中国大学生计算机设计大赛

软件开发类作品文档简要要求

作品编号：2025050873

作品名称：基于轻量预训练模型与知识融合的威胁感知系统

作　　者：刘兴琰，全飞扬，梁沃林，王实依

版本编号：V1.0

填写日期：2025.4.20

填写说明：

1、　本文档适用于所有涉及软件应用与开发类的各个小类作品，包括：（1）Web 应用与开发（2）管理信息系统（3）移动应用开发（非游戏类）（4）算法设计与应用（5）信创软件应用与开发（6）区块链应用与开发；

2、　本文档为简要文档，不宜长篇大论，需简明扼要，建议设计二级目录，逻辑性强；

3、　一级标题采用二号黑体，居中，二级标题采用三号黑体，靠左，根据需要可以设计三级标题，正文一律用五号宋体；

4、　提交文档时，以 PDF 格式提交本文档；

5、　本文档内容是正式参赛内容组成部分，务必真实填写。如不属实，将导致奖项等级降低甚至终止本作品参加比赛。

目录

第一章 需求分析

1.1 背景

在 21 世纪，随着互联网的快速发展和数据的指数级增长，网络安全已成为全球面临的重大挑战。企业和个人每天都产生大量的数据，这些数据中可能含有敏感信息，而存储着这些敏感信息的系统则会成为黑客攻击的目标。同时，网络威胁日益复杂多变，传统的安全防护措施难以应对新型的网络攻击手段，大量因为攻击而产生的流量数据也使得防守任务变得异常冗杂。在这样的背景下，如何有效地判别和分类网络流量，识别出潜在的网络威胁，保护数据安全和网络环境的稳定，成为了亟需解决的问题。

而在 2024 年《政府工作报告》中，一个新关键词引发热议——“人工智能+”行动，这是“人工智能+”首次被写入政府工作报告。在今年全国两会中，多位来自网络安全领域的专家都在提案中建议创新发展“AI+安全”，以提高应对网络空间安全风险与不确定性的能力，于是，一个高度先进的威胁感知平台——流智图灵，应运而生。

该平台需要利用最新的人工智能和深度学习技术，提高对网络流量的监控、分析和管理能力，以及快速准确地识别和响应网络安全事件。通过结合 RoBERTa-MiniLM 模型和 LSTM 网络，流智图灵能够加强对网络流量的语义理解和复杂数据模式的识别能力，能够有效判别出网络流量以及系统流量，做出报警，从而防御各种网络威胁。

此外，流智图灵融合了知识图谱技术，产生了网络安全领域的智能问答系统——Sherlock 之眼，这帮助用户在使用时，更便捷的去理解各种复杂的网络安全问题，并支持实时的决策制定、威胁应对和应急响应措施。这样的系统不仅能够提升网络安全管理的效率和效果，还能为用户提供深入的洞察和预测，帮助他们更好地抵御网络安全风险，保护信息资产免受侵害。因此，在当前网络安全形势日益严峻的背景下，像流智图灵这样的威胁感知平台，能够将流量判别这一过程自动化并有智能问答辅助的系统，在市面上还暂未发现竞品，所以该系统将具有广泛的应用前景。

1.2 需求点

针对现今复杂多变的网络安全环境，流智图灵系统面临的需求点如下：

（1）系统流量判别功能与网络流量分类功能需要结合先进的人工智能技术，如 RoBERTa-MiniLM 和 LSTM，AI 模型将精准的进行威胁识别和分类，以帮助用户迅速了解流量的性质和潜在风险；

（2）智能问答系统需结合知识图谱技术，提供深度、可定制的问答服务，帮助用户准确理解和应对网络安全问题；

（3）系统需要实现高效的数据处理能力，特别是在处理大量网络流量数据时，能够保持高速的分析和判断能力，确保网络环境的实时监控和防护；

（4）必须具备强大的数据保护和隐私保障机制，确保所有监测和处理的网络流量数据安全，防止数据泄露或被非法利用，从而维护用户和企业的信息安全。

1.3 目标用户

（1）网络安全管理员和分析师：负责监控和维护网络安全的专业人员，使用流智图灵可以帮助他们识别和防御网络威胁，提高安全监控的效率和准确性；

（2）企业和组织的 IT 部门：包括需要维护网络安全和数据保护的企业、政府机构和

非营利组织，使用该系统可以强化他们的网络防御能力，保护组织免受网络攻击和数据泄露；

（3）网络安全研究人员：从事网络安全研究的学者和技术专家，使用流智图灵可以帮助他们进行网络威胁分析和研究，加深对网络安全问题的理解和掌握；

（4）技术支持和咨询服务提供商：为企业或个人提供网络安全支持和咨询服务的机构，使用该系统可以增强服务能力，为客户提供更有效的安全解决方案。

第二章 概要设计

2.1 总体设计

流智图灵的总体设计完成了给事中——系统流量判别、网络流量分析与分类，Sherlock之眼——智能问答，数据大屏可视化和用户权限管理四个主要功能。这些功能依托于深度学习模型如 RoBERTa-MiniLM 和 LSTM，以及知识图谱技术，确保了高效准确的威胁识别和网络安全管理。给事中功能可以对各类型流量进行分析与判别，可以帮助用户更好的判别流量，让攻击更易于被发现，还可以将 API 对接到各种设备上，实现自动的告警功能。Sherlock 之眼功能提供了智能问答系统融合知识图谱，提供深入且可定制的问答服务，能够帮助用户很好的理解每一个攻击手段以及知识，出现突发状况时，还能辅助用户进行决策以及应急响应。可视化大屏功能，将系统运行状况、流量统计状况都以图表的形式动态的表示出来。除此之外，本系统提供用户登录、权限管理功能，且密码存储至云数据库，全程使用 HASH 加密，使密码无法破解，并且用户交互界面设计简洁友好、操作便捷，极大地降低了用户的学习、使用成本。

2.2 系统架构

为满足用户对于流量分析判断与智能问答的需求，本文架构了逻辑清晰的多层次系统，具体的架构设计如图 2-1 所示。

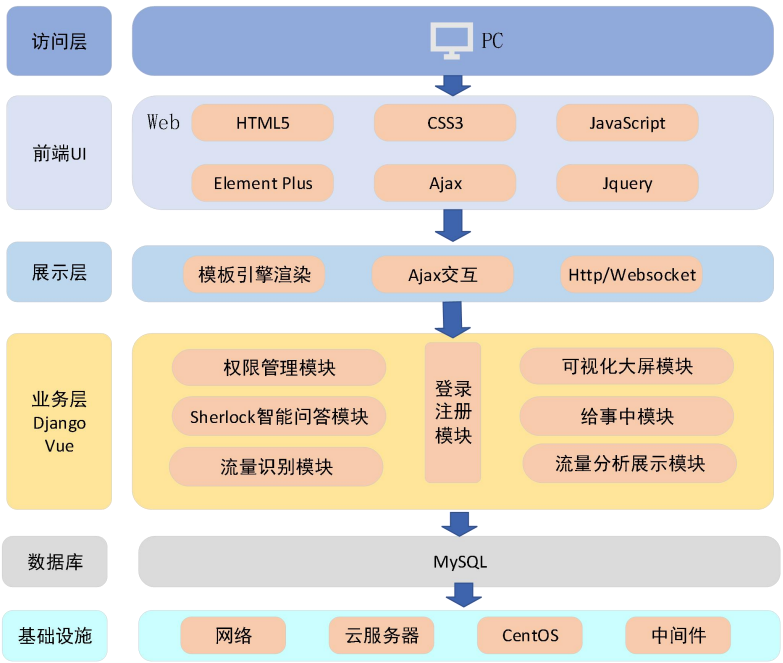


图 2-1 系统层次架构图

2.3 模块设计

本系统共有 5 个模块，分别是：流量检测体验（前台）、流量分析模块、智能问答模块、可视化模块和登录注册权限管理模块。当用户需要检测流量时，可以从前台以及给事中两个模块检测，前台适用于简单判断，且不记录数据，登录后给事中模块则会详细分析并且计入历史数据。当用户对流量以及攻击方式有疑问可以使用 Sherlock 之眼模块来辅助决策。为了方便展示、统计流量数据，系统内置了数据可视化板块。以及考虑到系统的实用性，我们建立了完善的用户权限管理模块，可以便于团队协作。

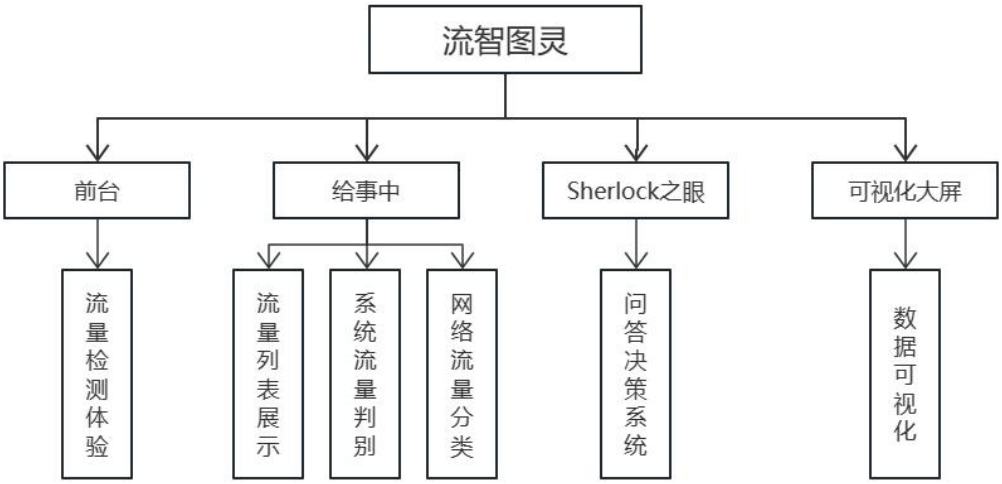


图 2- 2 模块设计

2.4 开发工具与运行环境

2.4.1 开发框架

Django：Django 是一个用于 Web 开发的开源 Python 框架，它由一个模型-视图-控制器（MVC）的架构设计而成，旨在为开发人员提供高效、快速和可扩展的 Web 应用程序开发。其使用松散耦合的组件和插件，这使得它非常适合于大型和复杂的应用程序。它还提供了广泛的第三方库和插件，以便开发人员可以轻松地扩展和定制应用程序。更重要的是，Django 提供了一组内置的安全特性，如跨站点请求伪造（CSRF）防护、XSS 保护和 SQL 注入防护等，可以有效地保护应用程序免受各种安全漏洞的攻击。

Flask：Flask 是一个小巧且灵活的开源 Web 框架，使用 Python 编写。它被设计为轻量级，提供必要的工具和库来构建 Web 应用程序。Flask 遵循简单易用的原则，提供了基本的路由、表单处理和模板渲染功能，使其成为快速开发简单 Web 应用的理想选择。Flask 不强迫开发者使用任何特定的项目或配置结构，给予开发者极大的灵活性。此外，Flask 拥有一个活跃的社区，提供大量的扩展库，使得开发者可以轻松地添加如数据库操作、用户认证和会话管理等功能。

Vue：Vue 是一个渐进式 JavaScript 框架，用于构建用户界面。Vue 的核心库专注于视图层，易于学习且易于与其他库或已有项目整合。Vue 框架是组件化的，允许开发者通过构建可复用的组件来创建复杂的应用程序。它具有响应式的数据绑定和组合的视图组件，使得创建交互式的 Web 界面变得简单高效。Vue 也提供了一个完整的生态系统，包括 Vue Router 进行页面路由处理，Vuex 进行状态管理，以及 Vue CLI 用于项目构建。这些特性和

工具为开发大型单页面应用（SPA）提供了强大的支持。

2.4.2 开发语言

Python：Python 是一种高级、解释型、动态类型的编程语言，可移植性强，且拥有面向对象的特性，拥有丰富的标准库和第三方库，这大大提高了 Python 的开发效率。可扩展性好，Python 支持多种编程范式，包括函数式编程、面向对象编程和面向过程编程，可以轻松地扩展和集成不同的代码和库。

2.4.3 服务端环境

Windows 环境、Linux 环境

第三章 详细技术

3.1 关键技术

本系统的实现有赖于高效的智能模型，本文要求模型能够快速、准确地完成流量特征的提取与分类，以及还需要实现网络安全的智能问答系统，因此我们着重考虑了模型的准确度与资源消耗。本文建立了两个模型，其一是用于识别解析系统流量的系统流量判别模型（System Flow Discriminate Model，SFDM），其二是用于分辨网络流量的网络流量分类模型（Network Flow Classificate Model，NFCM），两个模型均是给予 TensorFlow 深度学习框架。除此以外，我们基于知识图谱技术实现了网络安全领域智能问答系统（Sherlock）。

3.1.1 系统流量判别模型

预训练语言模型（如 BERT、LM）在各种 NLP 任务重取得了显著的成功，为了进一步的使大模型更加适用于我们的任务，在训练模型的过程中，我们运用了**深度自注意力蒸馏**的方法来压缩基于 Transformer 的大型预训练模型。[1]运用处理后的模型提取出系统流量数据中的特征，在通过特征工程和 K 最临近（K-NearestNeighbor，KNN）算法判别出此流量是否为危险流量。

知识蒸馏为深度学习领域新兴的模型压缩及加速技术。[2]-[5]针对特定任务的知识蒸馏不同，与任务无关的 LM 蒸馏模仿原始预训练 LM 的行为，学生模型可以直接在下游任务上进行调整。[6]我们使用基于任务识别的 TransformerLM 深度自我注意力提炼框架。其主要思想是深度模仿自我注意模块，这些模块是基于 Transformer 的教师和学生模型中最重要的组成部分,即将教师网络丰富的知识迁移至学生网络以实现模型压缩并尽量保存模型性能。我们对教师模型最后一次 Transformer 的自我注意模块进行提炼。与以往的方法相比，使用最后一个变换器层的知识而不是进行层与层之间的知识提炼，减轻了教师模型和学生模型之间层映射的困难，而且我们的学生模型的层数可以更加灵活。此外，除了现有研究中使用的注意力分布（即查询和按键的比例点击）之外，我们还引入了自我注意力模块中值之间的按比例点积作为新的深度自我注意力知识。使用自我注意力值之间的缩放点积还能将不同维度的教师和学生表征转换为具有相同维度的关系矩阵，而无需引入额外参数来转换学生表征。它允许学生模型有任意的隐藏维度。

蒸馏阶段，可细分为软损失优化阶段及赢损失优化阶段。软损失优化阶段。利用软损失使得轻量化学生模型能够通过软标签学习教师模型的预测分布。硬损失优化阶段，通过硬标签优化学生网络输出与实际类别间的匹配率。蒸馏形式可定义为：

$$P_{z_i T}=\frac{\exp\left(\frac{z_i}{T}\right)}{\sum_i^C \exp\left(\frac{z_i}{T}\right)}=soft\max\left(\frac{z_i}{T}\right)$$

其中，Z_i 表示第 i 个逻辑单元值，通过 Softmax 函数或 sigmoid 函数对逻辑单元进行处理即可获得模型分类的概率，T 表示蒸馏温度用于软化预测分布。由此，蒸馏过程中的软损失计算如下：

$$L_{\text{soft}}=L_{KL}\left(\mathrm{p}_{Z_t,T},\mathrm{p}_{Z_S,T}\right)$$

其中 Z_t 与 Zs 分别表示教师网络及学生网络的逻辑单元值，L_{KT} 表示 Kullback-Leibler 散度（Kullback-Leibler Divergence，KL 散度）损失，$\mathrm{p}_{z_t,T}$ 指预训练教师模型生成的软标签，$\mathrm{p}_{z_s,T}$ 表示学生模型软预测结果。通过优化 L_{soft} 可逐渐使逻辑单元值 Z_s 与 Z_t 相匹配，此为知识蒸馏的初衷。由此蒸馏过程总损失可定义为：

$$L_{\text{total}}=\lambda\cdot L_{\text{soft}}+(1-\lambda)L_{\text{hard}}$$

其中，λ表示动态调整蒸馏损失的变量因子。L_{hard} 表示硬目标和硬预测间的 CE 损失硬预测表示学生网络未经蒸馏温度软化处理所输出的逻辑单元。

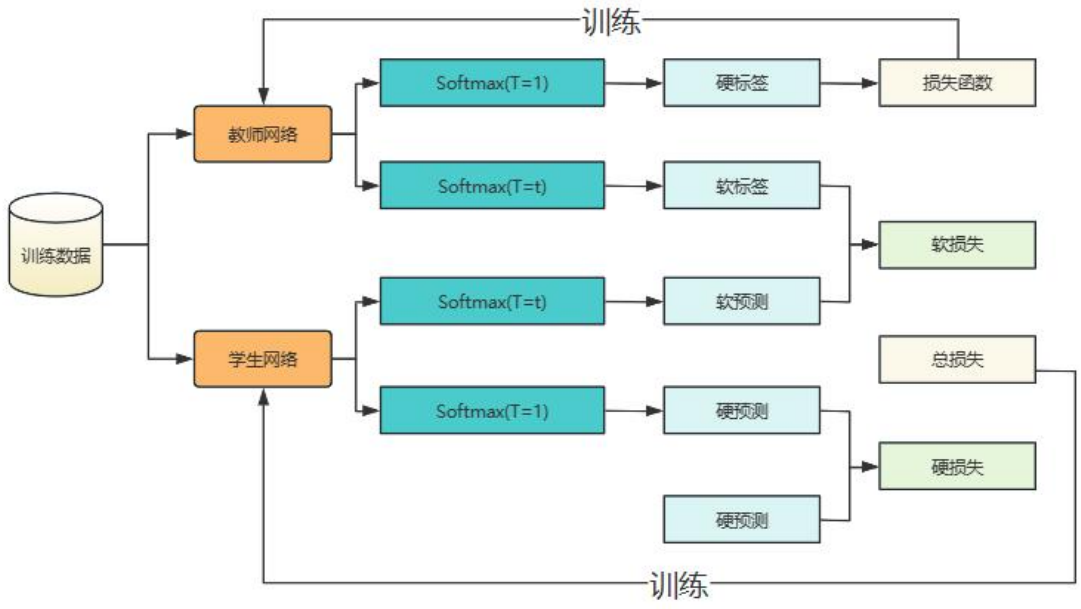


图 3- 1 蒸馏过程

通过输入的系统流量序列，使用 Sentence-Transformer 特征工程提取出特征向量，再将相应的标记嵌入、位置嵌入和段嵌入相加。计算出向量表示（$\{X_i\}_{i=1}^{|X|}$）。计算出相应的标记嵌入、位置嵌入和段嵌入。Transformer 被用来标记上下文信息。输入向量 $\{X_i\}_{i=1}^{|X|}$ 被打包

成：

$$H^{1}=Transforme r_{l}(H^{l-1}),l\in[1,L]$$

其中 L 为变换器层数，最终输出为：$H^{L}=[h_{1}^{L},\ldots,h_{|x|}^{L}]$。其中参数 h_{i}^{L} 在下文中用 x_{i} 表示。

轻量网络测试阶段，通过蒸馏过程所得的轻量化学生模型，只需将系统流量数据输入其中，在经过 KNN 算法的计算，即可预测出每一种类型的概率，然后取概率最高的为结论，进而完成分类任务。

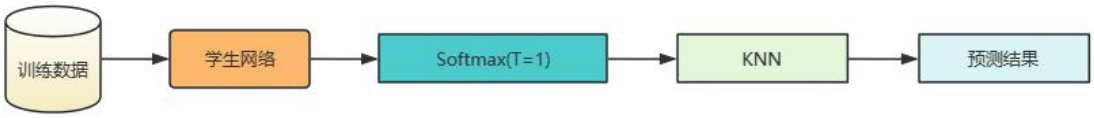


图 3- 2 SFDM 预测流程

3.1.2 网络流量分类模型

随机森林、K 最近邻等被应用于入侵检测中且取得了不错的效果。前,已有许多有监督学习算法,如传统的机器学习算法贝叶斯网络、支持向量机、决策树、机器学习算法通过学习已存在的入侵或者正常模式的网络数据包的特征来发现异常。[7]不过，传统机器学习技术的计算复杂度有限,传统机器学习技术的计算复杂度有限,而深度学习中的神经网络可以从原始数据中自动提取高级特征。本文使用的长短期记忆网络(Long Short-Term Memory Network,LSTM)，相较于传统机器学习方法降低了对于特征工程的依赖，且无需特意去了解专业领域的先验知识即可完成对网络流量数据特征的有效提取，是以能够满足网络流量异常检测及分类的准确性以及鲁棒性。[8]数据集的获取和预处理是训练模型的基础，我们团队训练模型所用到的**数据集**是由我们团队**手动收集、整理**，为了保证数据的平衡性与随机性，我们团队**每天**从流量设备的不同时间段中截取 5000 条数据，最后经过处理变成了训练模型所用到的 JSON 文件。训练模型过程中，我们使用 Tokenizer 分词器将数据集中的文本转换为一个个的单词（或词语）标记，并将其序列化为整数序列。再用 pad_sequences 函数将所有序列填充（或截断）到统一长度 100。再对获取到的标签进行处理，先是用 LabelEncoder 将文本标签转为整数标签，再使用 to_categorical 函数将整数标签进行 ONE-HOT 编码。

本文基于 seq2seq 体系设计了 NFCM，采用 LSTM 作为 Encoder，模型的架构设计主要包括三个主要部分。输入序列时标记为 x_{i}^{j} 的完成分词等预处理的网络流量序列，其中上角标 j 代表序列在序列集中的位置，下角标代表该词在第 i 条序列中的位置。序列中每一个词在输入到 LSTM 网络前要转化成机器可识别的数字编码[9]，我们设计了嵌入层(Embedding)将每个整数索引映射到一个 128 维的向量中，这有助于模型捕捉词汇之间的关系和语义信息。还设计了一个配置为 64 个神经元的 LSTM 层负责处理序列数据，此层设有 20%的输入丢弃率和循环丢弃率以减少过拟合风险。

（1）遗忘门

遗忘门是随着流量信息的深入，忘记前面需要忘记的信息，如之前的键对值和一些噪音，输出是一个 Sigmoid 函数，取值范围为 0-1，与上一课的细胞状态进行按位相乘，0 代表该位的信息彻底忘掉，1 代表该位的信息完全保留，计算公式如下所示：

$$\mathrm{f}_{\mathrm{t}}=\sigma\left(W_{f}\cdot\left[h_{t-1},x_{t}\right]+b_{f}\right)$$

（2）输入门

输入门是为细胞状态提供所需要的新信息，其输出是一个 Sigmoid 函数，取值范围是 0-1，是与当前的细胞状态按位相乘，计算公式如下所示：

$$\mathrm{i}_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\widetilde{C}_\mathrm{t} = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

进而新旧状态信息就可以合并一起了，形成最终的细胞状态，计算公式如下：

$$C_\mathrm{t} = f_t * C_{t-1} + i_t * \widetilde{C}_t$$

（3）输出门

最终的细胞状态与 Tanh 函数相加得到输出，其输出是一个 Sigmoid 函数，取值范围为 0-1，计算公式如下所示：

$$O_\mathrm{t} = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$\mathrm{h}_t = O_t * \tanh(C_t)$$

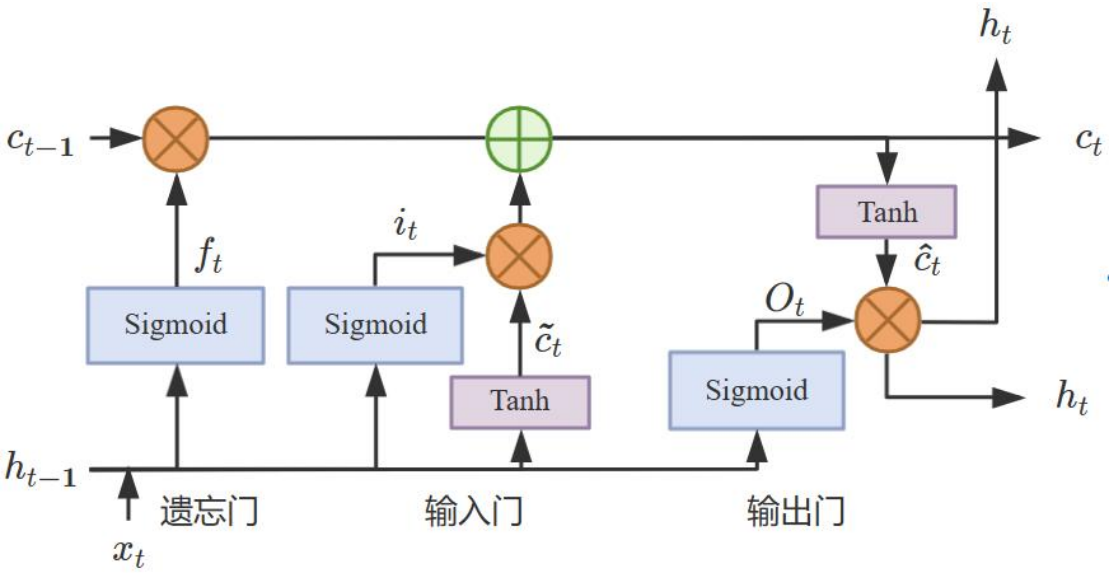


图 3- 3 LSTM 神经元

在上图中，ft 为遗忘门的输出信号，表示记忆单元 c 中的遗忘比例。it 为输入门的输出信号，表示当前输入信息在 c 中的输入比例。$\widetilde{\mathrm{c}}_\mathrm{t}$ 为将要输入到 c 中的预备信息，与 it 进行点乘，得到 c 中的信息。Ot 为输入门的输出信号，表示 c 输出到当前状态 s 中的比例。$\widetilde{\mathrm{c}}_\mathrm{t}$ 为将要输出到隐含层状态 h 的预备信息，与 Ot 进行点乘，得到 h 中的信息。在时刻 t，ct 经过遗忘门、输入门和输出门的筛选，得到 ht。

最后，一个全连接层（Dense）使用 softmax 激活函数输出每个类别的预测概率，输出层的神经元数量等于标签的类别数量，最后取概率最高的输出为 NFCM 预测的结果。

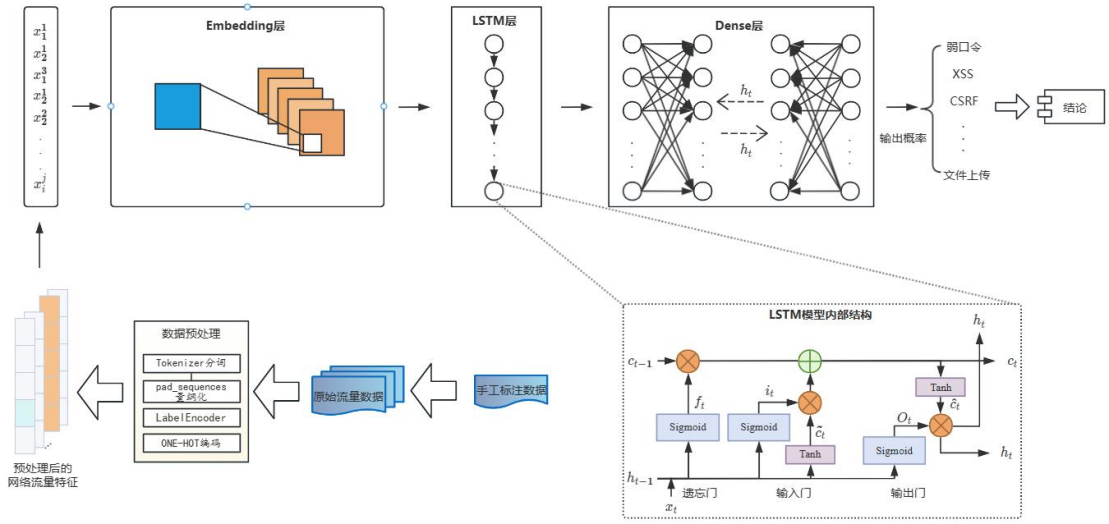


图 3- 4 NFCM 流程图

3.1.3 知识图谱

为了实现网络安全知识图谱——Sherlock 之眼（简称 Sherlock），运用的主要技术为 NEO4J。构建知识图谱首先要定义本体及其之间的关系。本体是知识图谱中的概念，可以将其理解为一类实体的集合，其描述了现实存在的事务。实体间存在的各种内在关联用关系来描述，图谱中丰富的关系有助于发觉深层只是和语义理解[5]。知识图谱一般用三元组表示，基本表现形式为<概念，属性，属性值>和<实体，关系，实体>等，其中实体是最基本的元素，不同的实体之间可能存在不同的关系。

实现 Sherlock 的第一步是数据集的构建，我们团队用于构建 Sherlock 的数据集，来源大致分为两种方式——手工标注整理和网络爬虫获取。前者是我们团队对常见的网络攻击手法的相关信息，如：概念、原理、防御方式等进行收集，从而构建出了一份精简的**网络安全防护数据集**。后者是我们团队运用爬虫技术从国家信息安全漏洞库（CNNVD）、国家互联网应急中心（CNCERT/CC）、白帽汇等国内外漏洞平台或相关论坛中获取到的数据，进过信息抽取、知识融合、知识加工等步骤最终构建出的**网络安全风险漏洞数据集**。

网站名称	地址链接	资源规模
国家漏洞数据库	https://www.cnvd.org.cn	1260 篇新闻类文章
国家互联网应急中心	https://www.cert.org.cn	324 条漏洞公告
国家信息安全漏洞库	http://www.cnnvd.org.cn	188188 条漏洞信息
白帽汇	https://nosec.org	约 100 篇威胁情报文章可看
360 网络安全响应中心	https://cert.360.cn/report	152 篇安全通告

我们团队从上表这些来源中，提取并整理出了属于自己的网络安全三元组数据集，数量较为丰富，与其他领域的知识图谱数据集比较如下表：

序号	数据集名称	领域	实体类别	关系种类	三元组个数
1	电网故障处置知识图谱	工业	6	8	54824
2	水利综合知识图谱	水利	14	13	2807442
3	苹果产业知识图谱	农业	7	6	-
4	水稻病虫害知识图谱	农业	4	6	-
5	金矿知识图谱	地理	7413	568	-
6	医药信息知识图谱	医学	7	11	-

7	炸药配方设计知识图谱	工业	7	6	10780
8	Sherlock（本文）	计算机科学	30	7	446

在我们团队制作数据集时，遇到了以下问题与困难：

（1）网络安全语料数据专业性较强，包含较多领域属于和专有名词，直接将现有的自然语言处理工具应用于“文件上传 ”、“杭州安恒信息技术股份有限公司”等漏洞、企业名称或其他专有名词时，会出现较大误差。

（2）网络安全语料一方面内容丰富，实体和关系的类别较多；另一方面聚焦性强，三元组个数较其他领域语料相对较少。如表所示，本文构建的模式层包含××种实体类别和××种关系种类，明显多余其他领域知识图谱以及相同领域知识图谱，但三元组个数少于其他领域知识图谱。

（3）部分网络安全命名实体会嵌套多个子实体，例如：“远程命令执行漏洞”又可以划分为“命令执行”、“远程执行”、“远程命令执行”三个子实体，导致实体边界识别困难。

本文根据上述网络安全语料数据的特点，提出了适用于此领域数据的知识抽取方法，使用 Google 预训练模型 BERT 和多种深度学习模型从语料中抽取命名实体及实体关系，并构建三元组。

关系类型	数量
概念	10
原理	10
攻击手法	10
防护方式	10
……	……
漏洞信息	23194

3.1.4 数据处理

以上三个模型训练使用了不同类型、不同形式的数据，但是数据处理的方法往往大同小异。在数据处理阶段，我们主要采用了数据清洗、数据增强、数据平衡和归一化来提高模型的准确度。

数据清洗的目的是去除无用信息和异常值，以确保数据的准确性和一致性。这个过程包括缺失值的填充、重复值的去除、异常值的识别和处理等操作。常用的数据清洗技术包括插值法、删除法、替换法等。数据增强则是通过对数据进行一定的变换和扩充，增加训练数据的多样性和数量，以提升模型的泛化能力。数据增强技术包括图像旋转、剪切、缩放等变化，以及数据样本的重复、合成等方法。以本系统的网络流量分类数据集举例。

（1）对于网络流量数据集的数据清洗，我们做了以下几个方面：

1. 去除重复数据：网络流量数据中可能存在相同或类似的内容，需要通过比较请求内容、访问路径等信息，将重复数据删除或合并。

2. 去除噪声数据：网络流量数据中可能存在无意义的标点符号、特殊字符、HTML 标签等噪声数据，需要通过正则表达式或其他方法进行过滤。

3. 处理缺失数据：网络流量数据中可能存在缺失请求方式、请求内容等信息，需要使用插值法、替换法等方法对缺失数据进行处理。

4. 处理异常数据：网络流量数据中可能存在错误的分类标签、请求内容等信息，需要通过数据预处理的方法进行检测和处理，例如基于自然语言处理的方法。

（2）对于网络流量数据集的数据增强，我们做了以下几个方面：

1. 文本转换：将请求内容、请求路径等信息进行编码转换、大小写转换等操作，以扩

充数据集。

2. 数据合成：将同类型的网络流量标签、请求内容等信息进行合并，形成新的数据

（3）对于数据平衡和数据归一化，具体操作如下：

1. 数据平衡

在网络安全数据集中，不同类别的数据量可能会出现不均衡的情况，例如某些类别的网络攻击流量数量比其他类别少很多。这种情况下，模型可能会对数量较多的类别更加敏感，而对数量较少的类别表现不佳。

为了解决这个问题，可以采用数据平衡的方法来处理数据集。我们数据平衡使用过采样方式：对数量较少的类别进行重复采样，使得不同类别的数据量相等。这样可以扩充数据集，增加训练样本的多样性。

2. 数据归一化

在网络流量中，不同的特征可能具有不同的量纲（scale），例如请求包大小和请求时间，这些特征的数值范围相差很大。在这种情况下，如果直接将数据输入到模型中进行训练，可能会导致训练过程收敛缓慢，甚至无法收敛。

为了解决这个问题，可以采用数据归一化的方法来对数据进行处理。常见的数据归一化方法有如下两种：

（1）Min-Max 归一化：将数据的值线性变换到[0,1]的范围内。具体来说，对于数据集中的每个特征，找到其最大值和最小值，然后使用以下公式进行变换

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

其中，x_{min}和x_{max}分别表示数据集中该特征的最小值和最大值。

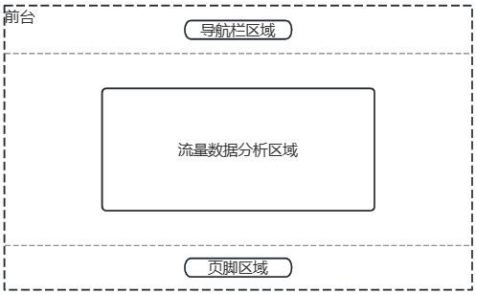
（2）Z-Score 归一化：将数据的值变换为标准正态分布。具体来说，对于数据集中的每个特征，计算其均值和标准差，然后使用以下公式进行变换：

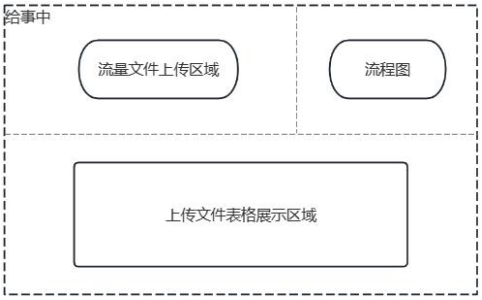
$$x_{norm} = \frac{x - \mu}{\sigma}$$

其中，μ和σ分别表示数据集中该特征的均值和标准差。

3.2 界面设计

本系统前端页面的实现主要是利用了 Vue 框架 Element Plus 前端 UI 组件来完成，布局主要采用了以下几种简约清晰的布局方式，如图 3-5 所示：





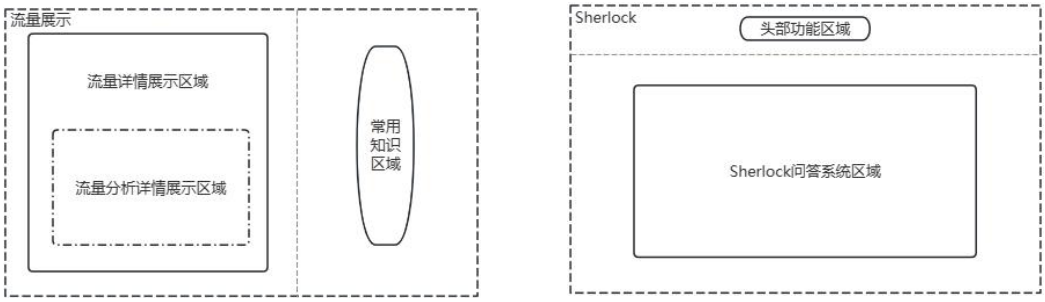


图 3- 5 前端布局

为了使人机交互友好，本项目在前台便可以直接使用部分功能，便于用户更快速的适应项目的使用，体验之后我们推荐登录后，开启本系统的完整功能，前台如图 3-6。

图 3- 6 前台页面

登录后的控制台对本系统的训练数据，模型准确率做了可视化图表用于展示：

图 3- 7 控制台页面

给事中部分总共有系统流量分析和网络流量分析两个功能，顶部的两个按钮都绑定了 Ajax 事件，用于选择想使用的功能。然后下面上传区域可以选择点击或者拖拽上传，下方的表格会将并解析成功的文件呈现出来，页面如图 3-8。

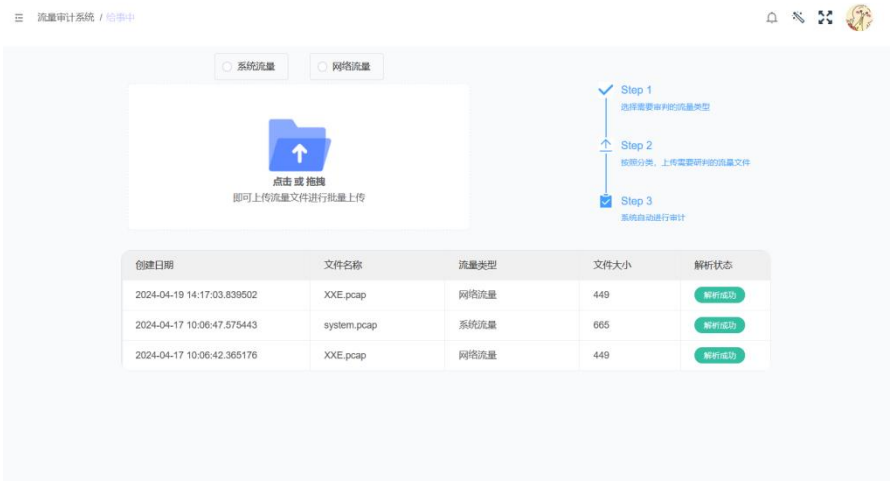


图 3- 8 给事中

然后上传解析成功的流量，会在流量审计系统-流量列表中展示出来：

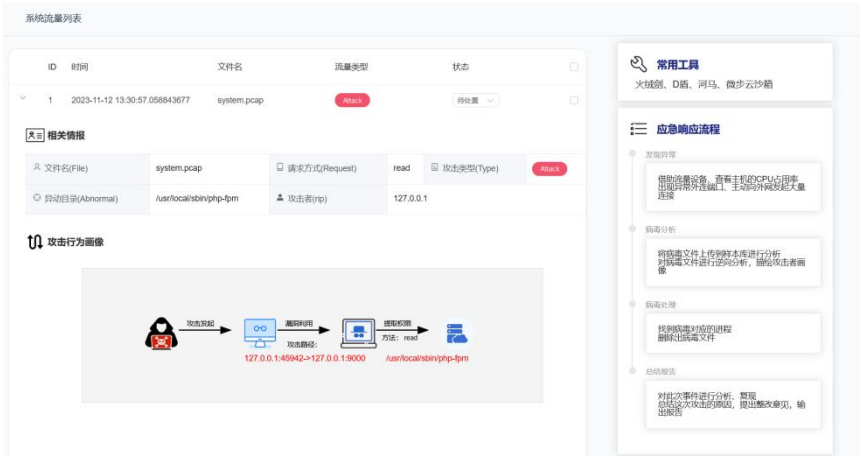


图 3- 9 系统流量展示

其余功能模块的前端部分大致相像，在此不做过多的展示。

3.3 后端设计

本系统的后端为了实现用户权限管理以及前台的临时体验功能，运用了 Flask 和 Django 两个框架，Flask 框架简约精练，我们用它完成了交互功能，而 Django 框架则是能更好的完成后端功能中复杂的逻辑调用，为前端提供可以使用的 API，更安全地与前端进行信息交流。本系统主要的映射如图 3-10：

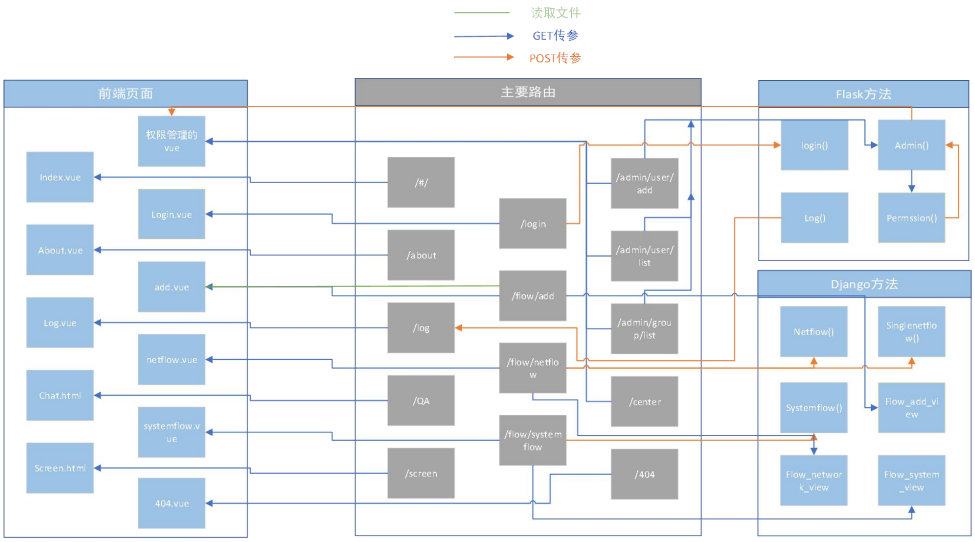


图 3- 10 路由映射图

系统中所有功能模块都是调用的自己的接口与方法，给事中模块中，同一个上传控件实现不同类型上传的核心代码如下：

```
<template>
<div style="text-align: center; margin-top:10px">
    <el-radio-group v-model="radioValue" style="display: inline-block;">
    <el-tooltip effect="light" content="<span><strong>
        <a href='https://www.baidu.com' target='_blank' class='fancy-link'>
          下载系统流量样式
        </a>
        </strong></span>" raw-content>
        <el-radio :value="1" size="large" border>系统流量</el-radio>
    </el-tooltip>
    <el-tooltip  effect="light" content="<span><strong>
        <a href='https://www.baidu.com' target='_blank' class='fancy-link'>
          下载网络流量样式
        </a>
        </strong></span>" raw-content>
        <el-radio :value="2" size="large" border>网络流量</el-radio>
    </el-tooltip>
  </el-radio-group>
</div>
</template>
<script>
Export default {
  computed: {
    uploadAction() {
    return this.radioValue === 1 ? 'http://localhost:8000/api/systemflowpredictor/'
                                 : 'http://localhost:8000/api/netflowpredictor/'
    },
  },
}
</script>
```

第四章 测试报告

4.1 实验设计

1. 系统流量判别功能测试：对于给定的一条系统流量数据，我们采用了八个模型进行预训练，采用 NLP 流行的技术指标进行测试，选取最好的来作为我们的最终模型。

2. 网络流量分类功能测试：对于给定的一篇中文新闻文章，采用人工评判，测试系统能否根据用户需求正确地提取三个不同长度的摘要。

3. 知识图谱功能测试：对于给定的一篇中文新闻文章，采用人工评判，测试系统能否正确地提取出文章的关键词。

数据集样式如图 4-1：

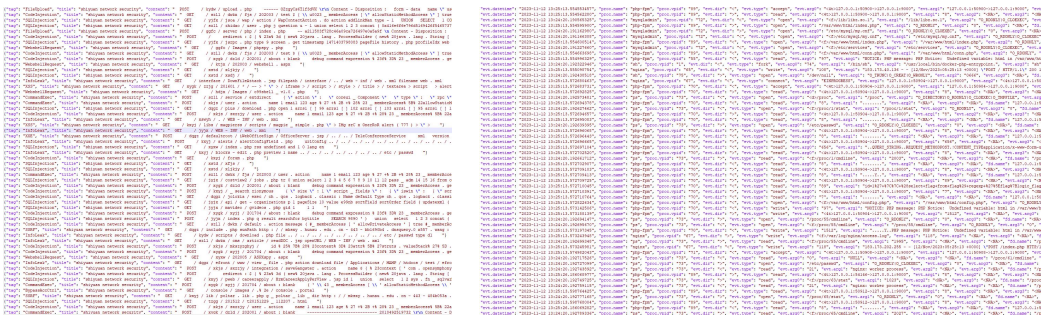


图 4- 1 数据集

4.2 技术指标

F1 score 是精确率（Precision）和召回率（Recall）的调和平均，是一系列用于文本分类和多分类任务结果质量的技术指标，广泛用于衡量分类模型的性能。F1 Score 主要包括三种：

（1）宏平均（Macro-average）：首先对每个类别独立计算 F1 Score，然后计算所有类别 F1 Score 的算术平均。因此所有类别被赋予同等重要性，不用考虑每个类别的样本数量。

（2）微平均（Micro-average）：先计算所有类别的累积真正例（TP）、假正例（FP）和假负例（FN），然后用这些累计值计算 F1 Score。此方式能够较好地处理类别不平衡问题。

（3）加权平均（Weighted-average）：先决步骤与宏平均一致，但在计算 F1 Score 时，会根据每个类别的样本数量来加权。因此样本量越大的类别对最终 F1 Score 分数影响更大。

本文智能模型的效果将以宏平均的 F1 score 作为评价指标，在日志条目上计算。具体计算公示如下：

$$F1\text{score} = \frac{2PR}{(P+R)}$$

其中 P 和 R 的计算公式为：

$$P = \frac{TP}{TP+FP}$$

$$R = \frac{TP}{TP+FN}$$

其中，TP 表示预测为攻击的日志条目确实为攻击日志条目的数量，FP 表示错误地将正常的日志条目预测为攻击日志条目的数量，FN 表示错误地将攻击的日志条目预测为正常日志条目的数量。

经过试验，我们的模型再尝试辅以不同的特征工程（doc2vec、fasttext、sentence-transformer）和算法（逻辑回归、随机森林、K 最临近、支持向量机、决策树）相结合在测试集（4W 多条）上部分实验结果如表 4-1 所示：

表 4- 1 实验结果

特征工程+算法	精确率(%)		召回率(%)		F1-Score		F1 Score
	1	0	1	0	1	0	
Sentence-transformer+KNN	0.64	0.94	0.97	0.48	0.78	0.80	0.79
Fasttext+ RandomForest	0.64	0.95	0.97	0.48	0.78	0.64	0.75
Sentence-transformer+SVC	0.64	0.97	0.98	0.46	0.77	0.62	0.72
Doc2Vec+ Logistic	0.63	0.85	0.91	0.48	0.75	0.62	0.70
Fasttext+ DecisionTree	0.67	0.96	0.98	0.53	0.79	0.68	0.75
Fasttext+KNN	0.63	0.87	0.92	0.48	0.75	0.62	0.70

最后我们经过实验得到的数据进行对比，选定了 Sentence-transformer+KNN，经过参数搜索，最终 F1 Score 来到 0.84。

4.3 实验样例

4.3.1 系统流量判别

对系统流量判别所做实验，我们输入一条系统攻击流量数据如下：


```
{
 "@timestamp": "default",
 "@version": "1",
 "proc.env": "default",
 "team_id": "default",
 "fileds": {
  "fields_under_root": "True",
  "filetype": "sysdig"
 },
 "evt.datetime": "2024-11-12 13:30:57.058843677",
 "proc.name": "php-fpm",
 "proc.vpid": "67",
 "evt.dir": ">",
 "evt.type": "read",
 "evt.arg0": "<4t>127.0.0.1:45942->127.0.0.1:9000",
 "evt.arg1": "8",
 "evt.arg2": "<NA>",
```

```
"evt.arg3": "<NA>",
"fd.name": "127.0.0.1:45942->127.0.0.1:9000",
"proc.pvpid": "51",
"proc.exepath": "/usr/local/sbin/php-fpm",
"evt.rawres": "<NA>",
"fd.lip": "127.0.0.1",
"fd.rip": "127.0.0.1",
"fd.lport": "45942",
"fd.rport": "9000",
"container.id": "af7ba95e745b",
"container.name": "default"
}
```


模型给出的判别为：Attack，即模型判断此条流量为攻击流量，并且在系统中给攻击者行为做出了描述如图 4-2：

图 4- 2 攻击者行为描述

通过与数据集标记对比得出，本系统的系统流量判别功能效果较好，与数据集的标记基本一致，具有极佳的判别能力。

4.3.2 网络流量分类

对网络流量分类所做的实验，我们准备了一条 XXE（XML External Entity）的网络攻击流量数据：

```
POST /Autodiscover/Autodiscover.xml ! DOCTYPE xxe
[ ! ELEMENT name ANY >
<! ENTITY xxe SYSTEM \"file : ///etc/passwd\">]> Autodiscover
<xmlns "http://schemas.microsoft.com/exchange/autodiscover
        /outlook/responseschema/2006a\">
<Request>
<EMailAddress>aaaaa</EMailAddress>
<AcceptableResponseSchema>xxe</AcceptableResponseSchema>
</Request>
</Autodiscover>
```

模型给出的分类结果为：XEE，即模型认为这条网络流量为 XEE 攻击，为了展现功能完善性，此次试验检测在前台的体验功能处完成，效果如图 4-3：

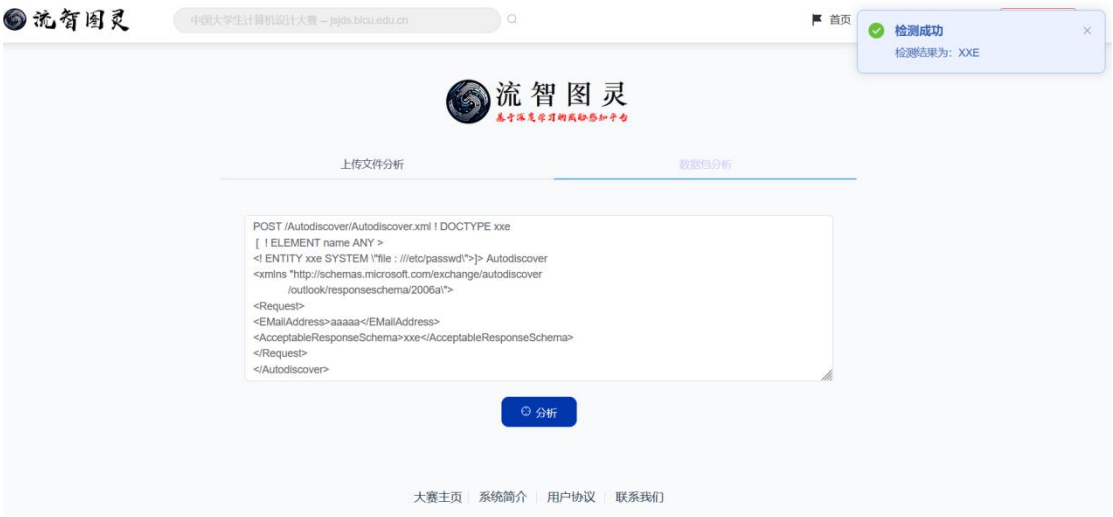


图 4- 3 前台网络流量分析

经过人工评判，网络流量分类功能效果良好，能够有效帮助用户检测出攻击流量类型。

4.3.3 知识图谱问答

我们的测试模拟了一个情景，我们在上一个实验中检测出了 XXE 攻击流量，我们便可以在 Sherlock 中进行询问：

问题：XXE（默认回答 XXE 的概念）

回答：XXE（XML 外部实体注入）是一种利用应用程序处理 XML 数据时配置不当的 XML 解析器来进行的攻击。攻击者可以利用这种漏洞访问服务器上的文件、执行远程代码，或与后端系统进行不当交互。

问题：XXE 的原理

回答：在 XML 处理过程中，如果应用程序未正确配置来禁止外部实体的加载，攻击者可以通过 XML 输入引入恶意内容。这通常通过在 XML 文档类型定义（DTD）中定义外部实体实现。

测试效果如图 4-4：

图 4- 4 Sherlock 之眼

可以发现它能够根据问答的范式，找出正确的答案，并完成前后端的交互，**可以帮助我们在实际应急响应中，辅助用户快速地梳理攻击者的手法，防守方应对攻击所需要进行的应急相应流程。**

4.4 总结

1. 运行速度：经过本地测试，流智图灵的运行速度较快，能够在较短时间内完成对系统流量的判别、网络流量的分类和有关网络安全问题与漏洞的问答等多项操作。

2. 安全性：流智图灵的功能实现使用 Django 框架，做了 SQL 注入和 CSRF 的防护，能够有效保障系统的安全性。

3. 扩展性：流智图灵使用 Django 框架和 Flask 框架作为后端，Vue 框架以及 Element Plus 组件库开发，具有很好的扩展性，可以轻松地添加新的功能模块。通过测试，所有功能接口功能全部正常，使用户使用更加便捷。

4. 部署方便性：翰墨流智图灵的部署非常方便，只需要将系统部署到服务器即可，无需进行复杂的配置。

5. 可用性：经过测试，流智图灵的各项功能均能够正常运行，且操作简便，易于使用，具有较高的可用性。

综合以上测试结果和技术指标，流智图灵是一款功能完备、运行速度快、安全性高、扩展性强、部署方便、可用性较高的威胁感知平台，可以满足用户在日常运维、监测过程中的各种需求。在今后的开发中，我们也会继续加强系统的优化和完善，提升其在各个方面的性能。

第五章 安装及使用

5.1 本地部署

我们提供源代码和已经训练好的模型进行，以便于用户进行本地部署。

5.1.1 环境要求

以下为平台安装环境的最低要求：
操作系统：Windows 11 及以上、Linux 系统
CPU 要求：四核 1.90GHz
RAM 容量：4GB
ROM 容量：60GB
网络：支持 WIFI 或有线网络

5.1.2 部署步骤

(1)下载压缩包“counterflow.zip”到本地；

(2)将电脑中的流智图灵压缩包解压至电脑，解压成功后将在本地获得“frontend”、“django”、“flask”文件夹；

(3)打开 Windows 工具栏，搜索命令提示符 cmd，如操作系统为 Linux，则打开终端；

(4)使用 cd 命令进入“frontend”文件夹，输入命令：npm install、npm run serve；

(5)使用 cd 命令进入“django”文件夹，输入指令：python manage.py runserver 0.0.0.0:8000；

(6)使用 cd 命令进入“flask”文件夹，输入命令：python starter.py；

上面每一个步骤出现网址后代表部署成功，此时将网址 http://127.0.0.1:8080/复制至浏览器即可使用。（Windows 与 Linux 命令相同）

第六章 项目总结

在本系统的开发过程中，我们团队经历了一系列的挑战和机遇。通过这个项目，我们掌握了如何进行项目协调、任务分解、克服困难、水平提升、升级演进和商业推广等诸多方面的知识和技能。

首先，我们的项目协调和任务分解非常重要，这对于项目的进展和成果至关重要。我们需要清晰地定义项目的范围、目标和时间表，并根据团队成员的技能和能力来分配任务，将其转化为可执行的任务列表。通过这个过程，我们可以确定每个团队成员的责任和角色，并确保每个人都明确自己的任务和工作计划。我们还使用了项目管理工具——Teambition，以确保项目按计划顺利进行。

其次，我们克服了一些技术上的困难，特别是在研究和训练深度学习的模型时。这需要团队成员不断学习和探索新技术，同时也需要投入大量时间和精力。但最终，我们成功地训练了可以完成网络流量分类和系统流量判别的模型，并实现了 Sherlock 智能问答以及其他的功能。

同时，通过这个项目，我们的团队也获得了极大的水平提升。我们不仅学会了如何使用自然语言处理算法和其他相关技术，而且还提高了我们的团队协作和沟通能力。在这个过程中，我们的团队成员更好地理解了彼此的技能和能力，并协同工作以实现最佳效果。

最后，我们计划将项目进行升级演进和商业推广。在未来，我们将继续改进和完善流智图灵的功能，包括增加新的特性和服务，并且我们将寻求商业合作伙伴，以将这个项目推向更广泛的市场。

总之，流智图灵的开发是一个挑战和机遇并存的过程。通过项目协调、任务分解、克服困难、水平提升、升级演进和商业推广等方面的努力，我们取得了成功，并且为未来的发展打下了坚实的基础。

参考文献

[1]Wang W, Wei F, Dong L, et al. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers[J]. Advances in Neural Information Processing Systems, 2020, 33: 5776-5788.

[2]Gou J, Yu B, Maybank S J, et al. Knowledge distillation: A survey[J]. International Journal of Computer Vision, 2021, 129: 1789-1819.

[3]黄震华,杨顺志,林威,倪娟,孙圣力,陈运文,汤庸.知识蒸馏研究综述[J].计算机学报,2022,45(03):624-653.

[4]邵仁荣,刘宇昂,张伟,王骏.深度学习中知识蒸馏研究综述[J].计算机学报,2022,45(08):1638-1673.

[5]昝红英,窦华溢,贾玉祥等.基于多来源文本的中文医学知识图谱的构建[J].郑州大学学报(理学版),2020,52(2):45-51.

[6]张天月.不同数据标签场景下的网络入侵检测研究[D].南京;南京邮电大学,2023

[7]王馨彤.基于深度学习的轻量化网络流量异常检测方法研究与实现[D].南京南京邮电大学,2023
[8]Hinton G, Vinyals O, Dean J. Distilling the Knowledge in a Neural Network[J]. stat, 2015, 1050: 9
[9]龚永罡,吴萌,廉小亲,等.基于 Seq2Seq 与 Bi-LSTM 的中文文本自动校对模型[J].电子技术应用,2020,46(03):42-46.