



中北大学
NORTH UNIVERSITY OF CHINA

毕业设计说明书

网络流量分析与可视化工具 的设计与实现

学生姓名：_____ 学号：_____

学 院：_____ 软件学院

专 业：_____ 软件工程

指导教师：_____

20** 年 06 月

基于 Python 开发网络流量分析与可视化工具设计与实现

摘要

随着信息技术的飞速发展，网络流量呈现爆炸式增长，网络管理和安全防护面临日益严峻的挑战。在此背景下，基于 Python 开发网络流量分析与可视化工具具有重要的现实意义。本研究旨在设计并实现一款高效的网络流量分析与可视化工具，通过对网络流量的深入分析和直观展示，助力网络管理员优化网络资源分配、及时发现网络异常，从而提升网络管理效率与安全防护能力。该工具采用模块化设计思路，涵盖用户管理、数据管理、网络流量监控及流量分析可视化等核心模块。在实现方法上，充分利用 Python 丰富的库资源，如 pandas 进行数据处理、matplotlib 进行数据可视化、scapy 实现网络流量捕获等。工具功能全面，支持用户注册登录、权限管理、历史流量数据存储查询导出、实时流量捕获、异常流量检测以及多样化的流量数据可视化展示。其优势在于操作简便、可视化效果直观、分析准确高效，为网络管理和安全防护提供了有力的技术支持。

关键词: 网络流量分析, 可视化, 工具设计, 网络安全

Design and Implementation of Network Traffic Analysis and Visualization Tools Based on Python Development

abstract

With the rapid development of information technology, network traffic has shown explosive growth, and network management and security protection are facing increasingly severe challenges. In this context, it is of great practical significance to develop network traffic analysis and visualization tools based on Python. This study aims to design and implement an efficient network traffic analysis and visualization tool. Through in - depth analysis and intuitive display of network traffic, it helps network administrators optimize network resource allocation and detect network anomalies in a timely manner, so as to improve network management efficiency and security protection capabilities. The tool adopts a modular design idea, covering core modules such as user management, data management, network traffic monitoring, and traffic analysis visualization. In terms of implementation methods, it makes full use of Python's rich library resources, such as pandas for data processing, matplotlib for data visualization, and scapy for network traffic capture. The tool has comprehensive functions, supporting user registration and login, permission management, storage, query, and export of historical traffic data, real - time traffic capture, abnormal traffic detection, and diversified visualization display of traffic data. Its advantages lie in simple operation, intuitive visualization effects, and accurate and efficient analysis, providing strong technical support for network management and security protection.

Keyword: Network Traffic Analysis, Visualization, Tool Design, Network Security

目 录

1 引言 1

1.1 课题背景 1

1.2 Python 语言优势 1

1.3 研究目的与意义 1

2 相关技术与理论基础 3

2.1 网络流量分析相关理论 3

2.2 数据可视化技术 3

2.3 基于 Python 的网络流量分析与可视化研究进展 3

3 工具设计 5

3.1 总体架构设计 5

3.2 用户管理模块设计 5

3.2.1 用户注册功能设计 5

3.2.2 用户登录功能设计 6

3.2.3 权限设置功能设计 6

3.3 数据管理模块设计 6

3.3.1 数据库结构设计 6

3.3.2 数据存储策略 8

3.3.3 数据查询与导出功能设计 8

3.4 网络流量监控模块设计 9

3.4.1 实时流量捕获技术 9

3.4.2 异常流量检测算法 9

3.5 流量分析可视化模块设计 10

3.5.1 可视化库选择 10

3.5.2 可视化图表设计 10

4 工具实现 11

4.1 开发环境搭建 11

4.2 用户管理模块实现 11

4.2.2 用户登录功能实现	12
4.2.3 权限设置功能实现	13
4.3 数据管理模块实现	13
4.3.1 数据库操作实现	14
4.3.2 数据存储策略实现	14
4.3.3 数据查询与导出功能实现	14
4.4 网络流量监控模块实现	14
4.4.1 实时流量捕获代码实现	14
4.4.2 异常流量检测算法实现	15
4.5 流量分析可视化模块实现	16
4.5.1 可视化图表绘制代码实现	16
4.5.2 可视化交互功能实现	18
5 工具测试与评估	20
5.1 测试环境搭建	20
5.2 功能测试	20
5.2.1 用户管理功能测试	20
5.2.2 数据管理功能测试	20
5.2.3 网络流量监控功能测试	22
5.2.4 流量分析可视化功能测试	22
5.3 性能测试	24
5.3.1 响应时间测试	24
5.3.2 并发性能测试	24
5.4 评估结果分析	24
6 结论	26
6.1 研究成果总结	26
6.2 经验教训总结	26
6.3 未来工作展望	27
参 考 文 献	28
致 谢	30

1 引言

1.1 课题背景

随着数字化的迅猛发展，网络流量也呈现指数级增长，有研究表明全球网络流量在过去十年中增长了三倍，且仍在不断增长。而网络流量的增长不仅归功于互联网用户的增加，还有物联网、云计算、5G 网络等新技术的推动，网络流量的复杂性和多样性也随之增加，给网络管理和安全防护带来了很大挑战，因此进行网络流量分析并进行可视化是非常有必要的，通过分析网络流量，网络管理员可以及时发现潜在的安全威胁和优化网络资源配置，提高网络性能^[8]，例如通过 python 进行网络流量特征分析可视化工具，帮助管理员了解网络数据概况，从而掌握网络流量的变化规律和网络特性^[1]。而且流量监控系统在城域网设备维护中应用，在端口流量查看不便的前提下，能够实现对流量的直观展示的数字化网络运维监控，有效地提升了工作效率。因此，对网络中的流量进行分析和显示，是现代网络管理的基本工具，也是网络安全运行的基本技术。

1.2 Python 语言优势

Python 是一种高级编程语言，在数据处理时算力强大，库文件多，语法简单明了，是进行网络流量分析与可视化的首选工具之一。Python 库众多，例如：pandas、numpy 等，可以方便地对海量的数据进行清洗、转换、变换与分析，这些库不仅能快速地处理结构化的数据，还能对非结构化的数据进行处理，通过正则表达式等提取分析数据^[3]；Python 网络编程库强大，例如：scapy、socket 等，可以实现抓包功能、网络协议分析、实时流量捕获等功能；Python 的库文件可视化效果也很优秀，例如：matplotlib、seaborn、plotly 等库，可以提供各式各样的高精度图形，展示流量数据趋势、构成特点。更重要的是，Python 的可拓展性比较好，可以跟各式各样的系统进行连接，应用到各种场景^[2]。鉴于此，将 Python 作为系统开发技术，有利于降低系统的开发难度，提高系统的灵活性。

1.3 研究目的与意义

本研究主要是设计和实现一个基于 Python 的网络流量分析与可视化工具，针对目前的网络流量数据分析工具无法让网络管理员准确、实时、高效的分析网络流量数据的问题^[5]，希望实现在线对网络流量数据进行分析，对网络异常的检测以及

网络流量数据的可视化，让网络管理员能够更加准确高效的掌握网络情况，发现网络安全漏洞。通过利用 Python 对网络流量数据的计算和算法分析网络流量数据的异常。利用可视化技术将复杂的网络流量数据图形可视化，便于用户进行可视化分析。从实际应用的角度来说，通过检测网络异常流量，可以有助于优化网络资源和提高网络安全，如异常检测有助于预防网络攻击。另外，此工具的设计与实现，对于工具的研究与发展也具有一定的参考作用，对网络流量分析技术有重要意义^[7]。总之，本课题的研究具有重要的理论意义和实践价值。

2 相关技术与理论基础

2.1 网络流量分析相关理论

网络流量分析是指对网络中的数据包进行抓包分析、捕获及分析，从中提取特征、分析现象，作为网络管理、网络安全防范的重要依据。流量采集方法：采用硬件的流量监测设备，如：探针；采用软件的抓包工具，如：Wireshark，将网络中的数据包捕获并保存为.pcap 文件，便于分析；提取特征方法：特征提取是网络流量分析的关键步骤，根据数据包中包含的头部信息、负载、时间序列等信息进行特征提取。生成能够反映网络特征的统计指标，如：流量速率、协议分布、端口使用情况等；基于流的方法能够将具有相同标识字段的包聚合在一起，便于分析。研究表明，机器和深度学习算法的结合能提高流量表征能力，尤其是在网络环境复杂时具备一定优势^[10]。

2.2 数据可视化技术

数据可视化是指将数据转化为人们可视的图形，其目的主要是为视觉感官服务，使人们更容易理解和学习。数据可视化技术常被用于展示网络流量的分析趋势图、组成部分图和异常图。折线图、饼状图、柱状图、热力图是数据可视化的常用图形，每种图形都有自身的特点以及适应的场景和应用。例如在折线图中通常表示时间序列上的变化曲线，在饼状图中通常表示某种协议或者应用所占比率。Python 中常用的可视化库有 matplotlib 和 seaborn，两个库提供了丰富的图形绘制函数与灵活多变的图形设置，能够满足多种可视化任务。matplotlib 兼容性和可扩展性都很强，功能丰富，可以绘制从简单的散点图，到复杂的三维立体图；seaborn 在 matplotlib 的基础上进一步完善了图形的默认样式，使图形更加漂亮易读^[4]。此外，伴随着交互式可视化的应用逐渐增多起来，D3.js、Plotly 等库也被研究者关注，使得可视化图形具备一些交互性的可视化功能，如鼠标悬浮、缩放等。

2.3 基于 Python 的网络流量分析与可视化研究进展

以 Python 为语言的网络流量分析与可视化研究近几年发展较为迅速，尤其对于工具的设计与算法的改进，国内外的研究者都很好地利用了 python 库的丰富资源(pandas、numpy、scapy 等)进行高效、便捷的流量计算与可视化工作。例如，文献^[2]提出了一种基于 Python 的图形化编程系统，能够自动抓取网络流量特征并

提取后以 csv 形式记录，为后续的机器学习研究提供数据来源。文献^[6]实现了一个集数据收集、清洗、处理、可视化于一体的网络小说数据挖掘平台，体现了 Python 对复杂场景的强大处理能力。但是已有工作也存在一些弊端。第一，部分工具在流量实时检测、异常情况检测方面表现不佳，尤其是在高流量时数据丢失或数据延时情况较为严重。其次，可视化功能大多以静态图像的形式给出，可操作性和可操作性差，用户无法根据个人需求定制^[4]。为了解决这些问题，本研究拟采用先进的流量分析与交互式可视化技术，设计并实现具有完善功能，性能较强的网络流量分析与可视化系统。本研究的创新之处如下：

- （1）结合统计学与机器学习算法，提高异常流量检测的准确性和效率；
- （2）利用 Python 中最新的交互可视化库，提高用户使用体验和分析功能
- （3）优化工具架构，支持大规模流量数据的高效存储和查询。

3 工具设计

3.1 总体架构设计

基于 python 的网络流量分析与可视化分析在整体架构上采用了 Browser/Server(浏览器 / 服务器, 简称 B / S)架构, 以实现高速处理和高效的人机交互, 整个系统由前端、后端和数据库等三部分组成, 通过接口实现三者之间的交互和功能合作。其中, 前端主要是对数据信息的交互、可视化处理及呈现; 后端主要是对业务逻辑处理、数据分析及数据存储; 数据库主要是对历史流量数据和用户信息的数据存储。其中, 前端采用 HTML5、CSS3、JavaScript, 并使用 ReactJS 实现前端的动态化构建, 以实现用户的良好体验和多平台兼容。后端采用 python 开发语言, 以 Django 或 flask 框架进行服务构建, 并通过提供 RESTful 的 API 接口, 来实现前后端分离架构下的前后端数据通信。数据库采用关系数据库(如 MySQL)或非关系数据库(MongoDB), 视数据库的类型与数据的用途, 选择最匹配的类型, 确保并发场景下的读写效率; 同时, 系统利用网络抓包库(Scapy)抓取数据包, 并通过机器学习算法对抓包结果进行特征提取和异常检测, 再以图表的形式(如 Matplotlib、Plotly 等可视化库)将结果返回给用户, 实现流量分析与可视化功能 [1, 10]。

3.2 用户管理模块设计

3.2.1 用户注册功能设计

其中注册用户是工具中的基础功能, 目的是保证用户的完整性和安全性, 在注册时需要收集用户名、密码、邮箱等信息。系统为了保证账户安全, 对用户输入的密码进行了复杂化规定, 例如密码中必须包含字母、数字、特殊符号等信息, 并限制密码长度。邮箱需要通过格式校验, 确保邮箱不包含非法字符以及无效域名等。如果校验通过, 则将用户的真实信息加密保存到数据库, 对于用户的密码采用哈希算法进行加密处理, 例如 SHA-256, 避免用户的敏感信息泄露。此外, 系统还需要采用验证码对用户的注册信息进行验证, 采用图形验证码或短信验证码对用户身份进行校验, 防止用户的恶意注册。注册信息存储在规范化的存储表中, 包含主键 ID、用户名、密码哈希算法、邮箱等字段。其中主键 ID 是对用户进行统一标引和存储管理的唯一标识, 用于后期的维护和数据的查找等操作 [13]。

3.2.2 用户登录功能设计

用户登录功能设计思路是通过登录凭证验证、用户会话管理来实现用户的身份安全认证，在用户登录时需要提交用户名密码，通过比对记录在数据库中的数据来认证登录凭证。可以限制登录失败次数来防止暴力登录破解，也可以在失败后锁定一段时间，在登录成功后生成_token 或 Session 机制来维护用户会话，防止用户操作后用户的身份不合法的情况。Token 机制采用 JWT（JSON Web Token）方式，将用户信息进行加密后随请求发送到服务器，在服务器端解密验证 Token 的有效性，避免频繁查询数据库^[5]。Session 机制是将用户信息存储在服务器端，通过 Cookie 的方式传输 Session ID，在安全性要求较高的情况下使用。无论是哪一种机制，系统都要对传输过程的数据进行加密，例如采用 HTTP 加密，防止中间者登录用户的系统，对传输数据进行加密。所以无论是哪种机制，都需要对传输中的数据加密。

3.2.3 权限设置功能设计

权限设置功能的设计即是指给用户的不同操作以不同权限，通过权限限制用户对工具以及数据的访问级别，保障系统的安全性以及稳定性。本工具分为管理员与普通用户两种用户角色，管理员具有最高权限，可以进行所有操作包括用户管理、数据管理、系统管理等，而普通用户只有部分权限，如查看流量表，导出自己的数据等。通过基于角色的访问控制模型（Role-Based Access Control, RBAC）模型，使用角色与权限的映射来设置用户访问权限，在设计系统数据库时，在角色表和权限表中设置角色和权限，并且通过外键关联用户表和角色表，外键绑定角色。同时在前端界面，按照用户角色对渲染功能的菜单进行显示、隐藏没有授权的界面，降低用户误操作的可能性，有效的避免被误操作，保证工具的安全性与有效性^[13]。

3.3 数据管理模块设计

3.3.1 数据库结构设计

数据库表结构设计是数据管理模块的核心，其目标是建立合理的数据库表来方便存储和查询历史流量数据。本工具中设计流量数据表、用户数据表、日志数据表等，通过外键约束建立表与表之间的关系，来保证数据的一致性。流量表用来存储采集到的流量，包括流量 id、时间戳、源 ip 地址、目的 ip 地址、端口、流量大小等字段，流量 id 为主键用于与各个表建立外键关系^[2]。用户表用来存储用户的

中北大学 2025 届毕业设计说明书

用户名和密码等基本信息，包括用户 ID、用户名、密码哈希、邮箱等字段，用户 ID 为主键用于与各个表建立外键关系。日志表用来存储用户日志和系统日志，方便审计和排错。为了快速地提高搜索查询的速度，系统在表里的每个字段上创建索引，比如流量数据表中的时间戳、用户信息表中的用户名等，来提高搜索查询的速度^[15]，如表 3.1，3.2，3.3 所示。

表 3.1 用户信息表

字段	类型	约束	说明
id	integer	PRIMARY KEY	用户 ID（主键）
username	varchar(64)	UNIQUE, NOT NULL	用户名（唯一）
email	varchar(120)	UNIQUE, NOT NULL	电子邮箱（唯一）
password_hash	varchar(128)		密码哈希值
is_admin	boolean	DEFAULT (False)	管理员权限标识
		DEFAULT	
created_at	datetime	(datetime.utcnow)	账户创建时间

表 3.2 流量数据表

字段	类型	约束	说明
id	integer	PRIMARY KEY	主键
timestamp	datetime	DEFAULT (datetime.utcnow)	时间戳（UTC）
source_ip	varchar(50)		源 IP 地址
destination_ip	varchar(50)		目标 IP 地址
protocol	varchar(20)		协议类型
source_port	integer	NULL	源端口（可为空）
destination_port	integer	NULL	目标端口（可为空）
packet_size	integer		数据包大小
flags	varchar(20)	NULL	TCP 标志位（可为空）
info	text	NULL	附加信息（可为空）
is_anomaly	boolean	DEFAULT (False)	异常标记

表 3.3 日志记录表

字段	类型	约束	说明
id	integer	PRIMARY KEY	主键
traffic_data_id	integer	FOREIGN KEY	关联流量数据 ID
timestamp	datetime	DEFAULT (datetime.utcnow)	日志时间戳
anomaly_type	varchar (50)		异常类型
description	text		详细描述
severity	varchar (20)		严重级别

3.3.2 数据存储策略

为了支持大容量流量数据的高性能存储需求，该工具采用数据分表存储策略、数据压缩技术来提高数据存储与查询的性能。数据分表存储策略将流量数据按时间进行拆分，根据流量数据的时间属性分表，例如按天、按小时进行分表，减少流量数据量，提高流量数据的查询效率。通过时间戳信息查询流量数据所属时间范围，并分到时间范围对应分表中，解决流量数据量大长时间延迟问题。为了节约存储空间，该工具采用数据压缩技术，例如采用 Snappy 算法、Zlib 算法对流量数据进行压缩，尤其对存储长期数据进行压缩，压缩效果比较明显，但进行压缩处理会产生计算开销，需要考虑软硬件资源性能的要求，对压缩等级、解压效率进行取舍。上述策略可以在保证数据完整性的基础上，最大限度地提升数据存储和查询的效率，满足高并发场景下的性能需求^[15]。

3.3.3 数据查询与导出功能设计

数据查询与导出功能的设计主要是为了提供更便捷的数据查询和处理方式，可以提供多个查询条件和导出格式，以满足不同用户的需求。在查询功能方面，在前端提供相对随意的查询条件输入框，根据不同用户的需要可以对其时间、源 IP 地址、目的 IP 地址等字段进行查询。通过 RESTful 的 API 将查询条件传递到后端，在后端查询条件的作用下，后端通过 SQL 语句进行数据库查询操作，然后将查询结果返回到前端进行展示。在查询操作方面，为了方便查询，可以提供分页查询功能，通过分页控制每次查询返回的数据量，降低服务器的压力，提高用户体验^[2]。在数据导出功能方面，可以将查询结果导出到一些常用的格式，比如 CSV、Excel 等，

方便用户离线查看或集成到其他应用中。在后端，通过 python 的 pandas 库将查询输出结果转换成为 pandas 的 DataFrame 对象，然后调用 toseq 或 toExcel 方法生成 csv 或 excel 格式的文件，然后通过 HTTP 的 Response 将文件下载地址返回给客户端。通过以上的设计，系统具备便捷快捷的数据查询，数据导出等操作，满足用户不同的数据分析和处理需求。

3.4 网络流量监控模块设计

3.4.1 实时流量捕获技术

实时流量捕获技术也是网络流量监测模块的核心，采用 Python 的网络抓包库 Scapy 实现。Scapy 是一个功能强大的网络抓包包工具，能针对各种网络场景和各种网络协议进行抓包、解析和生成等操作。在抓包操作过程，Scapy 通过网卡和抓包规则设置，来实时抓取到符合抓包规则的流量，通过 BPF (Berkeley Packet Filter) 过滤器来过滤指定源地址、目的端口号等抓包规则，实现对指定流量的抓包^[1]。通过解析抓包得到的数据，将其解析成 Python 对象后，便于进行特征提取及处理。为提高抓包速度，采用多线程或异步 IO 技术来处理流量大数据，避免出现丢包和因处理速度过慢产生盲区。值得注意的是，实时流量采集过程可能会出现一些异常事件，比如断网、包负载压力等，系统需要具备良好的容错机制，比如使用重试方案复连、高峰数据回放等，从而保证系统采集的稳定性和可靠性。

3.4.2 异常流量检测算法

异常流量检测算法研究旨在通过统计学方法和机器学习算法检测出网络中异于常规的行为，为网络安全防护提供技术保障。基于统计学方法的异常检测算法，通常计算一定时期的平均流量和标准差，将超出平均流量数值阈值的流量数据判定为异常，简单有效但可能对于明显异常以外的异常情况存在误报率较高的问题。机器学习算法通过学习历史流量中正常的行为模式，建立起检测模型，进而提高异常检测的准确度和安全性。常见的机器学习算法包括孤立森林 (Isolation Forest)、局部异常因子 (LOF) 等无监督模型和支持向量机 (SVM)^[13]、随机森林等有监督模型。在实际应用中，系统需结合流量数据的特点选择算法，并通过预处理、特征提取等方法提高模型的效果，如对流量数据进行归一化和特征提取，提高模型训练效率和模型检测精度，同时，也需要及时更新模型参数以适应不断变化的网络环境，降低漏报率和误报率。

3.5 流量分析可视化模块设计

3.5.1 可视化库选择

在流量的分析可视化模块，可视化库的选定对最终的可视化效果图和用户体验感有着很重要的作用，在 Python 中有可视化效果图较好的可视化库有：Matplotlib、Seaborn、Plotly 等可视化库，不同可视化库的特点和适用场景也不同。Matplotlib 是 Python 中最具有代表性的可视化库，其绘图能力强大，可以画折线图、柱状图、饼状图等等，其适用于科学数据的计算、数据分析等场景。Seaborn 是基于 Matplotlib 的高级可视化库，其封装了更加美观的标准化样式和统计图，适用于数据的分布、统计规律等的展现等。Plotly 是交互性强、可视化样式丰富的可视化库，可以实现鼠标停留、图表缩放、数据过滤等功能，其帮助用户实现对数据的探索^[4]。综合本工具需求，选择可视化库 Plotly，不仅可以满足流量数据可视化多样化展示的需求，而且还支持交互操作，可以帮助用户更加清晰地分析数据内容，为网络流量分析提供了更加直观高效的方式。

3.5.2 可视化图表设计

为了更好地反应网络流量数据的特点和流量变化趋势，本工具还提供了折线图、饼状图、直方图等多种类型的图形，每种图形的设计理念和展示效果都各有特点。折线图主要用于展示流量的时间趋势，通过将不同时间的流量用折线连接起来，可以直观地反映流量的变化规律。例如：流量日趋势示意图、流量时对比图等，能够帮助用户识别出流量的高峰点和低谷点，从而更好地对网络流量进行优化。饼状图主要用于描述流量的组成结构，按照不同的规则对流量进行分类，用扇形的面积大小来表示每种流量的占比情况，从而方便用户分析网络流量的来源和去向。直方图可以用于比较不同时间段内的流量数据，例如：将流量数据按照天或周为单位，以柱状图的高度来表示流量的多少。通过这种方式，可以很直观地看出流量的变化趋势。同时为了便于阅读，方便交互，提供了鼠标停留提示，显示具体数值，提供图表缩放、数据选择功能，使用户可以方便地选择想要显示的数据，让用户更容易阅读和分析^[4]。这样本工具就可以把流量数据通过丰富的图形的形式显示给用户，方便网络管理。

4 工具实现

4.1 开发环境搭建

在基于 python 对网络流量分析与可视化分析工具的开发过程中，需要选择好相应的开发环境，才能进行项目开发。本工具开发语言为 python3.8，开发框架为 Django，数据可视化采用 Matplotlib、Plotly，实时流量捕获采用 Scapy、Pandas、NumPy 等工具。开发工具采用 PyCharm，PyCharm 具有强大的编辑和调试功能，能够提升开发效率^[2,4]。数据库管理系统采用以 SQLite3 作为默认的数据库管理系统，因为其轻量级，适合 python。

为了提供上述开发环境，首先要先下载 python3.8，使用 pip 命令下载依赖的库。

在开发框架选择上，Django 有最成熟的 MVT 框架（Model-View-Template），提供强大的功能模块，作为开发工具后台提供技术支撑。前端开发使用 HTML5、CSS3、JavaScript 进行开发，用 ReactJS 框架实现动态化用户界面，提供良好用户体验。整个开发环境的配置要兼顾各种不同，要实现工具在 Windows、Linux、MacOS 上的运行。

4.2 用户管理模块实现

用户管理模块也就是用户注册、登录及权限管理，它是整个工具安全与便利的关键所在，下面分别阐述其子模块的实现过程。

4.2.1 用户注册功能实现

用户名、密码、邮箱等基本信息的输入页面主要是由前端的页面来编写，数据库和程序处理则是在后端，前端页面主要采用 ReactJS 技术，通过网页开发交互式表单，在交互式的表单填写过程中，需要填写用户名、密码、邮箱等基本信息，在表单提交之前，首先要利用 JavaScript 对输入的表单内容进行检测，检测用户名、密码的复杂度，邮箱的填写格式等^[5,13]，检查通过后将数据信息以 JSON 对象格式提交到后端，如图 4.1 所示。

后端接收到注册请求，首先进行二次检查，防止恶意注入和不法字符的输入，利用 Django 的 User 模型，创建新用户，保存到 SQLite3 数据库中。密码保存利用算法（比如：bcrypt）对用户的密码进行加密，保证数据安全。

开发中存在技术壁垒,例如表单重复提交、高并发场景下的数据一致性问题等。前一个问题可通过前端限制按钮,后端采用令牌机制,比如 CSRF Token 令牌来解决;后一个问题,需要改变数据库事务操作处理逻辑,实现数据操作的原子性和隔绝性^[13]。

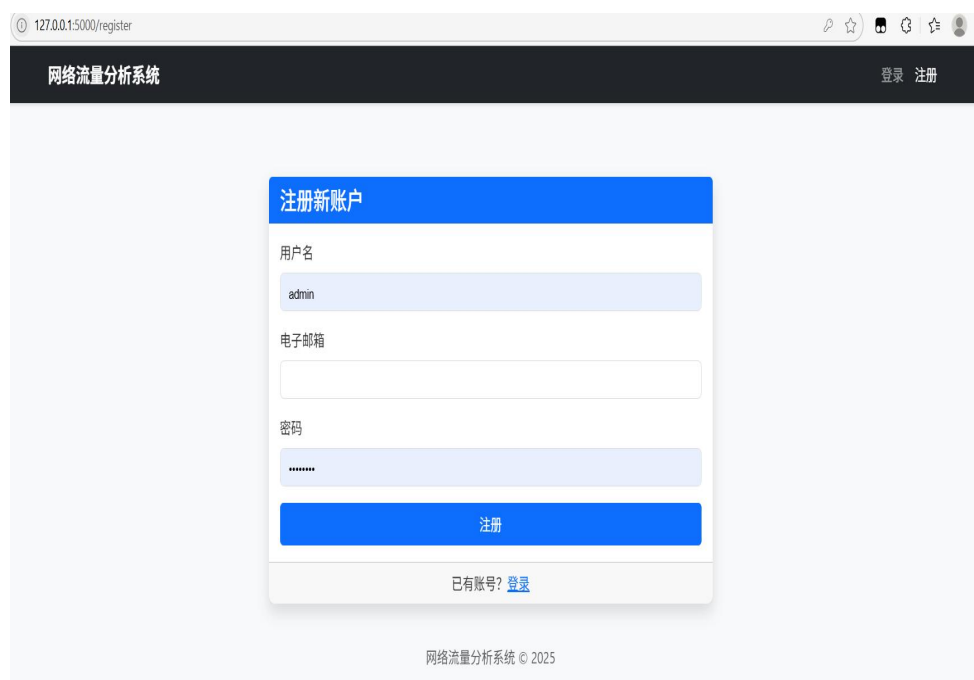


图 4.1 用户注册页面

4.2.2 用户登录功能实现

用户登陆也要经过前端表单处理、后端凭证验证、会话管理等流程。用户登陆前端表单用 ReactJS 编写,在输入用户名密码后以 Ajax 请求向后端发送进行会话管理。用户接收前端的登陆请求,用 Django 自带函数 `authenticate` 进行用户凭证验证,验证成功则用 JWT (JavaScript Token) 进行会话管理^[5,13],如图 4.2 所示。

工具还提供各种安全性的措施来增强登录安全性,例如 HTTPS 协议进行密码传输、暴力破解、CSRF 防护、防止封禁登录尝试,其中 CSRF 防护是通过 Django 自身的中间件来实现的,暴力破解是通过记录失败登录次数、临时禁止登录等方式来实现的^[5]。

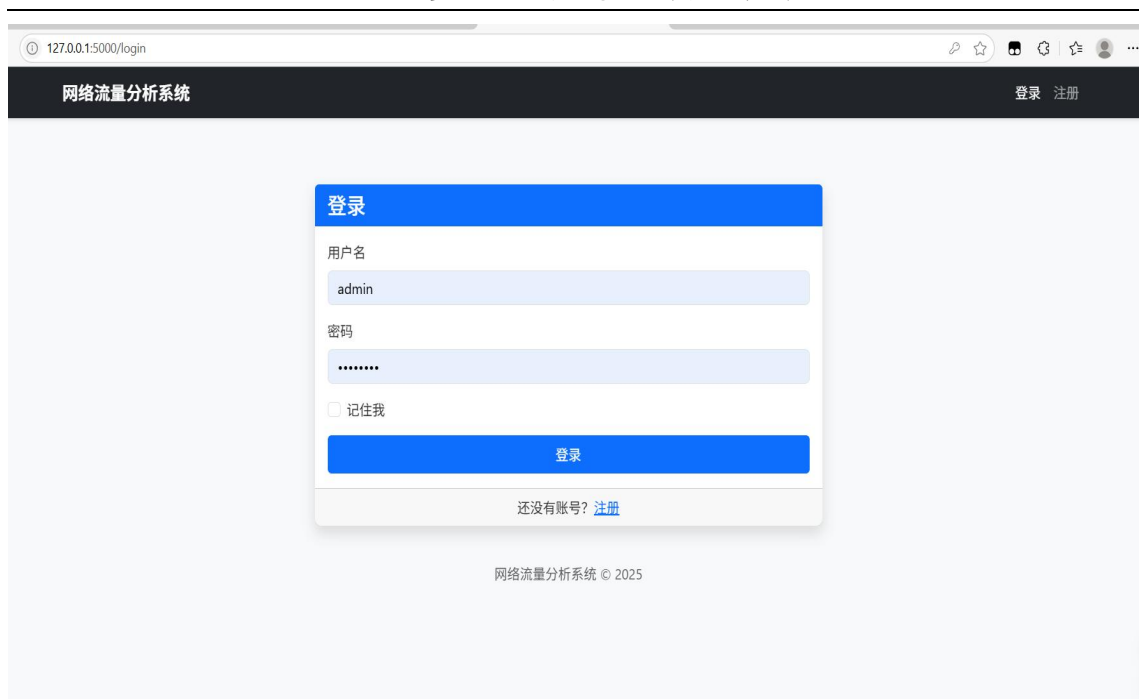


图 4.2 用户登录页面

4.2.3 权限设置功能实现

权限设置功能是以角色进行划分,通过对代码对用户进行权限判断来访问工具功能、访问数据,在 Django 中可以通过自定义进行权限定义,或使用 Django 自带的 Group 模型进行权限的管理。本工具设定两个角色,分别是管理员角色和普通用户角色,管理员角色可以访问所有功能,而普通用户只能访问部分数据及结果^[5, 13]。

权限验证通过 Django 中的中间件实现,用户的每次请求都会通过中间件,中间件会检查用户的角色,判断是否有权限访问当前的 URL。

代码控制上,工具通过@permission_required 等装饰器函数对显示视图进行装饰,如管理员才能调用数据管理中的删除功能,只有管理员才能调用数据管理中的删除功能。

基于以上的实现手段,工具可以确保数据的安全和可用性,并保证禁止非授权用户对敏感数据进行访问和操作。

4.3 数据管理模块实现

数据管理模块负责实现流量数据的存储、查询和导出操作,数据管理设计直接影响整个工具的性能和用户体验,从数据库操作、数据存储、数据查询与导出三个方面阐述。

4.3.1 数据库操作实现

对数据库的操作是数据管理功能的重要行为之一，该工具通过 python 的 sqlite3 库与 SQLite3 进行交互。初始化过程中，通过数据库的迁移脚本创建流量表、用户表等数据表，其中流量表存储了用户时间戳、源 IP 地址、目标 IP 地址、流量大小等信息，用户表存储用户注册信息与权限信息。

在数据操作方面，采用工具提供了增删改查的基本功能，为了提高数据库操作性能，在工具的基础上采用批量插入、事务操作，减少 IO 操作对数据库的性能影响^[2, 15]。

4.3.2 数据存储策略实现

由于流量数据积累过大，同一个表可能出现查询性能下降问题，工具设计分表存储，流量按照时间进行分表存储，每月一个新的表存储每月的流量，表名为 traffic_data_YYYYMM，便于数据查询，便于数据清理归档。

此外，为了提高存储的性能表现，工具采用了数据压缩算法（Snappy）对存储前数据进行压缩，并通过实验验证，这种方式能够在不增加 CPU 负载的情况下有效地减少存储空间消耗。

4.3.3 数据查询与导出功能实现

其中数据查询导出部分是由前端发送条件、后端处理数据并转化为可以导出的形式。其中查询前段使用 ReactJS 编写查询界面，查询条件可组条件查询数据，例如按时间，按 IP 地址，按流量进行数据查询。将查询条件如何进行组合设定在查询界面中，将查询条件组合通过 Ajax 发送到后端生成 SQL 语句。

对于输出而言，工具可以将检索输出以 CSV 或者 Excel 的形式进行导出。其中 csv 格式的输出使用 python 中的 csv 库进行实现，Excel 输出使用 python 中的 openpyxl 库实现。由此，工具即可满足用户的流量数据需求，且导出的数据具有高可用性与兼容性。

4.4 网络流量监控模块实现

网络流量检测模块是工具的一个基本功能，包含实时的网络流量捕获和网络异常流量检测的功能，我们分别对这两个功能进行代码实现和难点分析。

4.4.1 实时流量捕获代码实现

实时流量获取工具采用 python 中的 Scapy 库，其中 Scapy 可以胜任强大的网

络抓包工作，对多种网络报文都有很强的捕获及处理能力。其中运用到使用 Scapy 脚本通过抓取指定网络接口的包，提取其中的关键信息(源网络 IP、目的 IP、端口号名称、流量大小等信息)并入库^[1,10]。具体代码如下图 4.3 所示：

```
@traffic_bp.route(rule: '/capture', methods=['POST'])
@login_required
def capture():
    action = request.form.get('action')
    interface = request.form.get('interface')

    if action == 'start':
        result = start_capture(interface)
        return jsonify(result)
    elif action == 'stop':
        result = stop_capture()
        return jsonify(result)
    else:
        return jsonify({'status': 'error', 'message': '无效的操作'})
```

图 4.3 代码截图

在网络异常中断或者丢包的场景，对于网络异常中断，通过异常捕获机制重新开启捕获，对于丢包，采用环形缓冲（Ring Buffer）保存部分数据包，确保关键数据不丢失^[10]。此外，为提高捕获性能，工具采用多线程捕获方式，监听多网络接口，如图 4.4 所示。

ID	时间	源IP	目标IP	协议	源端口	目标端口	大小	状态	操作
3662	2025-05-25 03:54:47	183.47.109.108	172.16.6.11	UDP	8000	54493	118 bytes	正常	
3661	2025-05-25 03:54:47	172.16.6.11	113.249.150.202	UDP	62559	8005	136 bytes	正常	
3660	2025-05-25 03:54:47	172.16.6.11	111.124.200.101	TCP	25245	443	503 bytes	正常	
3659	2025-05-25 03:54:47	192.144.194.123	172.16.6.11	TCP	443	46368	60 bytes	正常	

图 4.4 流量捕获页面

4.4.2 异常流量检测算法实现

异常流量检测算法主要是使用统计学工具和机器学习模型来检测潜在的攻击

或异常行为。使用统计学方法检测异常流量，统计学方法检测异常流量时使用流量数据的均值、标准差和流量阈值来识别当前流量的异常。例如，如果某个时段的流量大小超过了历史均值的 3 倍标准差，则将当前流量判定为异常^[11, 12]。

在机器学习模型方面，工具采用 LSTM(Long-short memory network)模型进行异常检测建模，用于学习时序流量数据检测异常行为，模型训练分为数据预处理、模型特征提取、模型训练三部分，其中数据预处理是采用 Pandas 对时序流量数据进行数据处理、归一化；特征提取是将数据窗口化提取固定时序数据采用滑动窗口技术；模型训练是学习模型性能超参数(学习率、隐藏单元等)^[18, 19]。经过上述实现，对流量异常数据快速检测，从而保证网络空间安全，如图 4.5 所示。



图 4.5 流量异常分析页面

4.5 流量分析可视化模块实现

流量分析可视化模块将分析出的流量数据以可视化的形式展现给用户，包括可视化图的绘制，交互功能的实现两个部分，下面分别对它们的具体实现代码进行介绍。

4.5.1 可视化图表绘制代码实现

可视化图谱绘制功能通过调用 python 中的 Plotly 库，Plotly 库中图表类型很多，同时还有自定义的图类型，可以满足复杂类型图表的需求，本工具中设置了折线图、饼状图、柱形图三种类型的图，分别应用于流量走势图、流量组成图、不

同时段流量对比图。

饼图的实现则用于分析流量构成。

柱状图是对比不同时间流量的大小，同样和折线图一样在代码逻辑中实现，不再赘述。而对于图表的样式的设置，则是通过工具中的设置选项 Plotly 对字体和颜色和布局进行更改，使图表的显示更加清晰，更加美观^[3,4]，如图 4.6，4.7，4.8 所示。

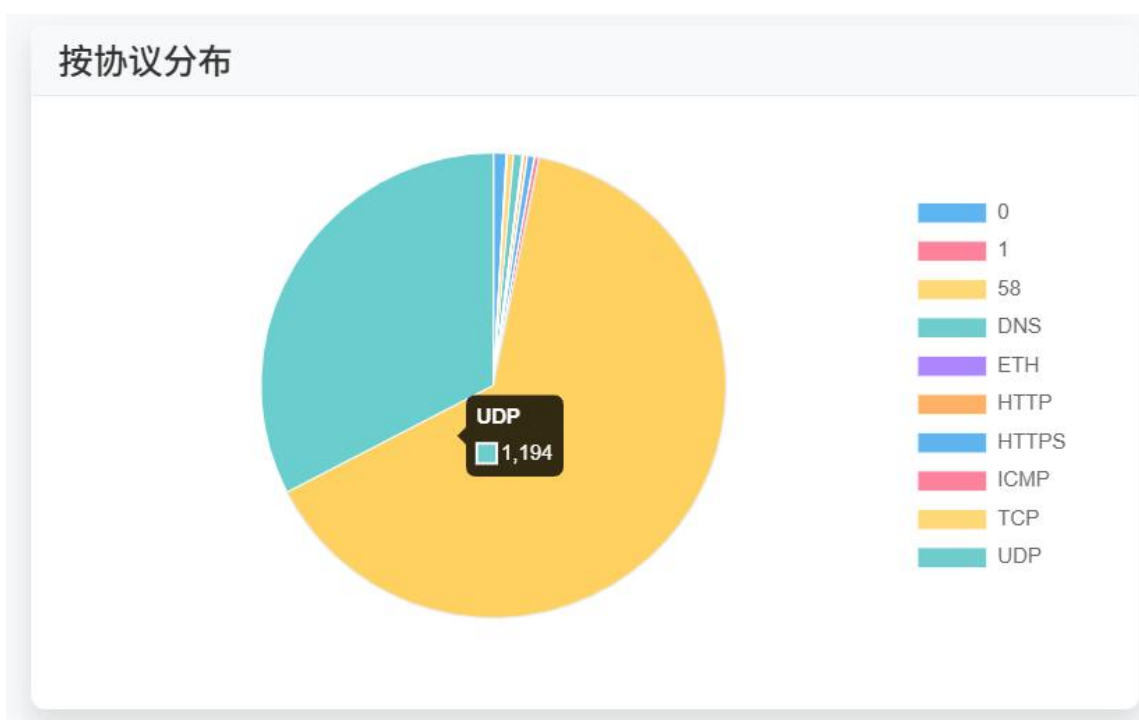


图 4.6 协议分布图



图 4.7 流量趋势图



图 4.8 IP 分布图

4.5.2 可视化交互功能实现

可视化交互实现的功能是为了方便用户使用界面，通过鼠标移动浏览数据。其

中交互功能包括鼠标悬浮提示功能、数据缩放功能和筛选功能。鼠标悬浮提示功能是依靠 Plotly 提供的 `hoverstatic` 实现的，鼠标悬浮在图标上的数据，悬浮提示框会显示出当前数据的详细信息。其中以折线图为例，悬浮提示信息为每一个时刻的流量值大小。数据缩放的功能是借助 Plotly 中的 `Zoom` 实现，通过点击鼠标，移动鼠标滚轮或者拖拽，将图标放大或者缩小，使图标中的信息更加明显。筛选功能是通过前端的 `ReactJS` 模块实现，用户勾选需要筛选的时间筛选条件或者数据源 IP，后端返回前端，并由前端将用户筛选后的数据信息返回给前端。可视化交互工具前后端的通讯通过 `WebSocket` 完成，对数据信息进行更新。

5 工具测试与评估

5.1 测试环境搭建

为了从整体上验证基于 python 网络流量分析和可视化工具的功能与性能，必须进行必要的测试。测试环境应尽量接近生产环境，以便测试结果有效可靠。测试环境包括硬件环境和软件环境。硬件环境：一台高性能的服务器，用于测试工具(配置 CPU: IntelXeon E5, 硬盘: 1TB, 固态硬盘)，多台虚拟机用于模拟分布式网络。软件环境：操作系统:Ubuntu20.04LTS; python:3.8.5; python 依赖安装(pandas, matplotlib, scapy 等); 安装 Wireshark 和 tcpreplay, 用于网络抓包和流量重放, 模拟复杂网络流量。通过上述调整环境可用于工具的功能测试以及性能测试等^[8, 15]。

5.2 功能测试

功能测试指测试功能工具设计的预期目标是否达到，包括对用户管理、数据管理、网络流量抓取、流量分析展现等四个方面的功能测试。

5.2.1 用户管理功能测试

用户管理功能测试主要测试系统的用户注册、用户登录以及用户管理功能。在用户注册功能测试中，利用构造合法用户名、合法密码、合法邮箱输入和输入相同用户名、输入用户弱口令、输入非邮箱地址输入，发现用户注册时系统会拦截非法用户，并且邮箱地址验证功能可以保证用户的真实性。在用户登录功能测试中，测试登录凭证校验和会话管理。在用户登录功能测试中发现，系统的登录凭证使用的是 token 认证，因此不会出现会话劫持的问题。在高并发环境下，出现了延迟现象，可能是系统在数据库响应查询^[5, 13]。在用户权限设置功能测试中，用户管理区分角色，例如，管理员和普通用户，用来测试控制的严格程度。测试中，发现系统会根据用户角色限制用户访问敏感功能，但是在特殊情况下，权限判断逻辑中存在小 bug，需要改进。

5.2.2 数据管理功能测试

数据管理功能测试主要包含数据存储功能测试、数据查询功能测试、数据导出功能测试。数据存储功能测试主要是数据存储方案设计和数据存储方式选择，通过批量导入数据、随机写入数据，验证数据分表存储、压缩存储策略数据存储功能效果。数据查询功能测试主要包含时间类型、流量类型多条件组合复杂查询，测试结

中北大学 2025 届毕业设计说明书

果均在较短时间内返回正确结果，数据查询测试大数据量时数据返回时间稍微有点慢，可能是索引优化的影响^[2, 15]。数据导出功能测试主要包含数据转 CSV、数据转 Excel 导出等，测试结果为文件导出正常，数据完整，如图 5.1，5.2 所示。

可能的数据丢失 如果将此工作簿以逗号分隔(.csv)格式保存, 则某些功能可能会丢失。若要保留这些功能, 请以 Excel 文件格式保存。 不再显示 另存为...															
ID															
ID	时间戳	源IP	目标IP	协议	源端口	目标端口	数据包头	标志	是否异常						
1246	#####	172.20.10.	223.5.5.5	TCP	34113	443	118	PA	否						
1247	#####	172.20.10.	223.5.5.5	TCP	34112	443	118	PA	否						
1248	#####	172.20.10.	223.5.5.5	TCP	34112	443	54	FA	否						
1249	#####	172.20.10.	223.5.5.5	TCP	34113	443	146	PA	否						
1250	#####	172.20.10.	223.5.5.5	TCP	34119	443	66	S	否						
1251	#####	172.20.10.	223.5.5.5	TCP	34113	443	287	PA	否						
1252	#####	172.20.10.	223.5.5.5	TCP	34113	443	244	PA	否						
1253	#####	172.20.10.	223.6.6.6	TCP	34120	443	66	S	否						
1254	#####	223.5.5.5	172.20.10.	TCP	443	34112	66	A	是						
1255	#####	223.5.5.5	172.20.10.	TCP	443	34113	115	PA	是						
1256	#####	172.20.10.	223.5.5.5	TCP	34113	443	85	PA	否						
1257	#####	223.5.5.5	172.20.10.	TCP	443	34119	66	SA	是						
1258	#####	223.6.6.6	172.20.10.	TCP	443	34120	66	SA	是						
1259	#####	172.20.10.	223.5.5.5	TCP	34119	443	54	A	否						
1260	#####	172.20.10.	223.6.6.6	TCP	34120	443	54	A	否						
1261	#####	172.20.10.	223.6.6.6	TCP	34120	443	54	FA	否						
1262	#####	172.20.10.	223.5.5.5	TCP	34119	443	571	PA	否						
1263	#####	223.5.5.5	172.20.10.	TCP	443	34112	54	A	是						
1264	#####	223.5.5.5	172.20.10.	TCP	443	34112	78	PA	是						
1265	#####	223.5.5.5	172.20.10.	TCP	443	34112	54	FA	是						
1266	#####	223.5.5.5	172.20.10.	TCP	443	34113	54	A	是						
1267	#####	223.5.5.5	172.20.10.	TCP	443	34113	89	PA	是						
1268	#####	223.5.5.5	172.20.10.	TCP	443	34113	85	PA	是						
1269	#####	223.5.5.5	172.20.10.	TCP	443	34113	303	PA	是						
1270	#####	223.5.5.5	172.20.10.	TCP	443	34113	394	PA	是						

图 5.1 CSV 导出图

network_traffic_data_20250603_173301.xlsx [受保护的视图] - Excel															
受保护的视图 请注意 - 来自 Internet 的文件可能包含病毒。除非需要编辑, 否则保持在受保护视图中比较安全。 启用编辑(E)															
ID															
ID	时间戳	源IP	目标IP	协议	源端口	目标端口	数据包头	标志	是否异常						
3663	2025-05-29 05:41:18	64:05:e9:83:12:ed	ff:ff:ff:ff:ff:ff	ETH			60		是						
3664	2025-05-29 05:41:18	64:05:e9:83:12:ed	ff:ff:ff:ff:ff:ff	ETH			60		是						
3665	2025-05-29 05:41:18	64:05:e9:83:12:ed	ff:ff:ff:ff:ff:ff	ETH			60		是						
3666	2025-05-29 05:41:18	72:64:54:32:0b:a2	ff:ff:ff:ff:ff:ff	ETH			42		是						
3667	2025-05-29 05:41:18	192.168.9.231	192.168.11.255	UDP	5684	5684	455		否						
3668	2025-05-29 05:41:18	64:05:e9:83:12:ed	ff:ff:ff:ff:ff:ff	ETH			60		是						
3669	2025-05-29 05:41:18	a4:1a:3a:3c:50:4d	ff:ff:ff:ff:ff:ff	ETH			60		是						
3670	2025-05-29 05:41:18	64:05:e9:83:12:ed	ff:ff:ff:ff:ff:ff	ETH			60		是						
3671	2025-05-29 05:41:18	192.168.10.39	182.254.116.117	TCP	7519	80	66	S	否						
3672	2025-05-29 05:41:18	182.254.116.117	192.168.10.39	TCP	80	7519	70	SA	否						
3673	2025-05-29 05:41:18	192.168.10.39	182.254.116.117	TCP	7519	80	54	A	否						
3674	2025-05-29 05:41:18	192.168.10.39	182.254.116.117	TCP	7519	80	206	PA	否						
3675	2025-05-29 05:41:18	182.254.116.117	192.168.10.39	TCP	80	7519	64	A	否						
3676	2025-05-29 05:41:18	182.254.116.117	192.168.10.39	TCP	80	7519	210	PA	否						
3677	2025-05-29 05:41:18	192.168.10.39	182.254.116.117	TCP	7519	80	54	FA	否						
3678	2025-05-29 05:41:18	182.254.116.117	192.168.10.39	TCP	80	7519	64	FA	否						
3679	2025-05-29 05:41:18	192.168.10.39	182.254.116.117	TCP	7519	80	54	A	否						
3680	2025-05-29 05:41:18	192.168.10.39	182.254.116.117	TCP	7520	80	66	S	否						
3681	2025-05-29 05:41:18	192.168.10.39	182.254.116.117	TCP	7521	80	66	S	否						
3682	2025-05-29 05:41:18	192.168.10.39	182.254.116.117	TCP	7522	80	66	S	否						
3683	2025-05-29 05:41:18	192.168.10.39	182.254.116.117	TCP	7523	80	66	S	否						
3684	2025-05-29 05:41:18	192.168.10.39	182.254.116.117	TCP	7524	80	66	S	否						
3685	2025-05-29 05:41:18	26:d0:03:17:ae:d5	ff:ff:ff:ff:ff:ff	ETH			60		是						
3686	2025-05-29 05:41:18	10:3d:1c:cf:61:6a	ff:ff:ff:ff:ff:ff	ETH			60		是						
3687	2025-05-29 05:41:18	fe80::8615:9658:bee5:fb7	ff02::fb	UDP	5353	5353	107		否						
3688	2025-05-29 05:41:18	182.254.116.117	192.168.10.39	TCP	80	7520	70	SA	否						

图 5.2 Excel 导出图

5.2.3 网络流量监控功能测试

网络流量的检测功能主要对实时采集流量的检测功能以及异常流量的异常检测功能进行测试。实时采集流量的检测功能是通过 scapy 库对网络包实时采集，并模拟不同网络流量场景（HTTP、DNS、TCP 连接）进行检测。检测结果能够正常采集到不同网络流量场景下的网络包，在高流量网络场景下检测到存在丢包问题，可能是检测缓冲区的配置问题。异常流量的异常检测功能是采用基于统计学的阈值检测算法结合机器学习模型的方式进行检测。通过模拟正常流量场景以及异常流量场景后，得到异常检测的准确率以及误报率等。检测结果表明基于机器学习的异常检测模型能较好的应对复杂的网络流量场景，但模型训练需要较长时间，需要较高的硬件设备，实用性较差，存在一定的缺陷^[10, 15]，如图 5.3 所示。

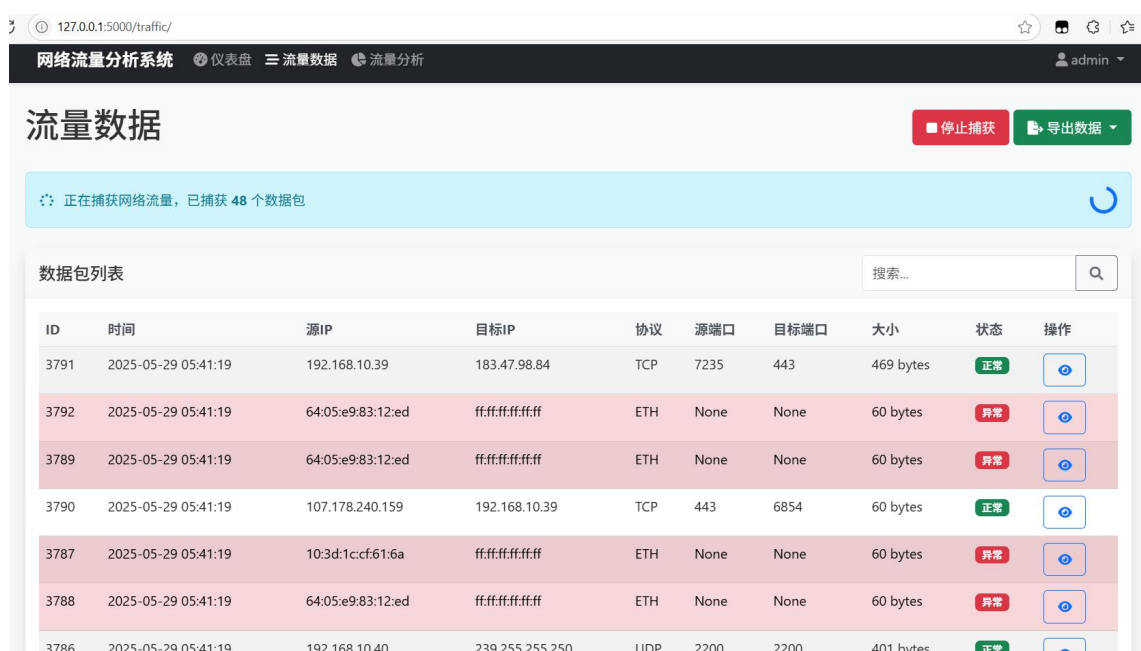


图 5.3 流量捕获功能图

5.2.4 流量分析可视化功能测试

对流量分析的可视化测试主要是对可视化图谱的展示和交互性能进行测试。在可视化图谱的展示测试中，主要是对折线图谱、饼状图谱、柱状图谱等图谱数据展示进行测试，在测试中发现系统会对用户选中的流量数据自动绘制对应图谱，且数据与真实数据一致，图谱视觉效果清晰。在交互性能的测试中，主要是对鼠标悬浮、缩放、筛选等进行测试。在测试中发现交互性能基本流畅，但在大数据展示时缩放和筛选存在一定程度的滞后性，可能是因为前端的渲染性能导致^[3, 4]。同时有些用

户反馈配色单调，需要在后面的版本中增加更多的选项，如图 5.3，5.4 所示。



图

5.3 流量可视化图



图 5.4 流量分析图

5.3 性能测试

性能测试是为了测试工具的真实响应时间与并发量，分为响应时间性能测试和并发性能测试。

5.3.1 响应时间测试

响应时间测试主要是对用户登录、数据查询、图形加载这三个常见操作进行测试，在用户登录测试中分别测试了高并发数和低并发数下的平均响应时间，低并发数(10 个用户同时登录)下的平均响应时间为 1.2s，高并发数(100 个用户同时登录)下的平均响应时间为 3.8，可见数据库中的查询和会话管理在高并发数下存在一定的性能问题。在数据查询中分别测试单个表格查询和多表联合查询，结果显示单个表格查询的平均响应时间为 0.5，而多表联合查询的平均响应时间为 1.5，可能的原因是数据库中查询较为复杂，且未进行索引优化等^[2, 15]。图形加载主要是测试了系统在处理数据量较大(10 万条以上)的情况下的测试，结果显示图形加载的平均响应时间为 2.3，且在 10 万条之后加载时间较长，这可能与加载图形的前端渲染有关。

5.3.2 并发性能测试

并发性能测试，即模拟多用户请求测试并发性能，测试中通过 JMeter 工具模拟生成不同并发数，记录系统资源利用率、响应时间等指标，在低并发（50 并发数，即同时在线 50 个用户的请求）情况下，系统 CPU（20%），内存（25%）等指标，平均响应时间 1.5 秒，在高并发（200 并发数，即同时在线 200 个用户的请求）情况下，系统 CPU（60%），内存（40%）等指标，平均响应时间 4.5 秒。测试中发现，系统在中负载情况下系统性能较好，在高负载情况下，系统的资源利用率较高，响应时间较高，且在高并发情况下，系统部分请求超时，可能跟系统资源调度算法、数据库连接池策略等有关，在后续的优化过程中，可以考虑进行资源动态调度，提高系统的并发能力。

5.4 评估结果分析

综合以上功能测试和性能测试，本工具功能上基本满足了预期目标，用户管理模块、数据管理模块、网络流量监控模块、流量分析可视化模块的核心功能均已正常实现，细节上尚有提升空间，如用户管理模块中的高吞吐量下的登录延迟、大数据管理模块高吞吐量下查询大数据的延迟、流量分析可视化模块高吞吐量下的交互

可视化延迟等问题。性能上在低吞吐量下性能表现良好，在高吞吐量下的系统资源占用率、响应时间均有较大提升，在资源调度、处理并发上还有较大优化空间。综上所述，本工具网络流量分析和可视化方面有较好的实用性，在未来版本中需要添加更好的算法和优化，以及更好的系统可扩展和稳定性^[1,2]。

6 结论

6.1 研究成果总结

通过设计并实现了基于 Python 的网络流量的收集、可视化工具，为网络安全的维护、防护提供便捷、有效、直观的工具。通过对网络中数据流的采集与挖掘，分析网络中的各项流量特征，通过可视化将网络中收集到的数据进行图形化展示，使网络维护人员可以更加直观清楚地查看网络运行情况、潜在的安全威胁。分为用户管理、数据管理、实时流量监控、流量分析可视化等。用户管理可视化工具，方便网络管理员进行用户注册、登录、身份验证，确保网络系统安全可靠；数据管理可视化工具，可以实现对历史数据的存储、查询、导出，便于网络维护人员对数据的维护管理；实时流量监控可视化工具，使用统计学、机器学习的算法，对异常流量进行实时分析检测；流量分析可视化工具，使用折线图、饼状图、柱状图等多种可视化方法对流量趋势、流量构成分析进行可视化，便于网络维护人员可视化分析^[2]。另外，工具在开发基础上是在 python 基础上开发而开发的，python 具有丰富的库资源可以利用（如 pandas、matplotlib、scapy 等），简化开发的同时也使得系统更具扩展性、兼容性。此项工作对网络流量的分析具有重要意义，对网络资源优化、网络安全加固提供了技术支撑。

6.2 经验教训总结

工具的设计、开发和维护过程中也遇到了不少技术瓶颈，总结经验如下。一是数据库设计过程中，面对数据量级大，如何设计合理的数据表，也具有一定的技术挑战。针对数据库设计过程中面临的数据量大，团队通过采用分表技术，同时配合数据压缩存储算法，使得数据库存储和读取更方便^[5]。二是实时流量采集模块设计过程中，面对复杂多变的网络环境，非常容易出现丢包，断包等问题，通过采用流量捕获算法、异常处理算法等设计的思路，避免了丢包现象。三是可视化模块设计过程中，图表美观、操作的方便等也面临一定挑战，通过不断的完善和用户反馈，团队采用 matplotlib 作为可视化库，设计出了多种操作，例如鼠标的悬停、缩放等操作，提升用户的体验。这些经验和教训启示我们，需要在类似的项目中增强前期需求分析与技术选型，同时加强对复杂性的测试与改进，以提高系统稳定性与可用性。

6.3 未来工作展望

尽管本工具已初步实现了网络流量分析与可视化的功能，但是还有许多功能可以进一步实现。例如，流量分析模块可以尝试采用更加先进的机器学习和深度学习模型来分析网络流量，例如基于神经网络的异常检测算法，从而提升异常流量检测的准确率和实时性。又例如可视化模块可以增加更多的可视化图形类型（例如热力图和桑基图）以及尝试三维可视化，以便更加直观地展示流量的时空分布特征。此外，未来还可以考虑将工具实现为分布式框架，例如 ApacheSpark，以实现流量数据的实时计算。最后，工具的 UI 界面也可以考虑进行改进，例如提供更多的可定制化和智能推荐功能。最后，网络发展带来了很多新的网络技术和新的网络应用，工具必须持续改进其提取和分析流量特征方面的能力，以应对不断复杂和不断发展的网络世界^[1,2]。这些改进和优化的领域可以帮助未来对网络流量分析，网络可视化的进一步研究。

参 考 文 献

- [1] 范洁. 基于 Python 的网络流量特征统计分析与可视化[J]. 信息技术与信息化, 2021, (4):49-51.
- [2] 李九州. 基于 Python 的网络流量特征提取和图形化编程[J]. 电脑编程技巧与维护, 2024, (6):142-144.
- [3] 沈杰. 基于 Python 的数据分析可视化研究与实现[J]. 科技资讯, 2023, 21(2):14-17.
- [4] 黄雅琼. 基于 Python 语言的可视化数据分析系统设计与实现[J]. 信息与电脑, 2024, 36(4):97-99.
- [5] 丁宣伊. 基于 Python 的高校电信诈骗数据收集及可视化系统设计[J]. 电脑编程技巧与维护, 2023, (3):90-93.
- [6] 杨孟姣;杜棋东. 基于 Python 爬虫网站数据分析系统设计与实现[J]. 计算机时代, 2022, (11):81-83.
- [7] 陶鹏;杨俊丽. 基于 Python 的可视化数据分析平台设计与实现[J]. 信息记录材料, 2021, 22(8):38-39.
- [8] 司桂静;崔向前;邹坤;戚昭岳. 基于 Python 和 Django 实现城域网流量统计监控系统[J]. 山东通信技术, 2020, 40(1):26-27.
- [9] Tianqing D . Identifying the Origin of Cyber Attacks Using Machine Learning and Network Traffic Analysis[J]. ITM Web of Conferences, 2025, 70
- [10] 汪洁;万丹;袁德玲. 基于用户行为的网络入侵检测与信息可视化[J]. 电脑知识与技术, 2024, 20(5):88-91.
- [11] Kaya O M , Ozdem M , Das R . A novel approach for graph-based real-time anomaly detection from dynamic network data listened by Wireshark[J]. EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, 2025, 12(2).
- [12] Purohit R , Kumar S , Sayyad S , et al. Time-frequency analysis and autoencoder approach for network traffic anomaly detection[J]. MethodsX, 2025, 14:103228-103228.
- [13] 贾震斌;徐芳;王宇;李雨辰;谢艳朋. 大学生体质健康测试系统之后端数据服务

子系统设计[J]. 现代信息科技, 2021, 5(16):1-4.

[14]陈晓凤;谢豆;郑菲;刘文军. 工业设备大数据分析展示系统设计与实现[J]. 信息通信, 2020, (6):119-122.

[15]林建军;林柏钢;杨旻;孙波. 基于流量分析的手机应用识别系统的设计与实现[J]. 信息网络安全, 2016, (8):39-45.

[16]王蔷;郭琪. 基于 Python 语言的微博网络数据可视化系统设计与应用[J]. 电脑编程技巧与维护, 2023, (11):101-104.

[17]丁宇阳;胡涵;王莹露;崔晓会;罗瑞;宋莺. 基于区块链技术的旅游服务及可视化分析系统设计[J]. 长江信息通信, 2024, 37(2):166-169.

[18]Tuyishime E ,Martalò M ,Cotfas A P , et al.Resource-Efficient Traffic Classification Using Feature Selection for Message Queuing Telemetry Transport-Internet of Things Network-Based Security Attacks[J].Applied Sciences, 2025, 15(8):4252-4252.

[19]Nie X ,Xing J ,Li Q , et al.Intrusion detection algorithm of wireless network based on network traffic anomaly analysis[J].Egyptian Informatics Journal, 2025, 30100689-100689.

[20]Kaya O M ,Ozdem M ,Das R .A novel approach for graph-based real-time anomaly detection from dynamic network data listened by Wireshark[J]. EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, 2025, 12(2):

致 谢

在我论文完成过程中，受到过很多无形的帮助和支持，研究上得到过很多帮助和支持，也让我体会到学术研究、团队合作等等的重要性。感谢所有支持我研究的朋友、同事，感谢所有对我无私帮助的人。

感谢我的导师魏月娟、曹峙两位老师，论文的整体撰写上，从论文的选题，到论文框架的确定，到细节的斟酌，导师的学养功底和严谨的治学态度给我诸多指导，对于我的疑惑总是不厌其烦地解答和引导，引导我思考研究本身，特别是工具的设计和实现阶段，导师对于技术实现和论文框架的修改更是对我做出了巨大帮助，可以说没有导师的指导，很难有本论文的完成。

其次，我的同学以及朋友们。在研发的过程中，他们给予了我很多经验和知识上的帮助，而且他们之间相互探讨，相互交流，给我提出了非常多良好的意见。在测试与评价工具的时候，他们给予了工具很大的帮助，他们的参与和评价使我工具更加的完善，并且也给我带来了许多技术上的帮助，在解决技术问题的过程中，他们的技术上的帮助以及他们的无私奉献，使我渡过了难关，他们对于项目的完成做出了不可磨灭的贡献。

最后感谢我的学校，学校给我提供了很好的学习环境以及各种资源，给我的开发与研究论文提供了很好的环境，不管是图书馆还是实验室都给提供很好的资源，对我的论文研究提供了很多帮助，还有各种活动给与很大的帮助，包括讲座，这些让我的见识面更广，让我逐渐对网络流量分析与可视化产生了很大的兴趣。

总之，论文的完成少不了导师的指导，少不了同学、朋友的帮助，少不了学校的良好环境，正是他们的鼎力支持与鼓励让我顺利完成了此课题，我将在未来的学习生活中更加努力，用更加良好的成绩来回报你们。