基于机器学习的恶意 URL 检测策略研究

赵文天

云盾智慧科技有限公司，北京，100032；

摘要：恶意 URL 作为网络攻击的常用载体，呈现出结构灵活、隐蔽性强、演变速度快的特点，给传统检测手段带来持续挑战。机器学习因其模式识别能力，在恶意 URL 识别中逐步取代基于规则的机制。文章从攻击特征出发，结合真实样本构建方式，提出一套以静态结构、序列语义、多模态行为为核心的特征提取体系，适配逻辑回归、树模型及神经网络等多类分类器。通过模型评估指标体系与部署路径，讨论检测精度、响应时效、可解释性之间的平衡策略，同时对对抗样本、概念漂移、多源融合和法律边界问题给出实用建议。

关键词：恶意 URL；机器学习；特征工程；对抗样本；实时检测

DOI：10.69979/3041-0673.26.03.022

引言

随着全球信息技术的广泛使用，互联网不仅为人们带来了便利和高效的服务，也伴随着各种安全问题。大量的个人和企业敏感数据在网络上容易被访问，这为网络犯罪活动创造了机会。统一资源定位符（Uniform Resource Locator，URL）作为访问互联网服务的关键途径，成为网络攻击的目标，承受着众多的安全威胁。因此，识别和过滤网络中的恶意 URL 对于保护网络安全至关重要，是确保网络环境健康的关键措施[1]。

1 理论基础与关键概念

1.1 恶意 URL 定义与典型攻击场景

恶意 URL 指向表面上看似正常的网页入口，实际承载窃取数据、植入程序、操纵终端等隐蔽意图，其危险性往往藏在跳转链和脚本细节中。典型场景包括伪装成银行或平台登录页面的钓鱼链接，捆绑下载木马的网盘短链，以及在后台与控制服务器保持心跳的指令通道。部分链接混杂灰色推广内容，使合规边界出现模糊[2]。准确把握恶意 URL 的内涵与外延，有助于在机器学习建模时选取贴近攻击逻辑的特征维度，而不被表象页面风格牵着走。

1.2 机器学习在网络安全中的应用趋势

机器学习进入网络安全领域以后，威胁识别逐步从经验驱动转向数据驱动，恶意 URL 检测正是变化较快的一条应用路径。早期系统主要依赖规则引擎和人工维护名单，面对攻击频繁变形往往反应迟缓。近年分类模型能够在大规模样本中识别复杂模式，在未知域名、变种路径和混淆字符面前维持较高识别率[3]。总体趋势显示，安全系统从依赖单一算法走向集成框架，将静态属性、行为迹象和上下文环境合并为一套可持续演进的智能防护能力。

1.3 URL 检测任务的形式化表达

在形式层面，恶意 URL 检测可以视作典型二分类任务，将每个链接映射为特征向量，交给分类器给出是否危险的判断。输入既包括长度、词形、符号分布等静态属性，也可以融合域名信誉、访问频度、脚本结构等侧面信息，从而在特征空间内拉开正常链接与高危链接的距离。输出通常采用概率分值形式，再结合业务场景设定不同阈值，用于告警、阻断或人工复核[4]。这样一套形式化表达，为后续模型选择、性能评估和系统落地提供统一框架。

1.4 技术挑战分析

围绕恶意 URL 构建自动识别体系时，技术障碍远比直观印象复杂。攻击者刻意采用短链、多级跳转、同形字符等手段干扰特征提取，导致单一维度难以稳定刻画风险模式，网络环境持续演化，旧样本中抽取的统计规律在新流量中逐步失效，模型出现性能衰减，安全设备资源有限，高精度算法往往计算开销偏大，实时检测压力突出。要在准确度、时延、可解释性之间找到平衡点，需要引入更精细的特征设计思路以及渐进更新的模型管理机制[5]。

2 特征工程与数据准备

2.1 数据采集与标注流程

恶意URL检测的精准度高度依赖于样本数据的完整性与标签质量。常见的数据来源包括公开威胁情报平台、恶意链接黑名单、以及企业自身积累的访问日志系统。前者提供高危样本的集中入口，后者则能更贴近业务场景，捕捉实际威胁形态。两类数据交叉构建，可提高覆盖广度与标签可信度。采集后，需对 URL 进行去重、统一主域表示、剔除无效结构，构造结构稳定的数据表。标注阶段采用二值编码，结合自动规则筛选与人工复核机制，确保高风险样本识别不遗漏、低风险误判不泛滥。为检验标注一致性，建议设定比例抽样回查，对标签漂

移进行定期修正，防止误差在模型训练中被放大。样本的构建不仅是数据准备的起点，也决定了后续建模的能力边界。

2.2 URL 静态特征提取

URL 本身即为一个结构规整的符号序列，其每一部分都蕴含潜在风险特征。以协议、主域名、路径、查询参数为基础，可细分出如域名长度、子域个数、路径深度、参数数目、符号使用频率等统计项。实践中常借助 Python 中的 urlparse 模块进行结构化拆解，再配合正则表达式识别是否存在裸 IP、混淆符号或敏感片段。进一步可引入字符熵、重复字符比例等统计学指标，捕捉自动生成 URL 所特有的分布异常。这些静态特征提取成本低、可解释性强，适用于大规模初筛与快速评分。但若缺乏针对性抽象，极易堆积冗余维度，造成信息稀释。高效的静态特征体系应在保持轻量结构的同时，兼顾对攻击语义的感知能力。

2.3 URL 序列与语义特征提取

将 URL 理解为一段字符或词语的序列，有助于模型从上下文关系中识别攻击模式。字符级方法通过构建定长向量，将每个字符映射至索引后送入卷积网络或循环网络，学习局部组合与位置相关性；实现上多使用字典编码与补零对齐策略。词级方法则更侧重语义，利用斜杠、下划线、数字边界等拆解规则提取出高频敏感词，如“login”“verify”“account”，这些词在钓鱼场景中频现，具有较高的指示性。近年来，预训练语言模型被引入 URL 表达中，捕捉字符间的上下文依赖与逻辑关系，显著提升模型对混淆与变形的鲁棒性。尽管深层语义建模带来识别精度提升，但模型参数激增、计算资源消耗也同步上升。因此，在序列建模中，如何控制编码维度与训练代价之间的张力，成为落地前必须面对的抉择。

2.4 特征清洗与选择

模型性能与特征质量呈强相关，初始特征集若不经过严谨清洗，极易在训练中放大噪声效应。常规清洗流程包括处理缺失值、异常值剔除、统一类型尺度与规范编码格式。在数值型特征上，需进行标准化或归一化处理；在类别特征上，则按数据稠密性选择独热编码或目标编码。完成预处理后，进入特征选择阶段。基于树模型的特征重要性排序、相关系数分析、递归特征消除等方法，可有效筛除冗余变量与共线因子，防止模型陷入无效学习路径。尤其需警惕信息泄露，如部分手工标签字段或后验生成字段必须彻底清除，以防模型训练过程中误学判断结果本身。最终特征集应在维度精简、表达充分与训练友好之间保持张力平衡，形成支持泛化能力的特征体系。

3 机器学习模型设计与实现

3.1 模型选择与参数讨论

模型选用应紧贴特征类型和运行环境，避免一味追求复杂架构。静态特征维度相对规整时，逻辑回归适合作为基线，用于给出清晰的线性分界；树模型如随机森林和梯度提升树在处理离散特征时表现稳健，对特征缩放要求较低，常用树数量、最大深度、最小样本分裂数等参数控制拟合程度。面对字符序列特征，卷积神经网络适合捕捉局部模式，长短期记忆网络则偏向时序依赖，二者需要结合 URL 最大长度、嵌入维度、层数和丢弃率做细致调节。恶意 URL 样本比例往往偏高或偏低，引入类权重、采样策略以及代价敏感损失函数，可缓解不平衡带来的偏斜，使模型输出更贴近防护需求。

3.2 训练流程与评价指标

训练流程通常从数据划分起步，常见做法是按时间或随机方式切分训练集和测试集，再在训练集内部进行交叉验证，防止偶然划分带来虚高表现。模型拟合阶段需要结合学习率、树深、正则化强度等关键参数进行网格搜索或贝叶斯搜索，减少人工反复试错。评价体系不宜只看整体准确率，在恶意 URL 占比偏低的情形下，这一指标容易产生错觉，精确率和召回率更加关键，前者反映告警质量，后者体现漏报风险，二者折中的 F1 值常被用作调参目标。实际安全场景还会关注接收者操作特性，采用 ROC 曲线和 AUC 指标观察不同阈值下的权衡，并结合业务侧容忍度设定告警分界点，使模型输出更易转化为可执行决策。

3.3 代码实现示例

在工程实践中，代码实现通常围绕一条清晰流水线展开：先实现 URL 特征抽取函数，再构造训练数据矩阵，随后封装模型和预处理步骤。以 Python 为例，可编写 extract_url_features(url)返回字典，之后用数据框工具将多条记录拼接成特征表，对数值字段做标准化，对类别字段做独热编码，最后交给学习器。一个常见写法是构建管道对象，将预处理模块和分类器顺序串联，例如 Pipeline([("preprocess",transformer),("clf",RandomForestClassifier(n_estimators=200,max_depth=20))])，再配合交叉验证接口完成拟合和评估。在线检测时，只需把离线训练得到的模型持久化到文件，业务系统在接收 URL 后调用同一特征函数，再喂入模型输出风险分值，整套流程保持接口一致，方便维护。

3.4 模型可解释性与部署考量

恶意 URL 检测一旦对接网关和终端，安全人员格外在意“为什么判为高危”。树模型可以给出特征重要性

排序，让人看到域名长度、可疑词出现次数等维度在决策中的权重；更精细的需求可以借助局部解释方法，对单条样本展示各特征对风险分值的正负贡献，使规则调整和误报分析更有抓手。部署层面还要面对时延和资源约束，深度模型放在核心链路可能引起明显延迟，因此常见做法是采用“轻量模型在前，复杂模型在后”的分级结构，前层快速筛除绝大部分低风险流量，后层集中处理少量可疑链接。模型更新频率同样关键，需要设计滚动训练和灰度发布机制，让新样本逐步进入体系，同时保留回滚通道，在安全和稳定之间取得相对平衡。

4 检测策略优化与实用建议

4.1 对抗样本与概念漂移应对

针对恶意 URL 的对抗样本，防守方若只依赖一次性建模，风险极高。攻击者常改动个别字符、插入无意义路径段或利用短链平台，使特征分布发生细小偏移，却足以诱导分类边界失真。应在训练阶段引入扰动样本，将高危链接按多种方式改写，再送入模型，使其逐步学会识别稳定模式而非表面符号。网络环境随时间演进，旧数据中提取的统计规律会逐渐失效，需要建设概念漂移监测机制，围绕告警率、阈值附近样本分布以及关键特征均值设定预警指标。一旦发现长期偏移，就以时间窗口重采样，开启增量训练或小规模重训，让模型始终停留在相对新鲜的威胁视野之中。

4.2 多模态融合特征策略

单看 URL 字符串，往往难以捕捉攻击全貌，多模态信息能显著提升刻画深度。除长度、词形这类静态特征以外，域名注册时间、解析结果、证书颁发机构、服务器地理位置、访问频次曲线等维度都值得纳入，同一链接在不同终端的访问路径和停留行为也能提供侧面印证。工程上可以先在各模态内构造子向量，再采用拼接或注意力加权方式形成统一表示，由集成模型完成最终判定。融合过程中要警惕一味堆叠维度，使模型陷入稀疏高维空间，比较稳妥的做法是先在每个模态内完成特征筛选，保留信息密度较高的少数指标。多模态策略一旦落地，恶意 URL 即使在字符串层面伪装得滴水不漏，也会在时序行为或基础设施画像上露出破绽。

4.3 实时检测架构建议

恶意 URL 检测一旦嵌入业务链路，时延约束立刻压在肩上，架构设计因此需要分层思路。入口节点宜配置轻量规则和黑白名单，对明显安全或明显高危链接迅速给出结论，中间区间交给机器学习引擎处理。核心检测层可以划分同步通道和异步通道，前者承担毫秒级判定，适合部署压缩后的树模型或浅层网络，后者接收小部分高风险样本，进入沙箱分析或复杂深度模型，在后台补充情报并反向更新策略。为避免单点压力，架构中还应设置本地缓存和结果回写机制，对重复链接直接复用历史判定，节约计算资源。整套系统一旦运转顺畅，就能在保持响应速度的前提下，实现较为细腻的风险分级处理。

4.4 数据隐私与法律合规

恶意 URL 检测依赖大规模日志和关联信息，一旦缺乏边界意识，很容易触碰隐私红线。设计数据管道时，应坚持最小必要原则，对用户标识采取脱敏处理，将账号、设备号和精确地理信息拆散，避免形成完整画像，对存储周期设置上限，过期日志按预定策略销毁。跨部门共享样本时，只保留与安全判断直接相关的字段，屏蔽内容正文和敏感业务信息，内部访问权限则交给专门角色管理。模型训练平台需要留下审计轨迹，记录数据来源、用途范围和访问记录，便于在出现纠纷时给出清晰说明。将这些约束视作系统设计的一部分，而不是额外负担，既能保护个体权益，也能为恶意 URL 检测体系赢得持续运转的信任基础。

5 结语

恶意 URL 检测的关键不在于算法选型本身，而在于特征建构的深度与策略更新的节奏。只有将识别逻辑嵌入实际攻防场景，在准确率、处理时延与合规性之间做出有意识的取舍，检测系统才能长期稳定运行。机器学习的意义，不止是提升识别率，更是赋予系统面对未知风险的演化能力。未来的防护体系，应向多源感知、边缘推理和可解释预测等方向延展，实现从“规则阻断”到“认知驱动”的转型。

参考文献

[1]赵世雄.基于深度学习的恶意 URL 检测[D].江苏科技大学,2024.
[2]杨胜杰,陈朝阳,徐逸,等.基于深度学习与特征融合的恶意网页识别方法研究[J].信息安全学报,2024,9(3):176-190.
[3]吴森焱,罗熹,王伟平,等.融合多种特征的恶意 URL 检测方法[J].软件学报,2021,32(9):2916-2934.
[4]顾傲.基于机器学习的恶意 URL 识别[D].阜阳师范大学,2023.
[5]盛蒙蒙,史建晖,沈立峰.基于 CBA 算法的恶意 URL 检测[J].数字技术与应用,2023,41(10):9-13+60.