KNN 模型相比，逻辑回归模型的测试数据各个方面都略低于 KNN 模型。

（3）用户提供 csc 测试集测试结果

表 5-6 用户提供 csv 测试集测试结果

	KNN 算法	逻辑回归算法
正常流量	5216 (29.0%)	3760(20.9%)
SQL 注入	1887 (10.5%)	3353 (18.6%)
XSS 攻击	4136 (23.0%)	4010 (22.3%)
目录遍历攻击	5081 (28.2%)	5104 (28.4%)
CRLF 注入攻击	1650 (9.2%)	1745 (9.7%)
其他种类恶意攻击	21 (0.1%)	19 (0.1%)
总数	17991	17991

从表 5-6 可以看出，当使用 KNN 模型对用户提供的 CSV 数据集进行测试时，KNN 模型和逻辑回归模型在 XSS 攻击流量、目录遍历攻击流量、CRLF 注入攻击流量和其他种类恶意攻击流量的预测上十分相似，但是在对正常流量和 SQL 注入攻击流量的预测上差别比较明显。

5.3.3 测试结果评估

从上述测试数据中可以发现，KNN 算法的精确率，召回率和 F1 的值总体上都比逻辑回归算法高。

一般来说，KNN 算法在处理非线性可分的数据集时，比逻辑回归算法表现更好。这是因为 KNN 算法基于距离度量对数据点进行分类，更加适合非线性边界的情况。而逻辑回归算法则基于对数几率函数对数据进行建模，更加适合线性可分的数据集。

如果在测试集上，KNN 算法的分类性能比逻辑回归算法更好，可能是因为测试集的分布与训练集的分布相似，同时测试集中的样本具有一定的非线性特征。在这种情况下，KNN 算法能够更好地捕捉到数据的非线性特征，从而提高了分类的准确率。

在本文选取的数据集中测试集和训练集的分布十分相似，导致 KNN 算法中不同类别的数据点之间距离比较相近，导致 KNN 算法的测试结果比逻辑回归算法更好，测试的结果和预期相符。

6 总结

本文旨在研究和实现基于机器学习的恶意流量检测系统。为了达到这一目标，首先介绍了恶意攻击的种类以及机器学习的基本概念和技术，包括监督学习、非监督学习和半监督学习等方法，并分析了这些方法在恶意流量检测中的应用。

其次，详细介绍了本文所采用的数据集和特征选择方法，并详细介绍了特征提取的每一种分量以及进行特征提取的方法。在特征选择的基础上，采用了多种机器学习算法进行模型训练和测试，包括 KNN 算法和逻辑回归算法，通过对比测试结果，分析测试结果和产生该测试结果的原因。

最后，实现了一个基于机器学习的恶意流量检测系统，并通过真实网络流量数据集的测试，证明了该系统在恶意流量检测方面的有效性和高效性。

本文的主要创新点在于采用了多种机器学习算法进行模型训练和测试，并通过多个二分类的机器学习模型实现了一个多分类的基于机器学习的恶意流量检测系统，该系统具有实用性和可操作性；并且设计了一个简易的 GUI 界面方便进行操作。

该系统的不足主要有以下几个方面：

（1）特征选择：特征的选择可能存在问题，选择的特征很难与恶意流量的全部特征相匹配，无法很好地识别恶意流量。

（2）实时性：本系统是对静态的数据集进行检测，并不能实时的针对网络流量进行检测。

（3）模型的泛化能力：模型在训练和测试过程中对于数据的依赖程度较高，模型的泛化能力不足，可能会导致模型在新的场景下表现不佳。

未来，该系统可以通过以下方面进行改进和展望：

（1）加强数据预处理：通过对数据进行预处理和调整，平衡正负样本比例，提高恶意流量分类的准确率。

（2）优化特征选择：采用更加科学和有效的特征选择方法，更好地捕捉恶意流量的特征。

（3）提高实时性：优化算法和系统架构，实现实时网络流量的检测功能。

（4）改进算法：研究和探索新的机器学习算法，提高模型的泛化能力和检测准确率。

综上所述，本文针对当前互联网安全领域的热点问题进行了深入研究，提出

了基于机器学习的恶意流量检测系统，并通过测试验证了该系统的有效性和高效性。未来的研究可以进一步扩大数据集规模和种类，并采用更加先进的机器学习算法，以提高恶意流量检测的精度和效率，实现更好的检测系统。

致谢

在我完成本科毕业论文之际，我深感到无比的荣幸和感激。在此，我要向所有关心和支持我的人致以最诚挚的谢意。

首先，我要感谢我的导 [redacted] 师，感谢您在整个毕业设计过程中的悉心指导和支持，是您不断给予我鼓励和帮助，让我在学术上不断成长。您的严谨治学精神和对知识的热爱，让我深受启发，对我的毕业论文起到了至关重要的作用。

在此，我要特别感 [redacted] 我毕业论文的大力支持和合作。此课题作为一项[redacted]的[redacted] 题，为我提供了宝贵的实践机会和研究资源。同时，我还要向企业导 [redacted] 师表示由衷的感谢。老师在整个研究过程中给予了我专业的指导和宝贵的建议，对我的研究起到了重要的推动作用。

同时，我也要感谢所有帮助我完成论文的老师和同学们，感谢您们提供的帮助、建议和支持，让我能够顺利地完成这篇毕业论文。在这段时间里，我也认识了很多优秀的同学和老师们，你们的精神风貌和对学术事业的热情深深地感染了我。

此外，我还要感谢我的家人和朋友们，感谢你们一直以来的鼓励和支持，让我在学业上没有后顾之忧，让我能够专注于我的毕业论文。

最后，我要向母 [redacted] 以最崇高的敬意和感激之情，感谢学校为我们提供的良好学习环境和学术氛围，让我能够在这里度过人生中宝贵的学习时光。

谨以此文，向所有关心和支持我的人致以最真诚的感谢和祝福。谢谢大家！

参考文献

[1] CNNIC 发布第 51 次《中国互联网络发展状况统计报告》[J].互联网天地,2023(03):3.

[2] 张蕾,崔勇,刘静,江勇,吴建平.机器学习在网络空间安全研究中的应用[J].计算机学报,2018,41(09):1943-1975.

[3] 林楚强.基于机器学习的恶意行为流量特征分类[J].自动化与仪器仪表,2022(06):7-12.DOI:10.14016/j.cnki.1001-9227.2022.06.007.

[4] 张晓双,徐依凌,刘渊.基于 Web 应用的网络安全漏洞发现与研究[J].网络与信息安全学报,2016,2(06):58-65.

[5] Shafiq, Muhammad, et al. "CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine-learning techniques." *IEEE Internet of Things Journal* 8.5 (2020): 3242-3254.

[6] 张诚,叶红,吕博良,程佩哲,周京.基于机器学习的隐蔽隧道恶意加密流量检测识别研究[J].中国金融电脑,2021(10):59-64.

[7] 郭益民. 基于深度学习的 Android 恶意应用网络流量检测方法[D].上海交通大学,2020.DOI:10.27307/d.cnki.gsjtu.2020.001944.

[8] 曾勇,吴正远,董丽华,刘志宏,马建峰,李赞.加密流量中的恶意流量识别技术[J].西安电子科技大学学报,2021,48(03):170-187.DOI:10.19665/j.issn1001-2400.2021.03.022.

[9] 骆子铭,许书彬,刘晓东.基于机器学习的 TLS 恶意加密流量检测方案[J].网络与信息安全学报,2020,6(01):77-83.

[10] 张稣荣. 基于深度学习的加密流量分类技术研究[D].战略支援部队信息工程大学,2022.DOI:10.27188/d.cnki.gzjxu.2022.000025.

[11] Malhotra H ,Sharma P . Intrusion Detection using Machine Learning and Feature Selection[J]. International Journal of Computer Network and Information Security, 2019.

[12] Sakr M M , Tawfeeq M A , El-Sisi A B . Network Intrusion Detection System based PSO-SVM for Cloud Computing[J]. International Journal of Computer Network and Information Security, 2019, 11(3):22-29.

[13] Denning D E . An Intrusion-Detection Model[C]// An Intrusion-Detection M

odel. IEEE Computer Society, 1986.

[14] Li L , Yu Y , Bai S , et al. Towards Effective Network Intrusion Detecti on: A Hybrid Model Integrating Gini Index and GBDT with PSO[J]. Journal of Sensors,2018,(2018-3-26), 2018, 2018:1-9.

[15] 夏聃, Omry, Haizler. 美国网络战的历史及其对现代网络作战组织和决策的影响[J]. 中国信息安全, 2017(4):7.

[16] Smitha R , Hareesha K S , Poornima P K . A Machine Learning Approa ch for Web Intrusion Detection:MAMLS Perspective. 2018.

[17] H Takcı, I Soğukpınar. Centroid-Based Language Identification Using Letter Feature Set[J]. DBLP, 2004.

[18] Zhang C , Ju J , Kamel M . Intrusion detection using hierarchical neural networks[J]. Pattern Recognition Letters, 2005, 26(6):779-791.

[19] Cho S , Cha S . SAD: Web session anomaly detection based on paramete r estimation[J]. Computers & Security, 2004, 23(4):312-319.

[20] 高国柱, 吴海燕. Web 应用安全监测系统设计与应用[J]. 计算机工程与设计, 2010(17):4.

[21] Suhaimi H, Suliman SI, Musirin I, et al. Network intrusion detection syste m by using genetic algorithm[J]. Indonesian Journal of Electrical Engineering and Computer Science, 2019, 16(3):1593.

[22] Wang, Zihao, Kar Wai Fok, and Vrizlynn LL Thing. "Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative st udy." *Computers & Security* 113 (2022): 102542.

[23] 唐政治,曾学文,陈君,郭志川.基于机器学习的网络流量分析综述[J].网络新媒体技术,2020,9(05):1-8.

[24] 鲁刚,郭荣华,周颖,王军.恶意流量特征提取综述[J].信息网络安全,2018(09):1-9.

[25] 王世伟.论信息安全、网络安全、网络空间安全[J].中国图书馆学报,2015,41(02):72-84.DOI:10.13530/j.cnki.jlis.150009.

[26] 彭珺,高珺.计算机网络信息安全及防护策略研究[J].计算机与数字工程,2011,39(01):121-124+178.

[27] 沈昌祥,张焕国,冯登国,曹珍富,黄继武.信息安全综述[J].中国科学(E 辑:信息科

学),2007(02):129-150.

[28] 彭沙沙,张红梅,卞东亮.计算机网络安全分析研究[J].现代电子技术,2012,35(04):109-112+116.DOI:10.16652/j.issn.1004-373x.2012.04.053.

[29] 蒋建春,马恒太,任党恩,卿斯汉.网络安全入侵检测:研究综述[J].软件学报,2000(11):1460-1466.DOI:10.13328/j.cnki.jos.2000.11.005.

[30] 谭秦红,田应信.基于人工智能的通信网络入侵检测系统设计[J].长江信息通信,2022,35(12):189-191.

[31] 吴凯林. 基于小样本深度学习的恶意流量识别研究与实现[D].南京邮电大学,2021.DOI:10.27251/d.cnki.gnjdc.2021.001639.

[32] 蒋磊. 基于机器学习的 SQL 注入检测技术研究[D].南京邮电大学,2017.

[33] 徐寅昊. SQL 注入及 SQL Server 的安全性研究[D].华东师范大学,2009.

[34] 李玲,任佳宁,韩冰倩,朱萍.基于 Web 应用安全的 SQL 注入漏洞与防御[J].电脑编程技巧与维护,2022(01):175-176.DOI:10.16184/j.cnki.comprg.2022.01.049.

[35] 罗超超. 基于深度学习的 SQL 注入和 XSS 攻击检测技术研究[D].中国工程物理研究院,2020.DOI:10.27498/d.cnki.gzgwy.2020.000096.

[36] 杨朋朋. 基于机器学习的 URL 攻击行为实时检测技术的研究[D].北京邮电大学,2018.

[37] 陈胜, 朱国胜, 祁小云,等. 基于机器学习的网络异常流量检测研究[J]. 信息通信, 2017(12):4.

[38] 聂盼盼, 臧洌, 刘雷雷. 基于对支持向量机的多类分类算法在入侵检测中的应用[J]. 计算机应用, 2013, 33(02):426-429.

[39] 杨剑锋,乔佩蕊,李永梅,王宁.机器学习分类问题及算法研究综述[J].统计与决策,2019,35(06):36-40.DOI:10.13546/j.cnki.tjyjc.2019.06.008.

[40] Muniyandi A P , Rajeswari R , Rajaram R . Network Anomaly Detection by Cascading K-Means Clustering and C4.5 Decision Tree algorithm[J]. Procedia Engineering, 2012, 30:174-182.