**基于可信度评估的多模型恶意代码检测系统详细设计说明书**

|  |  |  |  |
| --- | --- | --- | --- |
| 总计页数 |  | 版本编号 | V1.1 |
| 编制人员 | 臧玉杰、张嘉鹏 | 受控状态 |  |
| 编写日期 | 2024.05.15 | | |
| 评审日期 |  | | |
| 发布日期 |  | | |

**修订历史**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 修订描述 | 修订人 | 修订日期 | 审核人 | 审核日期 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目 录**

[一、 引言 4](#_Toc90322145)

[1.1 编写目的 4](#_Toc90322146)

[1.2 项目背景 4](#_Toc90322147)

[1.3 参考资料 5](#_Toc90322148)

[二、 设计概述 6](#_Toc90322149)

[2.1 任务 6](#_Toc90322152)

[2.2 运行环境概述 6](#_Toc90322153)

[2.2.1. 硬件要求 6](#_Toc90322157)

[2.2.2. 运行环境 6](#_Toc90322158)

[三、 系统详细设计 7](#_Toc90322159)

[3.1 系统总体框架 7](#_Toc90322162)

[3.2 情报收集模块设计说明 8](#_Toc90322163)

[3.3 训练数据可视化处理模块设计说明 8](#_Toc90322164)

[3.3.1. 模块描述 9](#_Toc90322168)

[3.3.2. 样本库数量信息可视化设计 9](#_Toc90322169)

[3.3.3. 域名分析结果可视化设计 10](#_Toc90322170)

[3.4 模型离线训练模块设计说明 11](#_Toc90322171)

[3.4.1. 模块描述 11](#_Toc90322173)

[3.4.2. 特征工程 12](#_Toc90322174)

[3.4.3. 基础检测算法 18](#_Toc90322175)

[3.5 基于可信度算法的多模型协同检测模块 30](#_Toc90322176)

[3.5.1. 模块描述 31](#_Toc90322178)

[3.5.2. 基于统计学习的可信度计算 31](#_Toc90322179)

[3.5.3. 多模型决策 35](#_Toc90322180)

[3.6 API扩展模块 36](#_Toc90322181)

[3.6.1. 接口设计 36](#_Toc90322183)

# 引言

## 编写目的

本详细设计说明书是针对恶意域名检测系统编写，目的是对该系统进行详细设计，在概要设计的基础上进一步明确系统结构，详细地介绍系统的各个模块，为进行后期实现和测试做准备。本详细设计说明书的预期读者为本系统的设计人员、开发人员、测试人员。

## 项目背景

域名系统(Domain Name System, DNS)是互联网中重要的基础设施之一, 它的主要功能是将域名映射成 IP 地址。当前互联网绝大多数应用都与 DNS 紧密相关, 除了正常应用之外, 很多恶意网络攻击行为也会用到 DNS, 如僵尸网络（Botnet）。僵尸网络管理员通常采用域名生成算法（Domain Generate Algorithm，DGA）生成海量恶意域名，并将这些域名作为命令和控制（Command & Control，C&C）服务器的候选集，从中选取极少数域名进行注册，作为C&C服务器与被控主机之间的“交汇”点。僵尸网络中的被控主机依次在域名系统中查询这些域名的IP地址，如果该域名没有被注册，域名系统将返回“域名不存在”；如果查询的域名是被注册的恶意域名，域名系统返回该域名对应IP地址，即C&C服务器的有效IP地址，接下来被控主机利用该IP地址与C&C服务器建立连接，不断接受来自C&C服务器的恶意指令，进行一系列的恶意活动。DGA算法生成的恶意域名数量数量庞大、时效较短、更新迭代迅速，有效规避了常规的黑名单检测机制，显著提高了僵尸网络的健壮性、可靠性。根据DGArchive的统计数据，到目前为止，已知有超过72种不同的DGA算法，预计DGA算法种类在未来还会进一步增加。对于此类恶意网络攻击，恶意域名的检测是一种重要遏制手段。

早期传统的恶意域名检测一般是通过设置黑名单完成的，但黑名单中包含的域名信息是反病毒组织收集的一些域名列表，是已经知道的恶意域名，因此这种方法有非常大的局限性，无法识别不在黑名单内或新生成的恶意域名，并且需要人工手动不停地补充恶意域名，人力成本极高，因此这样的方式根本无法应对DGA快速生成的大量恶意域名。

人工智能的出现为安全研究人员带来了解决上述局限性的机会，近年出现了很多利用机器学习检测DGA恶意域名的研究。这类方法最大的特点是拥有对未知域名进行预测的能力，但是目前许多的恶意域名检测方法基于静态阈值的方法得出检测结果，这种方法对于已有信息的挖掘和利用率是不充分的，并且对于快速迭代、数量巨大的未知恶意域名检测效果大幅降低。

为了解决这些问题，本文提出构建一个基于可信度模型的恶意域名检测系统，该系统在LSTM、XGBoost、B-RF等算法基础上，引入统计学习的可信度模型，通过Confidence和Credibility两个指标来度量恶意域名的可靠性，充分挖掘已知恶意域名的内在价值；通过控制显著性水平，繁殖出错概率可控的未知情报，提升对未知恶意域名的预判和检测能力，实现对恶意域名的主动检测，帮助维护网络安全。

## 参考资料

编写本文档时主要参考了如下的资料和文献：

1. 《恶意域名检测系统需求设计说明书》
2. 《恶意域名检测系统概要设计说明书

# 设计概述

设计并实现一种融合机器学习和深度学习的DGA域名检测系统。系统基于开源的海量恶意域名和非恶意域名数据进行训练，使用域名字符串作为主要分类特征进行轻量级检测，满足现实网络应用中快速检测的要求。在特征构建中，一方面采用人工的方式提取根据人类经验量化的特征，另一方面利用神经网络挖掘高级特征，增强模型的泛化能力。系统基于Flask框架实现Web页面，将检测结果进行可视化，使用户可以清晰直观的看到DGA检测结果。

## 运行环境概述

### 硬件要求

* 操作系统：Ubuntu 20.04LTS
* CPU： i7 4core 3GHZ或同性能CPU
* 内存：128GB或更多
* 磁盘：10TB或更多
* 数量：2台或更多

### 运行环境

表 1 运行环境说明

|  | 名称 | 版本 |
| --- | --- | --- |
| 编程语言 | Python | 3.5-3.8 |
| 依赖包 | Flask | 1.1.2 |
| pandas | 1.0.5 |
| tensorflow-cpu | 2.2.0 |
| Keras | 2.4.3 |
| scikit-learn | 0.22 |
| tld | 0.12.2 |
| xgboost | 1.1.1 |
| wordfreq | 2.3.2 |

# 系统详细设计

## 系统总体框架

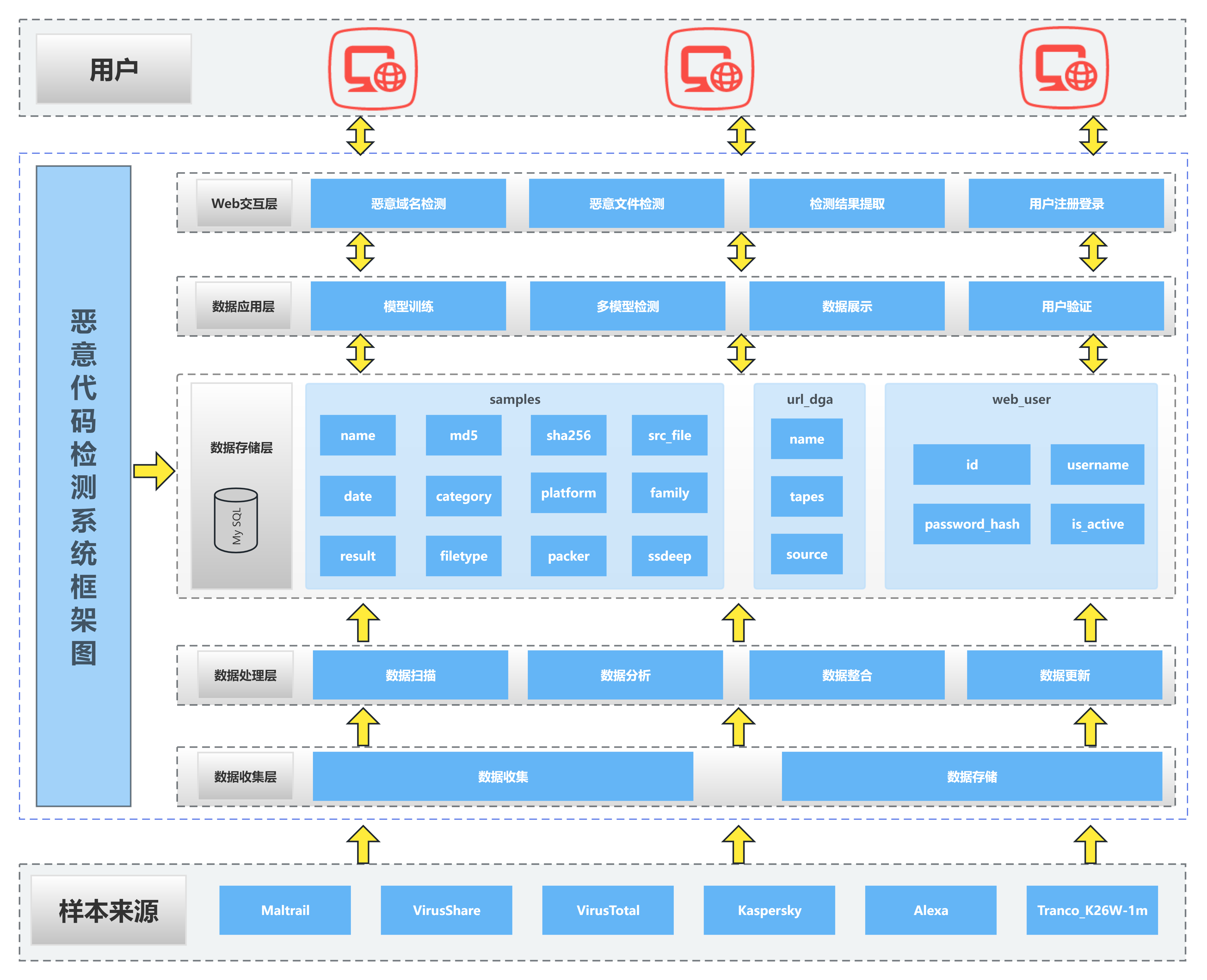


图 1 系统总体框架图

    如上图所示，恶意代码检测系统内部包含五层，外部包含两层，如下所示：

（1）样本来源层：为数据库中的数据获取来源；

（2）数据收集层：收集样本数据，主要包含通过maltrail进行恶意url的收集，以及手动下载VirusShare官网公布的恶意文件样本；

（3）数据处理层：为数据分析处理等操作，包含数据收集、数据存储、数据扫描、数据整合更新等事项；

（4）数据存储层:为将服务器中的数据更新至Mysql数据库中，用于系统开发；

（5）数据应用层：是系统的核心组成部分，包含模型训练模块、多模型检测模块、数据展示模块、用户验证模块等

（6）Web交互层：为用户提供系统的使用入口，是系统界面部分，主要负责接受用户输入、显示系统输出以及其他模块功能的协调调用。用户交互层在接受用户输入的待测样本后，转发给其他模块处理，将包含检测结果的输出展示给用户。此外，还包含用户友好的数据展示功能。

（7）用户层：为检测系统对应用户。

从图1中可得，恶意域名检测系统共包含了以下四部分：

1. 用户交互层：为用户提供系统的使用入口，是系统界面部分，主要负责接受用户输入、显示系统输出以及其他模块功能的协调调用。用户交互层在接受用户输入的待测域名后，转发给其他模块处理，将包含检测结果的输出展示给用户。此外，还包含用户友好的训练数据展示功能。
2. 业务应用层：是系统的核心组成部分，包含了情报收集模块、训练数据可视化处理模块、模型离线训练模块、基于可信度算法的多模型协同检测模块、API扩展模块。
3. 基础组件: 是指系统底层架构组成，其中flask是基于python的web应用框架，本系统利用flask搭建服务器端；Bootstrap是前端框架，本系统利用该框架并结合html、css、js开发实现前端界面；echarts是一个数据可视化组件。
4. 基础设施：是指系统的硬件依赖。为了提高系统的并发行和可用性，我们将该系统部署在云端服务器上，用户可以通过主机访问系统，获取服务。

## 情报收集模块设计说明

情报收集模块的主体内容就是不断更新繁殖的情报库。初始情报库的来源主要有两大部分组成：第一部分是通过网络爬虫技术对360、卡巴斯基、烽火台安全威胁情报联盟等63个可靠情报源的数据，进行持续不断的收集和整合处理，形成含有域名、分类、IP、情报源等信息的情报库；第二部分是通过Alexa提供的网站流量信息，Alexa是一家专门发布网站世界排名的网站，是当前拥有URL数量最庞大，排名信息发布最详尽的网站，本系统将2021年Alexa网站统计的全球访问量排名列表作为良性域名样本的采集池。截至2021年12月10日，整个数据集中已有315,602,090条良性样本及205,016,624条DGA恶意样本，为模型的构建提供了充足的数据储备。规模持续扩大的样本库支撑了各个模型的不断更新，迭代和优化。如REF \_Ref90326136 \h图 2所示。

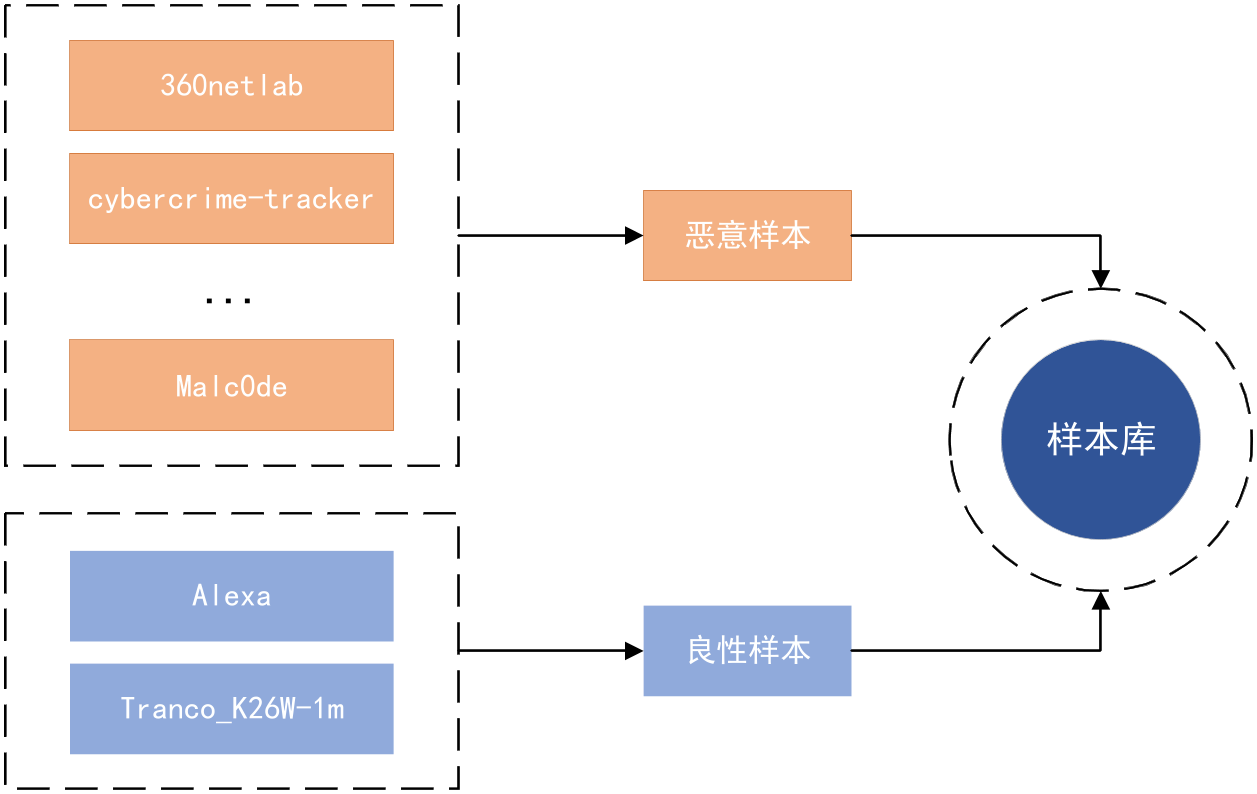


图 2 情报收集流程图

## 训练数据可视化处理模块设计说明

### 模块描述

本模块可以划分为样本库数量信息可视化以及域名分析结果可视化两部分。

### 样本库数量信息可视化设计

本模块基于对样本库数据规模、数据分布的统计，利用echarts组件进行可视化展示，嵌入在前端网页部分，帮助用户进一步了解样本库数据。

表 2 样本库信息可视化页面设计概要

|  | 内容 |
| --- | --- |
| 页面输入 | 空 |
| 页面输出 | 恶意样本总量 |
| 良性样本总量 |
| 每日收集恶意样本数量折线图 |

其中，第一块为训练数据的样本数量展示，包括项目采集的良性样本总数和恶意样本总数，如REF \_Ref90326170 \h图 3所示；第二块为近几个月恶意数据的采集情况，如REF \_Ref90143616 \h图 4所示。



图 SEQ 图 \\* ARABIC3 训练数据量展示图



图 SEQ 图 \\* ARABIC4 近期恶意样本采集情况统计图

### 域名分析结果可视化设计

本模块接收到多个模型返回的检测数据后，进行统一展示。

表 3 域名分析结果可视化设计概要

|  | 内容 |
| --- | --- |
| 页面输入 | 域名字符串 |
| 页面输出 | 最终检测结果（安全、危险、可疑） |
| 各个模型检测结果 |
| 各个模型得出的恶意概率 |

展示的内容包含各个模型给出的预测结果和恶意概率，以表格形式展示给用户，如REF \_Ref90143779 \h图 5所示。在此基础上，系统在域名输入框下方给出最终的检测结果，分别以绿色、红色、橙色字体显示安全、危险、可疑的检测结果，如REF \_Ref90143791 \h图 6所示。



图 5 各模型检测结果展示



图 6 最终检测结果展示

## 模型离线训练模块设计说明

### 模块描述

为了更加全面、多角度的挖掘域名信息，本系统集成了多种传统机器学习方法和深度学习方法，从人工特征以及神经网络高级特征多角度全面分析域名字符串。对于原始情报库中的域名数据，首先进行统一数据清洗、格式化处理后，构建特征集，然后通过多种机器学习算法进行模型的训练。主要流程如REF \_Ref90326322 \h图 7所示。

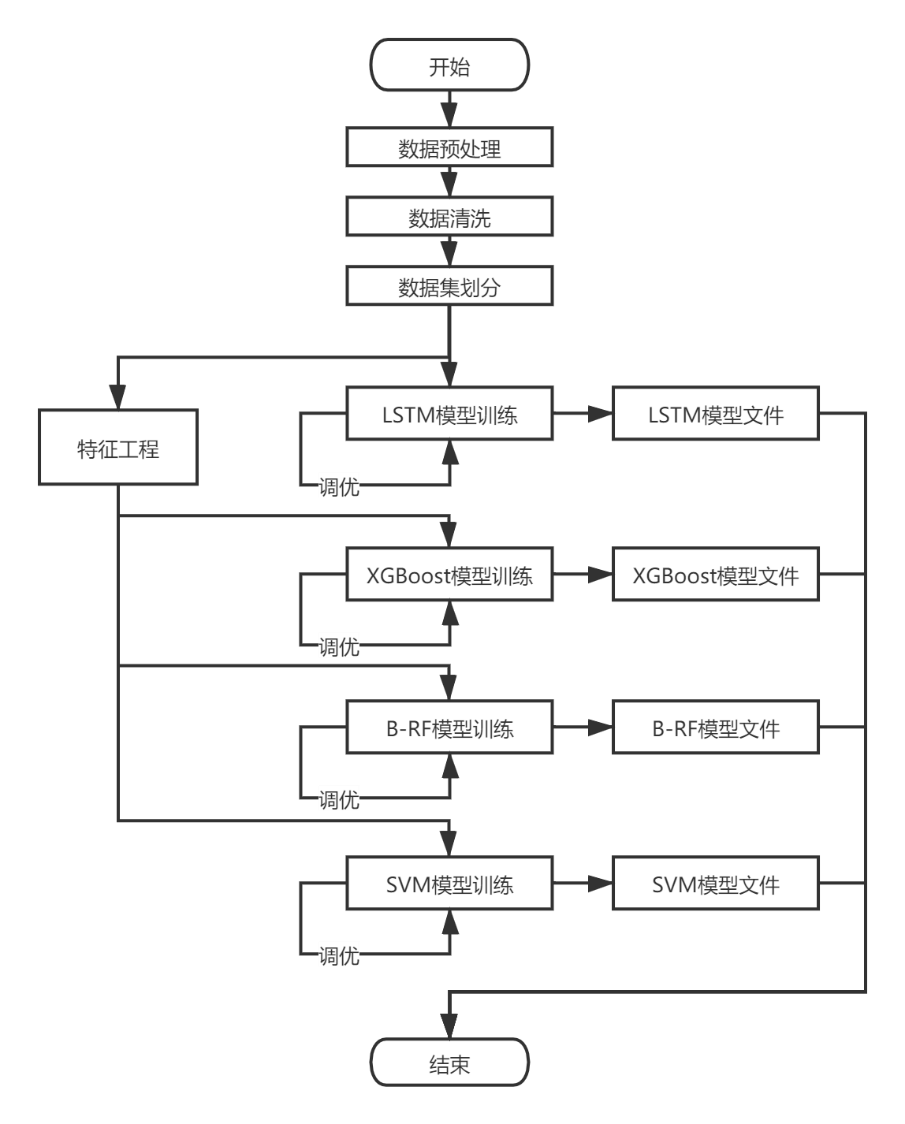


图 7 离线训练模块主要流程

### 特征工程

传统的机器学习方法依赖于人工构建的特征集分析域名，将域名映射到指定特征空间，特征的质量直接影响模型的检测效果。由于DGA生成算法主要对二级域名（SLD）进行操作，因此本系统只关注二级域名和顶级域名（TLD）。当原始域名中还包含了其他标签时，忽略并删除，如“bbs.at.worl.com”，只保留“worl.com”作为研究对象。下文的域名特指由二级域名和顶级域名组成的字符串。本系统的特征工程部分共提取了域名字符串的25个特征，从域名结构、语言模式和字符分布三个角度构建特征集合。

1. **结构特征（Structural Features）**

结构特征共有6种，该类特征代表了域名在字符串结构组成上的特点。表 4对这些特征进行了总结，并以d1和d2为例说明这些特征的具体取值，d1=baidu.com是一个众所周知的正常域名，d2=159vthg1nqmpuyh6.viajes是一个DGA恶意域名。

表 4 域名结构特征表

| # | 特征 | 取值类型 | 意义 | d1 | d2 |
| --- | --- | --- | --- | --- | --- |
| 1 | domain\_len | 整数 | 域名总长度 | 9 | 23 |
| 2 | sld\_len | 整数 | 二级域名长度 | 5 | 16 |
| 3 | tld\_len | 整数 | 顶级域名长度 | 3 | 6 |
| 4 | tld\_dga | 布尔 | 是否包含某些恶意顶级域名 | 0 | 1 |
| 5 | tokens\_sld | 整数 | 由“-”分割的token数 | 0 | 0 |
| 6 | flag\_dig\_sld | 布尔 | 是否以数字开头 | 0 | 1 |

为了更好的理解，下文对表 4中的#4特征展开详细介绍：

**（#4）tld\_dga：**表示顶级域名是否与恶意活动相关，是一个布尔值，1表示相关，0表示无关。文本选用SPAMHAUS[[1]](#footnote-0)在3月1日公布的顶级域名列表作为判断依据，列表中包含了“fail”、“viajes”、“exposed”、“london”、“gq”、“surf”、“cam”、“ml”、“cf”和“work”十个经常出现在恶意活动中的顶级域名。如果顶级域名属于该列表，则认为该域名可能为恶意域名。

1. **语言特征（Linguistic Features）**

语言特征共15种，该类特征主要关注正常域名和DGA域名在语言模式上的差异，对于机器学习分类具有重要作用。REF \_Ref90148201 \h表 5中列出了这些特征信息，以及d1、d2两个实例的具体取值，其中d1=baidu.com，d2=geb3jnercfn28qeq.org。需要注意 的是，REF \_Ref90148201 \h表 5中#10~#21都是针对二级域名提取的特征。

表 5 域名语言特征表

| # | 特征 | 取值类型 | 意义 | d1 | d2 |
| --- | --- | --- | --- | --- | --- |
| 7 | uni\_domain | 整数 | 域名唯一字符数 | 8 | 13 |
| 8 | uni\_sld | 整数 | 二级域名中唯一字符数 | 5 | 12 |
| 9 | uni\_tld | 整数 | 顶级域名中唯一字符数 | 3 | 3 |
| 10 | digits\_sld | 整数 | 数字总数 | 0 | 3 |
| 11 | sym\_sld | 实数 | 特殊字符占比 | 0.0 | 0.0 |
| 12 | hex\_sld | 实数 | 十六进制字符占比 | 0.6 | 0.56 |
| 13 | dig\_sld | 实数 | 数字占比 | 0.0 | 0.19 |
| 14 | vow\_sld | 实数 | 元音字母占比 | 0.6 | 0.19 |
| 15 | con\_sld | 实数 | 辅音字符占比 | 0.4 | 0.63 |
| 16 | rep\_char\_ratio\_sld | 实数 | 重复字符/唯一字符的占比 | 0.0 | 0.25 |
| 17 | cons\_con\_ratio\_sld | 实数 | 连续辅音占比 | 0.0 | 0.38 |
| 18 | cons\_dig\_ratio\_sld | 实数 | 连续数字占比 | 0.0 | 0.13 |
| 19 | gib\_value\_sld | 布尔 | Gib成文检测 | 1 | 0 |
| 20 | hmm\_log\_prob\_sld | 实数 | Hmm成文检测 | 0.0 | -999 |

为了更好的理解，接下来针对REF \_Ref65516820 \h表3.2中的#7、#11、#12、#14、#15、#19、#20特征进行详细介绍：

**（#7）uni\_domain：**表示域名中的唯一字符数，即只出现一次的字符数（不包含“.”、“-”和“\_”）。以d1为例，该域名中的唯一字符是“b”、“a”、“i”、“d”、“u”、“c”、“o”、“m”，因此。特征#8、#9同理。

**（#11）sym\_sld：**是指在二级域名中，特殊符号（“.”、“-”和“\_”）的出现频次与二级域名总长的比值。一般情况下，正常域名为了方便记忆，很少出现特殊字符，但是DGA域名由算法随机生成，因此有较大的概率包含一些特殊字符。

**（#12）hex\_sld：**是指在二级域名中，十六进制字符数（0-9以及a-f）与二级域名总长的比值。一些DGA域名生成算法利用各种加密算法对随机种子加密以及各种转换得到域名序列，因此DGA域名中十六进制字符的占比会较高[54]。

**（#14）vow\_sld：**是指二级域名中，元音字符（“a”、“e”、“i”、“o”、“u”）与二级域名总长的比值。字符序列中元音字符较多时，该字符序列更易读。正常域名为了更加方便人们的记忆，因此可读性更高，而通过算法随机生成的DGA域名的发音拗口，甚至不具有可读性。因此该特征可以帮助模型区分正常域名和DGA域名。

**（#15）con\_sld：**是指二级域名中，辅音字符（“b”、“c”、“d”、“f”等）与二级域名总长的比值。该特征与#14类似，也用于衡量字符序列可读程度，当字符序列中辅音字符占比越高，序列的可读性越低。

**（#19）gib\_value\_sld：**利用Gibberish 方法检测二级域名字符串的可读性，该特征是一个布尔值，1表示字符串可读，0表示字符串不可读，难以发音。Gibberish的原理是使用两个字符的马尔可夫链（2-Character Markov Chain），判断字符串是否为可读字符串。

**（#20）hmm\_log\_prob\_sld：**该特征是利用隐马尔可夫链（Hidden Markov Model，HMM）衡量二级域名的可读性，从而区分正常域名和恶意域名。正常域名一般由常见单词的组合或者某些单词的缩写构成，因此利用这些常见英文词语、单词缩写等训练构建HMM模型。对于一个待测域名，如果该特征值的HMM系数越大，说明该域名更像一个正常域名；HMM系数越小，说明该域名与常用词语组合不同，有较大的概率是一个DGA域名。

1. **统计特征（Statistical Features）：**

统计特征共7个，该类特征从字符分布等角度区别正常域名和DGA域名。REF \_Ref65522771 \h表3.3中总结了这些特征的基本信息，并以d1=baidu.com和d2=geb3jnercfn28qeq.org为例，说明这些特征的具体取值。

表 6 域名统计特征表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | 特征 | 取值类型 | 意义 | d1 | d2 |
| 21 | entropy\_sld | 实数 | 香农熵 | 2.32 | 3.45 |
| 22 | gram2\_med\_sld | 整数 | 2元字符出现频次的中位数 | 4.38 | 3.32 |
| 23 | gram3\_med\_sld | 整数 | 3元字符出现频次的中位数 | 3.17 | 1.56 |
| 24 | gram2\_cmed\_sld | 整数 | 复制SLD中2元字符出现频次的中位数 | 4.23 | 3.32 |
| 25 | gram3\_cmed\_sld | 整数 | 复制SLD中3元字符出现频次的中位数 | 3.17 | 1.49 |
| 26 | gni | 实数 | Gini值 | 0.8 | 0.90 |
| 27 | cer | 实数 | 字符的分类错误 | 0.8 | 0.81 |

为了更好的理解，下文针对REF \_Ref65522771 \h表3.3中的#21、#22、#24、#26、#27特征进行详细的介绍：

**（#21）entropy \_sld：**表示二级域名的香农熵，该值可以衡量二级域名的随机性，一般情况下，正常域名的随机性都较低，香农熵较低，而算法生成的DGA域名的随机性较高，相应的熵值也较高。香农熵的计算方法如公式1所示，其中表示二级域名字符序列，表示其中的每一个字符， 表示二级域名中唯一字符出现的频率。

|  |  |
| --- | --- |
|  | (1) |

**（#22）gram2\_med\_sld：**指二级域名字符串中二元（2-gram）字符组出现频次的中位数。在自然语言中，n-gram字符组的分布是不均匀的，因此该特征可以从n-gram字符组的出现频次这一角度对正常域名和DGA域名进行区分。本文利用wordfreq[[2]](#footnote-1)计算该值。

**（#24）gram2\_cmed\_sld：**该特征也是二元字符组出现频率的中位数。与特征#23不同之处在于，计算该特征之前，需要将二级域名重复依次并首尾相接构造一个新的字符串，假设“baidu”是二级域名，我们使用“baidubaidu”计算该特征。重复操作可以增加字符串长度，有利于n-gram的计算，除此之外，重复操作也可以放大字符串的特点，有利于分类，假设二级域名为“aaaa”，重复后形成“aaaaaaaa”，该字符串看起来更加异常。

**（#26）gni：**指二级域名中字符的基尼值，计算方法如公式2所示，公式中，表示二级域名的唯一字符数（#8），表示唯一字符在该二级域名中出现的频率。

|  |  |
| --- | --- |
|  | (2) |

**（#27）cer：**表示二级域名中字符错误的分类，计算方法如公式3所示，表示唯一字符在该二级域名中出现的频率。

|  |  |
| --- | --- |
|  | (3) |

特征提取工作将每一个输入模型的域名字符串转换为一个25维的输入向量，检测模型根据这些输入向量进行离线训练或实时检测。以下为特征工程的部分代码：

### 基础检测算法

1. **K-最近邻（K-Nearest Neighbor，KNN）**

KNN算法是机器学习算法中最基础、最简单的算法之一。它既能用于分类，也能用于回归。KNN通过测量不同特征值之间的距离来进行分类。KNN算法的思想非常简单：对于任意n维输入向量，分别对应于特征空间中的一个点，输出为该特征向量所对应的类别标签或预测值。

KNN算法是一种非常特别的机器学习算法，因为它没有一般意义上的学习过程。它的工作原理是利用训练数据对特征向量空间进行划分，并将划分结果作为最终算法模型。存在一个样本数据集合，也称作训练样本集，并且样本集中的每个数据都存在标签，即样本集中每一数据与所属分类的对应关系都是已知的。输入没有标签的数据后，将这个没有标签的数据的每个特征与样本集中的数据对应的特征进行比较，然后提取样本中特征最相近的数据（最近邻）的分类标签。一般而言，只选择样本数据集中前K个最相似的数据，这就是KNN算法中K的由来，通常K是不大于20的整数。最后，选择K个最相似数据中出现次数最多的类别，作为新数据的分类。

以下为使用KNN算法进行模型训练的关键代码：

1. **支持向量机（Support Vector Machine, SVM）**

支持向量机（Support Vector Machine, SVM）的主要思想是：建立一个最优决策超平面，使得该平面两侧距离该平面最近的两类样本之间的距离最大化，从而对分类问题提供良好的泛化能力。对于一个多维的样本集，系统随机产生一个超平面并不断移动，对样本进行分类，直到训练样本中属于不同类别的样本点正好位于该超平面的两侧，满足该条件的超平面可能有很多个，SVM在保证分类精度的同时，寻找到这样一个超平面，使得超平面两侧的空白区域最大化，从而实现对线性可分样本的最优分类。

SVM算法具有以下优点：

* 不需要很多样本，不需要有很多样本并不意味着训练样本的绝对量很少，而是说相对于其他训练分类算法比起来，同样的问题复杂度下，SVM需求的样本相对是较少的。并且由于SVM引入了核函数，所以对于高维的样本，SVM也能轻松应对。
* 结构风险最小。这种风险是指分类器对问题真实模型的逼近与问题真实解之间的累积误差。
* 非线性，是指SVM擅长应付样本数据线性不可分的情况，主要通过松弛变量（也叫惩罚变量）和核函数技术来实现，这一部分也正是SVM算法的精髓所在。

以下为使用SVM算法进行模型训练的关键代码：

1. **朴素贝叶斯分类（Naive Bayesian Classification）**

朴素贝叶斯分类是一种非常简单快速的分类算法，通常适用于维度非常高的数据集。因为运行速度快，而且可调参数少，因此非常适合为分类问题提供快速粗糙的基本方案。它的分类思想是这样的：对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，哪个最大，就认为此待分类项属于哪个类别。朴素贝叶斯分类的正式定义如下：

* 设为一个待分类项，而每个为的一个特征属性
* 有类别集合。
* 计算。
* 如果,则。

之所以称为“朴素贝叶斯”，是因为如果对每种类别的生成模型进行非常简单的假设，就能找到每种类别生成模型的近似解，然后就可以使用贝叶斯分类。不同类型的朴素贝叶斯分类器是由对数据的不同假设决定的。

以下为使用朴素贝叶斯分类算法进行模型训练的关键代码：

1. **逻辑回归(Logistic Regression, LR)**

逻辑回归是机器学习中的一种广义的线性回归分析模型，由于算法的简单和高效，在实际中应用非常广泛。逻辑回归从本质来说属于二分类问题，它的本质是：假设数据服从这个分布，然后使用极大似然估计做参数的估计。逻辑回归实际上是使用线性回归模型的预测值逼近分类任务真实标记的对数几率，其优点有：

* 直接对分类的概率建模，无需实现假设数据分布，从而避免了假设分布不准确带来的问题（区别于生成式模型）
* 不仅可预测出类别，还能得到该预测的概率，这对一些利用概率辅助决策的任务很有用
* 对数几率函数是任意阶可导的凸函数，有许多数值优化算法都可以求出最优解

以下为使用逻辑回归算法进行模型训练的关键代码：

1. **决策树（Decision Tree, DT）**

决策树是一种十分常用的分类方法。决策树是一种树形结构，其中每个内部节点表示一个属性上的判断，每个分支代表一个判断结果的输出，最后每个叶节点代表一种分类结果。

* 决策树学习的算法通常是一个递归地选择最优特征，并根据该特征对训练数据进行分割，使得各个子数据集有一个最好的分类的过程。这一过程对应着对特征空间的划分，也对应着决策树的构建：  
  首先，构建根节点，将所有训练数据都放在根节点，选择一个最优特征，按着这一特征将训练数据集分割成子集，使得各个子集有一个在当前条件下最好的分类。
* 如果这些子集已经能够被基本正确分类，那么构建叶节点，并将这些子集分到所对应的叶节点去。
* 如果还有子集不能够被正确的分类，那么就对这些子集选择新的最优特征，继续对其进行分割，构建相应的节点，如果递归进行，直至所有训练数据子集被基本正确的分类，或者没有合适的特征为止。
* 最后，每个子集都被分到叶节点上，即都有了明确的类，这样就生成了一颗决策树。

以下为使用决策树进行模型训练的关键代码：

1. **二分类随机森林（Binary Random Forest，B-RF）**

随机森林（Random Forest，RF）是多个决策树（Decision Tree，DT）的集合，每一颗树都依赖于随机变量的集合。随机森林常被用于解决分类和回归问题。

当训练好的随机森林模型用于分类任务时，向该随机森林模型中输入待分类样本，森林中的每一颗决策树将对该样本独立进行分析和判断，通过统计所有决策树的分类结果，选择得票数最多的分类结果作为最终结果。由于随机森林在决策时整合了所有决策树的分类结果，因此有效克服了单个决策树的局限性，提高了分类准确度。除此之外，随机抽样训练决策树以及随机选择特征集作为特征分裂候选集，有效防止了过拟合情况的发生。随机森林中树与树之间是相对独立的，可以并行构建，加快了模型的训练速度。

以下为使用B-RF算法进行模型训练的关键代码：

1. **自适应提升（Adaptive Boosting, AdaBoost）**

AdaBoost是一种boosting集成学习算法，基本原理就是将多个弱分类器（弱分类器一般选用单层决策树）进行合理的结合，使其成为一个强分类器。Adaboost采用迭代的思想，每次迭代只训练一个弱分类器，训练好的弱分类器将参与下一次迭代的使用。也就是说，在第次迭代中，一共就有个弱分类器，其中个是以前训练好的，其各种参数都不再改变，本次训练第个分类器。其中弱分类器的关系是第个弱分类器更可能分对前个弱分类器没分对的数据，最终分类输出要看这个分类器的综合效果。

AdaBoost用于短决策树。在创建第一个树之后，每个训练实例上的树的性能用于加权创建的下一个树应该关注每个训练实例的注意力。难以预测的训练数据被赋予更多权重，而易于预测的实例被赋予更少的权重。模型一个接一个地顺序创建，每个模型更新训练实例上的权重，这些权重影响序列中下一个树所执行的学习。构建完所有树之后，将对新数据进行预测，并根据训练数据的准确性对每棵树的性能进行加权。

AdaBoost算法具有以下优点：

* 很好的利用了弱分类器进行级联
* 可以将不同的分类算法作为弱分类器
* AdaBoost具有很高的精度
* 相对于bagging算法和Random Forest算法，AdaBoost充分考虑的每个分类器的权重

以下为使用AdaBoost进行模型训练的关键代码：

1. **梯度提升决策树（Gradient Boosting Decision Tree, GBDT）**

GBDT是boosting算法的一种，按照boosting的思想，在GBDT算法的每一步，用一棵决策树去拟合当前学习器的残差，获得一个新的弱学习器。将这每一步的决策树组合起来，就得到了一个强学习器。GBDT的训练过程如REF \_Ref90315235 \h图 8所示。

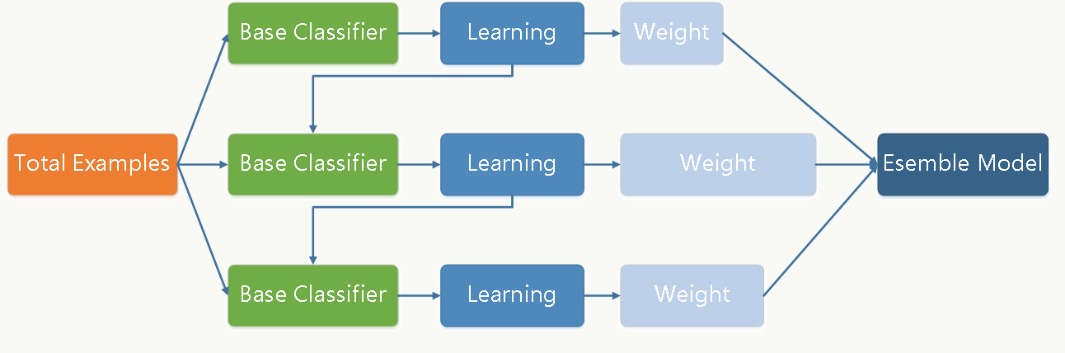


图 8 GBDT训练过程示意图

GBDT通过多轮迭代,每轮迭代产生一个弱分类器，每个分类器在上一轮分类器的残差基础上进行训练。对弱分类器的要求一般是足够简单，并且是低方差和高偏差的。因为训练的过程是通过降低偏差来不断提高最终分类器的精度，弱分类器一般会选择为分类回归树（Classification And Regression Tree, CART）。由于上述高偏差和简单的要求 每个分类回归树的深度不会很深。最终的总分类器是将每轮训练得到的弱分类器加权求和得到的,也就是加法模型。GBDT的算法流程如REF \_Ref90316041 \h图 9所示。

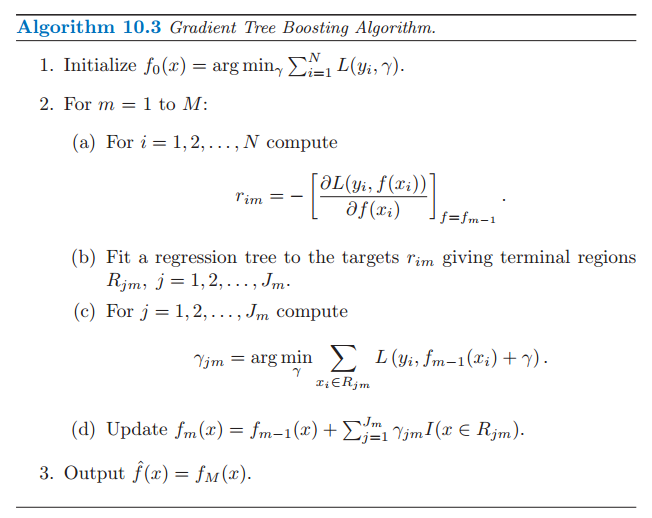


图 9 GBDT算法

除了由损失函数引起的负梯度计算和叶子节点的最佳残差拟合值的计算不同，二元GBDT分类和GBDT回归算法过程基本相似。二元GBDT分类算法和逻辑回归思想一样，用一系列的梯度提升树去拟合这个事件对数几率，其分类模型可以表达为：

以下为使用GBDT进行模型训练的关键代码：

1. **极端梯度提升（eXtreme Gradient Boosting，XGBoost）**

极端梯度提升（eXtreme Gradient Boosting，XGBoost）是一种可扩展的树提升系统（Tree boosting），是数据挖掘、机器学习中的常用方法，在解决数据预测、数据分类等问题的应用中取得了很好的效果。树提升的基本思想如图10所示，利用带有权重的数据依次训练得到多个弱学习器，并且在训练过程中不断拟合上一个弱学习器产生的误差，最后通过集成所有弱分类器构成一个性能更加优越的强学习器。训练数据的权重值决定了训练过程中弱学习器对该数据样本的关注程度。训练数据的初始权重值是随机产生的，其他弱学习器的训练权重则根据上一个弱学习器的学习误差率以及更新权重系数进行调整。XGBoost方法的核心思想就是在迭代过程中，不断向原有树群中添加新树，并且在训练过程中拟合上一轮预测产生的误差。

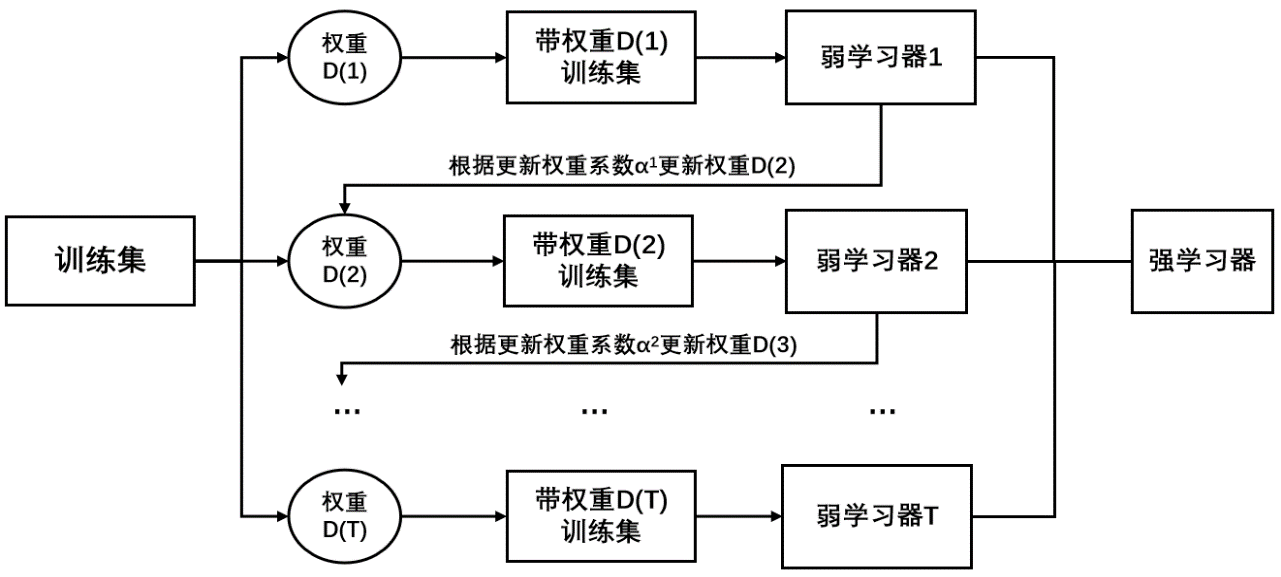


图 10 树提升的基本思想

相比于其他树提升系统，XGBoost方法具有以下优点：

* 避免过拟合。XGBoost利用正则化项、收缩技术和特征子采样三种方式防止过拟合。
* 训练速度快。XGBoost利用加权分位图（Weighted Quantile Sketch）处理加权数据，更新相似树的权重，并且采用并行和分布式的计算加速了学习的过程，除此以外，XGBoost将数据压缩在内存单元中，形成数据块，每个块的数据以压缩列的格式存储，降低了树学习中排序的时间成本，加快了训练速度。
* 处理稀疏数据。XGBoost可以根据训练损失自动“学习”最佳缺失值，因此可以更加有效地处理数据中不同的稀疏模式。
* 处理大规模的数据。XGBoost采用了核外计算、数据压缩以及缓存感知等技术，使得XGBoost可以充分利用有限的计算资源处理大规模的数据，具有很强的扩展性。

以下为使用XGBoost进行模型训练的关键代码：

1. **长短期记忆网络（Long Short-Term Memory，LSTM）**

循环神经网络（Recurrent Neural Network，RNN）不同于传统的神经网络模型，它在前向神经网络的基础上增加了同一层神经元之间的数据流动，也就是说RNN隐藏层的输入不仅包含了上一层的输出，还包含了上一时刻的输出，这使得RNN具有一定的记忆功能，但是传统的RNN存在梯度消失或梯度爆炸等问题，这也意味着RNN只拥有短期的记忆，无法学习序列内部的长期依赖关系。为了解决该问题，研究人员提出了长短期记忆网络。

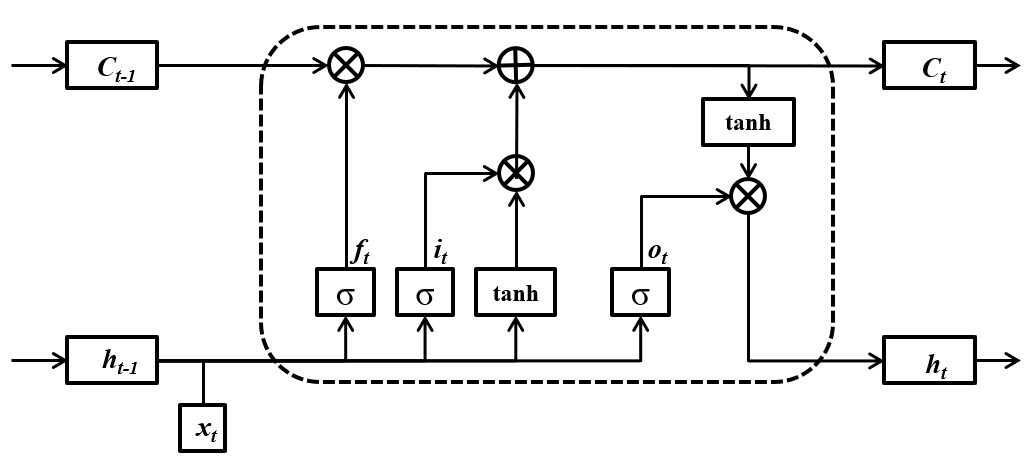


图 11 LSTM内部结构图

长短期记忆网络是一种时间循环神经网络，是为了解决一般的循环神经网络存在的长期依赖问题而专门设计出来的。它的内部结构如图11所示，可见，LSTM在原有RNN的基础上引入了细胞状态（Cell state）并增加了输入门（Input gate）、遗忘门（Forget gate）和输出门（Out gate）三个门控单元，实现了对长序列的记忆，缓解了梯度消失问题。LSTM内部计算过程如公式5至10所示。

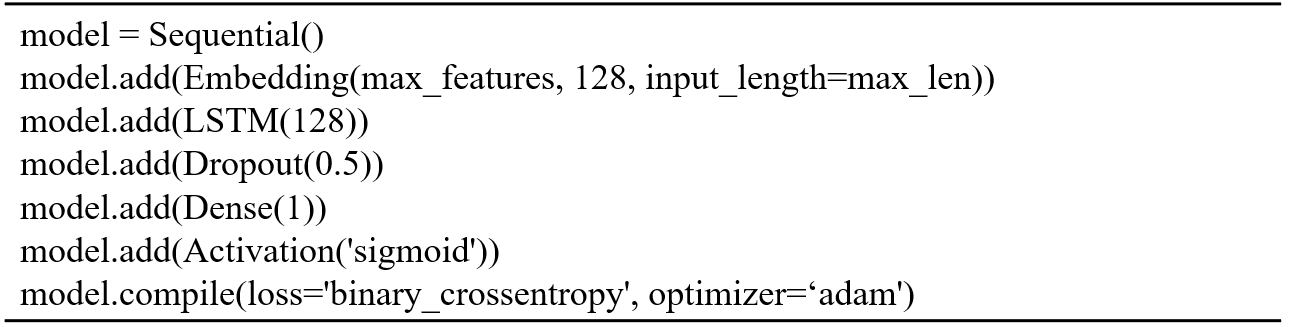
|  |  |
| --- | --- |
|  | (5) |
|  | (6) |
|  | (7) |
|  | (8) |
|  | (9) |
|  | (10) |

其中公式5、6、7分别表示遗忘门、输入门和输出门的计算公式。公式8、9、10分别表示临时细胞状态、当前细胞状态和当前隐藏层状态的计算过程。

对于输入信息和，首先利用公式2.9计算遗忘门对历史信息的遗忘程度，其中表示Sigmoid函数，取值范围是[0-1]，其中0表示完全忘记，1表示全部保留。然后利用公式2.10计算输入门对信息的保留程度。接下来，根据遗忘门、输入门、临时细胞状态以及上一时间步细胞状态计算当前细胞状态。最后根据、和计算输出门以及当前隐藏层状态，其中决定了输出信息。

LSTM利用内部的门控单元对信息进行了选择性的删除和保留，扩充了神经网络的记忆存储容量，使得LSTM更加适合学习序列中的长期依赖关系。目前LSTM已经在语言建模（Language Model）、语音识别（Speech Recognition）和DGA检测等方面有了广泛的应用。

以下为LSTM神经网络训练的关键代码：



## 基于可信度算法的多模型协同检测模块

### 模块描述

在本系统的多模型协同检测模块中，服务器程序预先读取离线训练得到的模型，在收到域名检测请求后，针对不同模型，将相应的特征工程应用于待测域名，输入分类器后得到多个预测结果及分类器以及各模型根据统计学习中的可信度算法给出的恶意概率，如图 12所示，以典型的四个算法为例，给出多模型协同检测模块的主要流程。

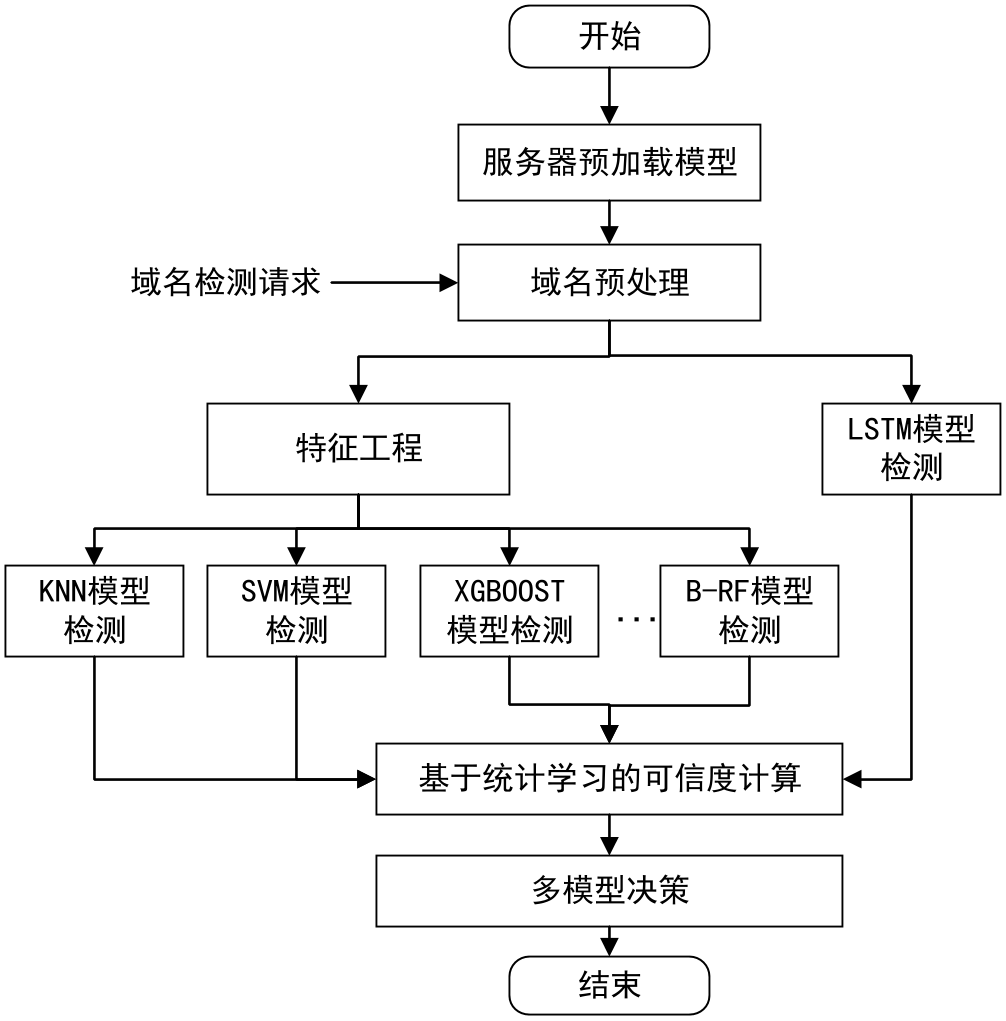


图 12 基于可信度算法的多模型协同检测模块主要流程

### 基于统计学习的可信度计算

在机器学习的分类问题中，一般是通过一个有限的训练样本集来训练模型。而这种样本除了有对象以外，还会包括对象所对应的标签。当给定一个新的样本时，机器学习的目标就是利用之前训练好的模型预测出新样本的标签。然而，随着近年来数据的爆炸式增长以及机器学习的普及应用，单一的分类结果在一些高风险领域（如各种故障检测、医疗诊断和金融评估等。）已经完全得不到满足。这些领域的人们可能更想要了解算法对于新数据的预测在多大程度上接近真实值。

一致性预测（Conformal Prediction）和韦恩预测（Venn Prediction）是近几年来在机器学习领域中提出的可以对预测结果进行可靠性评估和保障的方法。本节接下来将简单描述Conformal Prediction及其相关概念知识。

1. **显著性水平**

在机器学习的分类问题中，我们假设样本集中的对象为，，标签为，，则定义样本空间，为样本空间中的元素。一般地，当给出一个新的对象 时，通过这样一个方程，机器学习就可以简单预测出它的标签结果为：

|  |  |
| --- | --- |
|  | (11) |

一致性预测就是评估的每一个潜在标签的可信程度，然后输出所有标签中可信程度超过预设显著性水平的标签集合。于是就额外增添了一个名为显著性水平（Significance Level）的参数，名为置信度水平（Confidence Level），能够反映出对预测结果的信任程度。即一致性预测会输出标签空间Y的一个子集：

|  |  |
| --- | --- |
|  | (12) |

对于任意的，该子集满足如下公式，这表明对于不同的，预测的区域是标签空间Y的嵌套子集。

|  |  |
| --- | --- |
|  | (13) |

由此公式，我们也可明显得知：当显著性水平比较小，即置信度水平比较高的时候，子集会是一个比较大的区间。

所以，我们就可以通过修改的值来调节预测区间的大小。只要错误的预测区间比例不超过预设的显著性水平，就认为一致性预测可靠性的评估是有效的。即用户能够自由定义可容忍的最大预测出错概率。

但是在保证预测结果有效（置信度水平高）的同时，我们也希望预测区间内标签的数量少一点。然而，可容忍的最大出错概率越小的话，又会导致比较大的预测区间。因此我们通常需要根据实际应用场景来弹性平衡两者的关系。

1. **一致性预测**

一致性预测的核心就是一致性度量方法，即用来测量新样本和已知样本序列的相似程度。

首先，我们将规模为的样本集合定义为，又叫做数据包。然后，假设一致性度量函数为A，则可以为样本序列中的每个样本映射一个数值：

|  |  |
| --- | --- |
|  | (14) |

其中，是一个多重集合。指一个特定的算法，大多数是基于已知的机器学习算法。并且，数据包中的元素可以任意交换位置，而不影响的值。即表示样本的映射值与公式14中数据包内样本排列的顺序是无关的。就代表样本与其数据包的不一致性程度。

因为不同的算法规则不同，最后的得分也不同。所以单就不同对应的值，是无意义的比较，也丝毫没有可比性的。于是，为了能够直观地比较新样本与不同对应的样本集合之间的相似程度，才出现了p-value的概念。

我们假设，则对于样本序列，由映射可以得到一个数值序列：，然后通过比较与其他的，计算p-value，如公式15。

|  |  |
| --- | --- |
|  | (15) |

p-value的数值范围为之间。对于每一个可能的标签y，都能计算得到一个p-value，即。如果很小，则表示在原数据包中，其他样本的不一致性得分大部分要比低，这就意味着与原数据包的不一致性程度比较高。反之亦然。同时，也表示假设标签y成为的真实类别的显著程度。

最后，在预先给定的显著性水平下，以方法为核心的一致性预测输出的预测结果为：

|  |  |
| --- | --- |
|  | (16) |

综上，一致性预测的结果明显是嵌套的。

一致性预测的有效性是指预测事件错误的概率小于等于显著性水平：

|  |  |
| --- | --- |
|  | (17) |

1. **可靠性评估**

由上描述可知，一致性预测是通过调整显著性水平，进而控制整体预测的错误率来保障整体预测结果的可靠性。而对于个体预测结果的可靠性评估，则需要通过更加量化的两个指标：Confidence与Credibility来实现。

在分类问题中，Credibility为最大的p-value，Confidence为1减去第二大的p-value。以二分类问题为例：当一个新样本的时，，表示新样本拒绝标签“1”而选择最大的对应的标签0作为预测结果时的可信程度；，表示选择最大的对应的标签0作为预测结果时，该样本与样本序列中标签为0的样本的一致性程度。

通俗来讲，Confidence代表一个样本与某样本集合最不一致的程度，即我有多大的把握认为其不属于某个样本集合；而Credibility则代表与某样本集合最一致的程度，即我有多大的把握认为其属于某样本集合。如果某样本的预测结果对应的Confidence接近1而Credibility不接近0，我们就认为这一预测结果是可靠的。如果Credibility接近0则说明这个新样本很可能是个离群点，与训练集中的其他样本都不一致。

一致性预测的在线模式（Online Mode）与离线模式的区别在于是否将新样本以及对应的正确标签加入原始样本集，从而进行不断地更新。不过在实际应用中，根据固定样本集生成预测结果的离线模式已经能够满足需求。

以下为计算p-value的关键代码部分：

### 多模型决策

确定基础模型的预测标签后，构成了多模型预测标签集合。接下来对此标签集合进行统计和分析，并根据以下策略产生多模型决策结果。

1. 如果所有基础模型的标签结果一致，且标签为1或0，这表示所有基础模型都以高置信水平产生了相同的预测标签，因此直接输出该作为最终决策结果；
2. 如果所有基础模型的标签结果一致，且标签为“可疑”，说明所有基础模型都认为该样本与所有类别的相似程度不满足预设显著水平。针对这种情况，基于P值计算每一个模型对于该样本预测结果的可信度和置信度，综合考虑样本与预测类别的相似程度以及样本与其他类别的差异程度，全面评估基础模型的预测质量。由第二章第四节内容可知，当可信度和置信度都最大时，模型的预测质量最高，因此本文利用乘积的方式（）选择预测质量最高的基础模型，并将该模型的预测标签作为最终决策结果；
3. 如果所有基础模型的标签结果不一致，则采用投票的方式产生最终检测结果（“可疑”不计入其中）。如果出现票数相等的情况，综合考虑Confidence和Credibility择优产生最终决策结果。

综上所述，在基于统计学习的多模型决策部分，本文根据样本与训练集中每一类样本子集的拟合程度度量样本的相似度，利用人工预设的显著水平对基础模型预测结果进行二次筛选和细分（0、1、可疑），确定基础模型预测标签。通过统计分析基础模型标签集合，缩小“可疑”类范围，根据投票、可信度和置信度等多种策略产生最终决策结果。基于统计学习的多模型决策方法提高了基础模型检测结果的可靠性和准确度，并且在多模型决策时有效识别并淘汰无效检测结果，最大程度的保证了算法的有效性和鲁棒性。

## API扩展模块

本系统利用配置文件维护依赖文件信息，包括了模型路径、输出文件路径等，当用户需要添加模型时，只需要在服务器端的模型文件夹中添加模型文件，并同步修改配置文件，就可以实现系统模型扩展，保证了系统的可扩展性。

### 接口设计

本系统通过get\_result接口实现前后端数据的交互，接口涉及的交互数据如表所示。

表 7 get\_result接口信息表

|  |  |
| --- | --- |
| **Get\_result接口** | |
| 上送字段 | |
| 字段名 | 描述 |
| Input\_val | 输入的域名字符串 |
|  | |
| 返回字段 | |
| 字段名 | 描述 |
| SVM | SVM模型检测结果、恶意概率、P值 |
| BRF | BRF模型检测结果、恶意概率、P值 |
| XGBOOST | XGBOOST模型检测结果、恶意概率、P值 |
| CNN | CNN模型检测结果、恶意概率、P值 |
| LSTM | LSTM模型检测结果、恶意概率、P值 |
| RESULT | 最终的检测结果 |

## 数据库设计说明

为了确保数据库系统能够有效地满足用户需求，并且为数据库的开发、实施和维护提供详细的指导和参考对象,特编写此说明书。

数据库只写数据库的设计。样本的来源和存储是另外一个模块。

#### 3.7.1 数据库设计框架

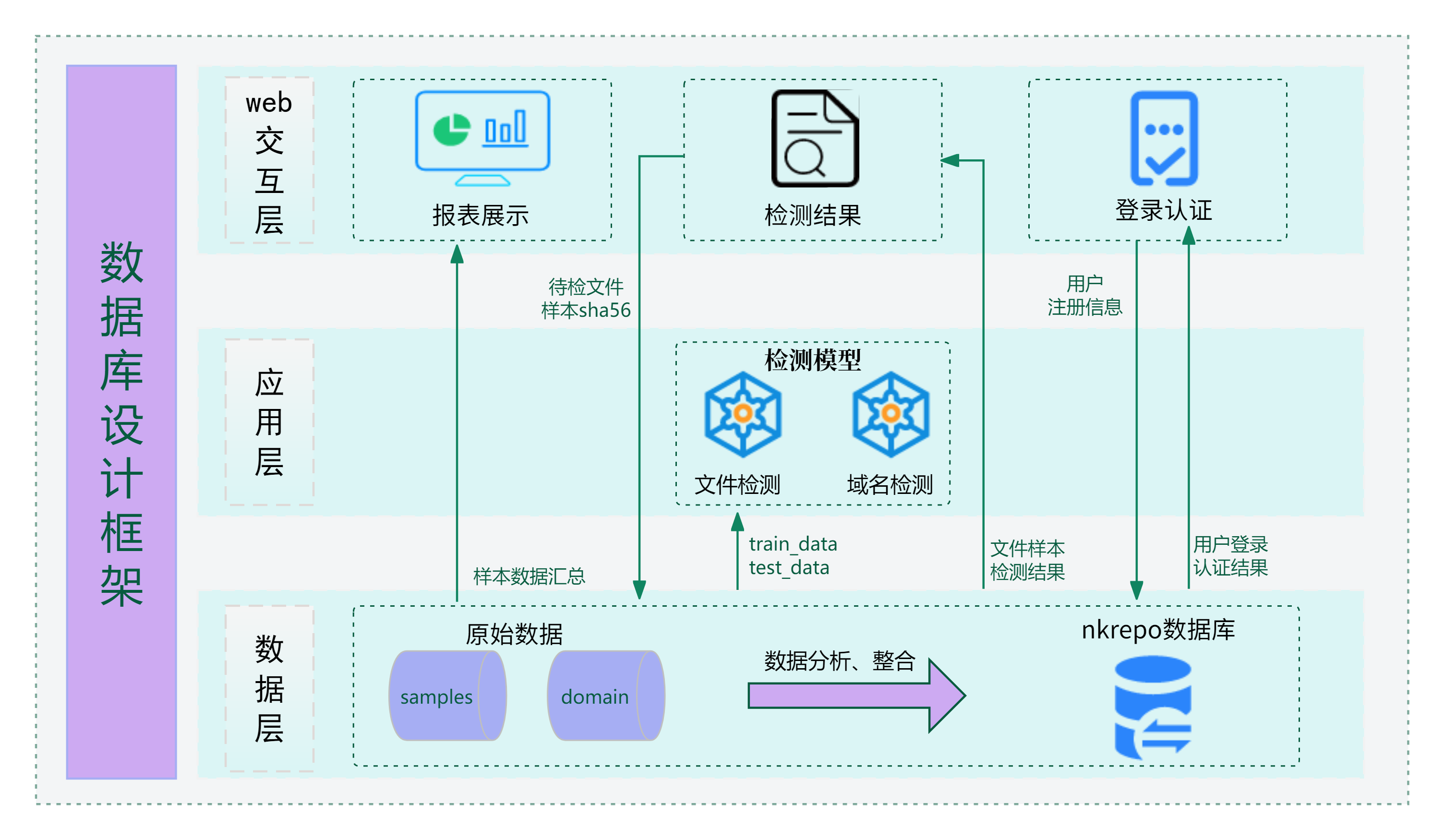


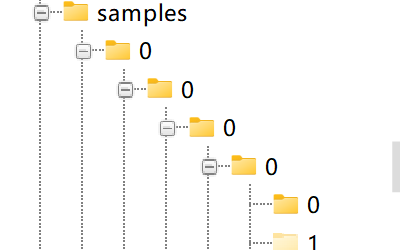
图 5-3-1 数据库设计框架图

如上图所示，数据库设计框架共包含三部分，其中数据层为本章节核心部分，包含原始数据获取、数据清洗、数据分析、数据整合等，并将整合好数据更新至nkrepo数据库中，用于系统开发；数据层与应用层的交互为数据层为应用层模型训练提供训练数据及测试数据；数据层与web交互层的交互分为三部分，一是报表展示，二是检测结果，三是登录认证。

#### 3.7.2 数据库各部分详细介绍

**1. 数据层：**为数据库框架中核心部分，该部分包含系统整体架构中“数据来源层、数据收集层、数据处理层及数据存储层”。以下将通过数据层介绍及nkrepo数据库介绍进行说明。

**（1）数据层介绍：**本部分数据来源于两部分，其中恶意域名样本通过maltrail进行收集，恶意文件样本通过VirusShare官网进行收集，通过VirusTotal和Kaspersky等提取扫描结果，收集到的样本存储在服务器的"./nkrepo/data"中，其中恶意域名样本存储在“trail”文件夹中，恶意文件样本存储在“samples”5层文件夹中（根据样本sha256前5位进行5层存储）。通过不同方式获取的数据进行分析整合，并更新至MySQL数据库nkrepo中。



**（2）nkrepo数据库介绍：**用于恶意域名检测系统开发的数据储存在nkrepo数据库中，该数据库共包含3类数据,文件样本（samples）、域名样本（domains)、用户信息（users)，说明如下。

**①samples:**为文件样本数据，包含恶意文件样本和良性文件样本。数据库中共包含256个数据表，数据表命名格式统一为“sample\_xy”，其中“xy”为文件样本sha256前2位，即将sha256值前2位相同的文件信息储存在同一个“sample\_xy”数据表中。数据表中所包含字段如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 说明 | 格式 |
| id | 索引 | —— | int（auto\_increment primary key） |
| name | 样本名称 | 命名格式“VirusShare\_MD5” | varchar(255) |
| md5 | 样本md5值 | 样本md5值 | varchar(255) |
| sha256 | 样本sha256值 | 1. 样本sha256值 2. 文件样本及VirusTotal扫描结果存储在“./nkrepo/data/samples”文件夹中，根据样本sha256前5位置进行5层存储。 | varchar(255) |
| src\_file | 样本来源 | 取自压缩包的名称：VirusShare\_XXXXX同“virusshare官网” | varchar(255) |
| date | 样本更新时间 | 样本更新时间，同“virusshare官网” | varchar(255) |
| category | 样本分类 | 卡巴斯基检测到的威胁或事件的分类 | varchar(255) |
| platform | 样本影响平台 | 卡巴斯基检测恶意样本影响的操作系统或设备的平台 | varchar(255) |
| family | 样本家族分类 | 卡巴斯基检测恶意样本家族分类 | varchar(255) |
| result | 样本总结 | 1. 恶意样本：卡巴斯基检测到恶意样本总结或分类； 2. 良性样本：标记为“clean” | varchar(255) |
| filetype | 样本的类型 | 描述恶意样本的文件类型，如：.exe、.dll、.scr、.bat等 | varchar(255) |
| packer | 样本的packer | 样本的packer | varchar(255) |
| ssdeep | 样本的模糊哈希值 | 用于识别潜在的恶意软件变种、查找相似的文件、判断两个文件是否同源等 | varchar(255) |

**②domains:恶意url数据**，截止目前，数据库中共包含641个数据表，各数据表以日期命名，格式为“domain\_yyyymmdd”，数据表中所包含字段如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 说明 | 格式 |
| id | 索引 | —— | int（auto\_increment primary key） |
| name | 名称 | 为恶意url域名 | varchar(255) |
| category | 类型 | 恶意域名类型 | varchar(255) |
| source | 来源 | 恶意域名来源 | varchar(255) |

**③users:用户信息**，用于存储用户注册信息，数据库中共包含1个数据表，各数据表以“user”，数据表中所包含字段如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 说明 | 格式 |
| id | 索引 | —— | int（auto\_increment primary key） |
| username | 用户名 | 用户注册时填写的用户名 | varchar(80) |
| password\_hash | 用户密码哈希值 | 为保护用户隐私及安全，用户密码采用哈希存储方式 | varchar(128) |
| is\_active | 用户账号状态 | “1”：正常用户，密码验证通过后可登录  “0”：异常用户，密码验证通过后仍无法登录，用于异常用户禁用 | tinyint(1) |

1. **数据层与其他层之间的交互：**

**（1）数据层与应用层：**在模型训练时，数据层为应用层提供训练数据和测试数据集。

**（2）数据层与web交互层：该部分分为三部分，如下所示：**

**①报表展示：**数据层为web报表展示页面提供所需数据。

**②检测结果：**在文件样本检测时，web交互层通过向数据层传送样本的sha256值，获取样本文件的基本信息，包含“category、platform、family、result、platform”等。

**③登录认证**：用户注册时，web交互层向数据层传送用户注册信息（username，password\_hash)，数据层存储用户信息；用户登录时，web交互层向数据层传送用户的username，查询与提交的用户名匹配的用户记录，找到用户后，获取password\_hash，进行密码验证。

1. https://www.spamhaus.org/statistics/tlds/ [↑](#footnote-ref-0)
2. https://pypi.org/project/wordfreq/ [↑](#footnote-ref-1)