计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2020,29(11):11−20 [doi: 10.15888/j.cnki.csa.007461]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于半监督学习的恶意 URL 检测方法①

麻瓯勃, 刘雪娇, 唐旭栋, 周宇轩, 胡亦承

(杭州师范大学 杭州国际服务工程学院, 杭州 311121)
通讯作者: 刘雪娇, E-mail: liuxuejiao0406@163.com

摘　要: 检测恶意 URL 对防御网络攻击有着重要意义. 针对有监督学习需要大量有标签样本这一问题, 本文采用半监督学习方式训练恶意 URL 检测模型, 减少了为数据打标签带来的成本开销. 在传统半监督学习协同训练 (co-training) 的基础上进行了算法改进, 利用专家知识与 Doc2Vec 两种方法预处理的数据训练两个分类器, 筛选两个分类器预测结果相同且置信度高的数据打上伪标签 (pseudo-labeled) 后用于分类器继续学习. 实验结果表明, 本文方法只用 0.67% 的有标签数据即可训练出检测精确度 (precision) 分别达到 99.42% 和 95.23% 的两个不同类型分类器, 与有监督学习性能相近, 比自训练与协同训练表现更优异.

关键词: 恶意 URL 检测; 半监督学习; 协同训练改进算法; Doc2Vec; 分类器训练

引用格式: 麻瓯勃,刘雪娇,唐旭栋,周宇轩,胡亦承.基于半监督学习的恶意 URL 检测方法.计算机系统应用,2020,29(11):11–20. http://www.c-s-a.org.cn/1003-3254/7461.html

Malicious URL Detection Based on Semi-Supervised Learning

MA Ou-Bo, LIU Xue-Jiao, TANG Xu-Dong, ZHOU Yu-Xuan, HU Yi-Cheng

(Hangzhou Institute of Service Engineering, Hangzhou Normal University, Hangzhou 311121, China)

Abstract: Detecting malicious URL is important for defending against cyber attacks. In view of the problem that supervised learning requires a large number of labeled samples, this study uses a semi-supervised learning method to train malicious URL detection models, which reduces the cost overhead of labeling data. We propose an improved algorithm based on the traditional co-training. Two kinds of classifiers are trained by using expert knowledge and Doc2Vec pre-processed data, and the data with the same prediction result and the high confidence of the two classifiers are screened and used for classifiers learning after being pseudo-labeled. The experimental results show that the proposed method can train two different types of classifiers with detection precision of 99.42% and 95.23% with only 0.67% of labeled data, which is similar to supervised learning performance and performs better than self-training and co-training.

Key words: malicious URL detection; semi-supervised learning; co-training improvement algorithm; Doc2Vec; classifier training

万维网是人们接入互联网的主要入口, 用户能够通过 URL (统一资源定位符) 直接或间接地获取互联网上的各类信息. 在这种方式给生活带来便利的同时, 攻击者也可利用恶意 URL 实现不同类型的攻击. 据 2018 年卡巴斯基安全公告[1] 中统计, 2017 年 11 月至 2018 年 10 月间, 该实验室 Web 防病毒组件共识别 554 159 621 个恶意 URL, 而其在 2014 年的安全公告中统计通过恶意 URL 实现的攻击占当年整个网络攻击的 75.76%,

① 基金项目: 浙江省自然科学基金 (LY19F020021); 浙江省大学生科技创新活动计划 (新苗人才计划) (2019R426035)
Foundation item: Natural Science Foundation of Zhejiang Province (LY19F020021); Graduates Scientific and Technologic Innovation Program of Zhejiang Province (Young Talents Program) (2019R426035)
收稿时间: 2019-11-18; 修改时间: 2019-12-11; 采用时间: 2019-12-25; csa 在线出版时间: 2020-10-29

这一数值在 2015 年也达到了 73.70%. 因此, 检测恶意 URL 成为了应对网络攻击的重要组成部分.

恶意 URL 是指欺骗用户访问, 达到执行恶意行为或非法窃取用户数据目的的 URL. 攻击者在 URL 中嵌入恶意代码就可以实现 XSS、SQL 注入等攻击, 用户访问这些 URL 会被窃取个人隐私信息, 例如账号密码、个人资料, 或者被迫下载和执行恶意程序或脚本(例如病毒、木马、蠕虫等)[2]. 为了防止被检测系统拦截, 攻击者不断设计新型恶意 URL, 如何及时有效应对这些恶意 URL 成为了一大挑战.

目前恶意 URL 检测主要基于黑名单 (blacklisting) 和规则库[3], 这种方式实现简单、检测高效, 但却难以应对新型恶意 URL. 机器学习已经在入侵检测领域有了广泛应用, 它可以一定程度解决未知攻击难以检测的问题, 所以有一些研究[4–10] 将其应用在恶意 URL 检测上. 应用机器学习实现恶意 URL 检测遇到的主要困难是: 与丰富的攻击手段相似, 网络中数据的高复杂性使得统计特征有较大的可变性[11]. 这导致无监督学习训练出的检测模型虽然能够判断出一些未知的恶意 URL, 但模型若没有高度可靠, 则易出现误报[12]. 许多检测系统每日报警数可达到几十万次, 人为从所有报警中排错是一件极其困难的事情, 所以低误报率的检测系统更具实用性. 由于存在以上问题, 结合有监督学习训练恶意 URL 检测模型是目前的主要应用方式. 但有监督学习训练检测模型需要大量有标签样本, 为样本打标签将增加成本开销并消耗更多时间.

本文结合半监督学习训练恶意 URL 检测模型, 改进了协同训练算法, 只需用少量有标签数据和大量无标签数据即可实现两个不同分类器的相互学习和共同进步. 此外, 在数据预处理中, 除了基于专家知识外, 还引用基于统计的 Doc2Vec 工具将 URL 作为带情感文本处理, 这种方式考虑了词序, 保留上下文联系, 有助于训练的分类器区分恶意 URL 与正常 URL.

1 相关工作

1.1 恶意 URL 检测

黑名单是检测恶意 URL 最常用的方法, 其本质是过去已被确认为恶意 URL 的数据库. 每当访问新 URL 时, 都会执行数据库查找. 如果该 URL 存在于黑名单中, 则被判定恶意, 系统生成警告, 否则判定为良性. 但是由于每天都会生成新的 URL, 维护一个详尽的恶意 URL 列表是不现实的. 文献 [3] 中研究显示, 为了逃避黑名单检测, 许多攻击者会对原始 URL 进行少量修改, 或通过混淆将 URL 修改为“看起来”合法的形式迷惑检测系统[13]. 因此, 黑名单方法具有严重的局限性, 绕过它们并不是一件困难的事情, 尤其是黑名单对于新生成的恶意 URL 缺乏检测能力[4]. 尽管黑名单面临着上述问题, 但由于其实现简单且查询效率高, 仍是当今恶意 URL 检测系统最常用的技术[14].

启发式 (heuristic) 方法是对黑名单的一种扩展, 主要思想是创建“签名黑名单”, 识别常见的攻击, 并将签名分配给该攻击类型. 检测系统可以在网页上扫描此类签名, 并且在发现某些可疑行为时发出标记, 这种方法可以检测新 URL 中的威胁, 比黑名单具有更好的泛化能力. 但是, 启发式方法只能用于有限数量的常见威胁, 不能推广到所有类型的新型攻击, 且攻击者使用混淆技术依然可以绕过[4].

为了提高恶意 URL 检测器的通用性, 近年来对其与机器学习结合的研究日益受到关注. 机器学习方法可基于统计属性, 训练得到的分类器可以区分 URL 为恶意或良性, 能够一定程度检测新型恶意 URL. 支持向量机 (SVM) 是监督式学习方法之一, 在恶意 URL 检测中有着较多的应用[15–17], 它基于结构风险最小化原则避免了过学习问题, 泛化能力强, 但应对大规模训练样本时存在计算量过大、训练时间长的问题. 逻辑回归是恶意 URL 检测中另一种常用的监督式学习方法[18,19], 实现简单, 计算量小, 训练速度快, 但容易欠拟合, 得到的模型预测准确率相比其它方法不高. 文献 [15,18] 还介绍了朴素贝叶斯 (naive Bayes) 与决策树 (decision trees) 在恶意 URL 检测中的应用.

1.2 半监督学习

有监督学习需要大量有标签数据训练模型, 准确判断一条 URL 是否恶意需要丰富的专家知识, 这会造成时间开销的增加[20,21]. 并且当需要标记的 URL 数量过多时, 标记的准确性会受到影响. 由于新型恶意 URL 的产生速度快, 数量多, 只应用有监督学习训练检测模型显得低效. 半监督学习的引入就是为了一定程度解决上述问题. 随着网络应用的普及, 无标签数据的获取变得更为容易. 半监督学习因为可以借助大量的未标记数据来辅助少量的有标记数据提高训练模型的性能而受到关注[22,23], 其利用分类器代替人力进行数据标注, 并在此过程中不断学习提升自身区分数据类型的

能力. 半监督学习的基本依据在于: 数据的分布必然不是完全随机的, 通过一些有标签数据的局部特征, 以及更多没标签数据的整体分布, 就能得到可以接受甚至是非常好的分类结果. 这表明半监督学习训练的分类器性能不一定优于有监督学习, 而最终的训练效果与应用的有标签数据量有关, 这需要训练者根据目标进行抉择. 一些典型半监督学习算法的优劣势比较如表 1 所示.

表 1 典型半监督学习算法的优劣势比较

方法	参考文献	优势	劣势
Self-training	[24,25]	单视图实现, 操作简单, 计算量小	分类器错误累积现象严重, 造成性能下降
Co-training	[26–28]	实现简单, 计算量小, 分类器间可相互学习进行强化	训练得到分类器性能一般, 两个初始分类器性能需要相近
Co-regularization	[29–32]	算法实现多样, 得到分类器性能较好	对数据预处理要求高, 否则会较大程度影响性能
单视图协同训练变体	[33,34]	将协同训练与决策树算法相结合	对标记的置信度要求高, 不适用于少量标签情况
Tritraining	[35]	多分类器共同判断, 可以提高伪标签准确率	单视图造成多分类器间独立性不够

自训练 (self-training) 只需要一个分类器和少量有标签数据就可以实现, 核心思想是选择高置信度的未标记样本来扩充训练集, 存在的不足是如果无标签 URL 预测错误, 随着训练的进行会造成错误的累积[24,25]. 协同训练由 Blum 和 Mitchell 等[26,27] 提出, Nigam 等曾将其应用在文本处理中[28], 它需要两个不同分类器共同工作, 实现简单, 计算量小, 但若两个初始分类器性能不够接近, 弱分类器容易对强分类器产生较大负面影响. 协同正则法 (co-regularization)[29] 基于正则化框架, 试图直接最小化有标记样本上的错误率和两个视图上未标记样本的标记不一致性, 不涉及对未标记样本赋予伪标记的过程. 该方法有多种算法实现[30,31], 并可在信息论框架下解释工作原理[32]. 但其对于视图构建要求苛刻, 易出现高误报率, 不适合用于恶意 URL 检测. Goldman 和 Zhou[33] 提出了一种可用于单视图数据的协同训练法变体, 通过使用两种不同的决策树算法在相同属性集上生成两个不同的分类器, 然后按协同训练法的方式来进行分类器增强. 这种方法严重依赖 10 折交叉验证法[34] 估计标记置信度, 只适用于大量有标记样本的情况. Zhou 和 Li[35] 提出三体训练法 (tri-training), 该方法从单视图训练集中产生 3 个分类器后利用预测结果以“少数服从多数”的形式来挑选数据. 但由于只用了单视图, 分类器间的独立性不足, 相互学习效果不佳.

基于以上半监督学习算法遇到的问题, 本文提出一种以协同训练与自训练的思想为基础的半监督学习算法, 基于双视图, 结合两个分类器预测结果共同判断来提升标记伪标签的准确率, 依赖的原始打标签数据量少, 计算量较小.

2 恶意 URL 检测方案设计

本章将先设计基于协同训练的改进算法, 用于训练恶意 URL 检测分类器. 在 2.2 节介绍两个初始分类器的训练细节.

2.1 协同训练改进算法

图 1 给出了本文方法的训练流程. 我们先分别用基于专家知识和统计预处理过的少量带标签 URL 数据训练两个初始分类器, 用这两个初始分类器对剩余的无标签训练集进行预测, 设定预测结果良性为正例, 恶意为反例. 对于同一无标签训练集中的某条 URL, 只有两个分类器给出的预测结果相同才会通过第一轮筛选, 预测结果不相同则将该 URL 重新放入无标签训练集中等待下一轮预测.

在第二轮筛选中, 会将每条数据的两个分类器预测结果置信度求和, 以求出的置信度和作为标准由高到低排序, 且正例与反例数据分开处理. 本文方法在训练开始前会定义每轮挑选的正例数 p 和反例数 n, 即选出置信度和前 p 名的正例和前 n 名的反例. 将选出的共 $p+n$ 个数据以分类器预测结果打上伪标签, 加入有标签训练集. 应用这两个新训练集重新训练生成两个分类器, 即代表完成一轮类协同训练.

以上步骤循环执行, 直至某一轮训练中两个分类器共同判断的正例个数不足 p 个或者反例个数不足 n 个, 则跳出训练循环. 该阶段无标签训练集中的数据量较少, 若通过改变 p 和 n 的值继续训练, 则后续两个分类器的相同预测结果会不断减少, 导致每一轮都需要修改 p 与 n 的值, 降低了训练效率, 且由于每一轮打上伪标签的数据过少而使模型训练效果变化不明显.

基于以上原因, 本文提出在跳出原有循环后, 以协同训练算法继续工作, 重新设定一个后续不再改变且数值更小的 p 与 n, 直到所有的训练集都被打上伪标签为止. 基于上述流程, 算法 1 中给出了算法实现.

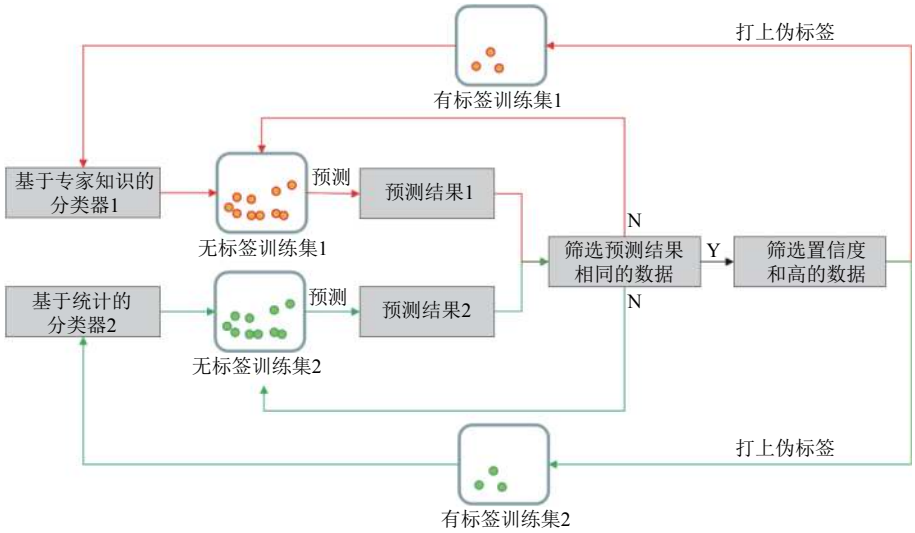

图 1 本文方法工作流

算法 1. 改进算法

输入:

有标记样本集: $D_l=\{(<x_1^1,x_1^2>,y_1),\cdots,(<x_l^1,x_l^2>,y_l)\}$;

无标记样本集: $D_u=\{(x_{l+1}^1,x_{l+1}^2),\cdots,(x_{l+u}^1,x_{l+u}^2)\}$;

每轮挑选的正例数 p;

每轮挑选的反例数 n;

预测正例数 q;

预测反例数 w;

基学习算法$\mathfrak{I}$.

流程:

```
初始化q=p, w=n;
for j=1,2 do
    D_l^j={(x_i^j,y_i)|(<x_i^j,x_i^{3-j}>,y_i)∈D_l};
end for
while q>=p且w>=n do
    for j=1,2 do
        h_j←𝔍(D_l^j);
        筛选h_1和h_2预测结果相同的URLs, 更新q和w;
        if q<p or w<n then
            break
        考察h_j在D_u^j={x_i^j|<x_i^j,x_i^{3-j}>∈D_s}上的分类置信度, 将两个分类器得到的预测分类置信度求和, 挑选p个正例置信度最高的样本D_p⊂D_s、n个反例置信度最高的样本D_n⊂D_s;
        由D_p^j生成伪标记正例: D_p^{3-j}={(x_i^{3-j},1)|x_i^j∈D_p^j};
        由D_n^j生成伪标记反例: D_n^{3-j}={(x_i^{3-j},0)|x_i^j∈D_n^j};
    end for
    if h_1,h_2均未发生改变 then
        break
    else
        for j=1,2 do
            D_l^j=D_l^j∪(D_p^j∪D_n^j);
        end for
    end if
end while
当q<p或者w<n时则按常规协同训练继续学习, 直到训练集全部打上伪标签且被用于训练分类器, 或者分类器不再变化.
```

输出: 分类器h_1,h_2

提供的有标签数据量越多, 则两个初始分类器 URL 恶意与否的区分能力越强. 这对算法执行的影响是, 在整体数据集数量相同的情况下, 初始分类器性能强代表判定 URL 类型正确的可能性更大, 则设置的每轮挑选正例数p与反例数n可相对较大, 整体的训练轮数较少. 初始分类器弱则相反.

2.2 数据预处理与两个初始分类器训练

本文设计构建两种视图用于分类器训练, 分别基于专家知识和统计. 在构建视图时, 若生成的视图不够充分, 学习过程会受到标记噪声和采样偏差的制约, 仅以学习器相互提供伪标记样本这种方式很难学得近似最优分类器. Wang 和 Zhou[36] 对此分析指出, 分类器在提供预测结果之外, 还可提供对预测结果置信度的估计, 则能在一定程度上缓解标记噪声和采样偏差的制约, 提升学习效果. 这表明我们得到的视图即使不充分, 基于分歧的半监督学习仍是可行的.

图 1 中两个分类器训练所需数据在预处理上并不相同. 前者需要应用专家知识, 通过已判别为恶意 URL

中的一些特殊规则来处理数据生成视图一, 并训练出分类器 1. 基于统计的方法将结合文本处理实现, 应用 Doc2Vec 工具预处理数据后生成视图二, 训练分类器 2. 由于两个分类器的预测结果要共同判断, 所以原始数据应该完全相同, 分别通过两种不同方式进行数据预处理, 并划分为训练集和测试集, 如图 2 所示. 对于得到的每份训练集需要进行第二次划分, 保留大部分没有标签的 URL 条目, 并将小部分人工打上标签, 用于初始分类器的训练, 如图 3 所示.

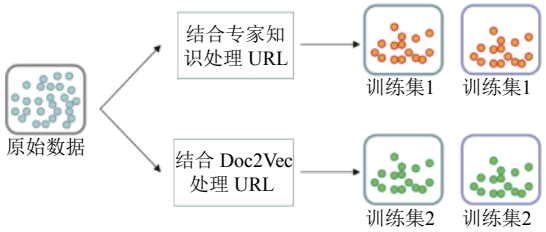

图 2 基于不同方法的数据预处理

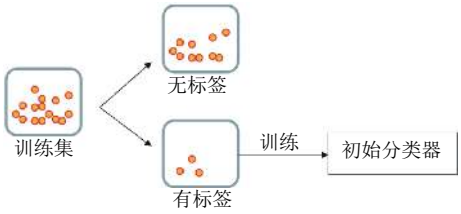

图 3 训练集中的数据划分

2.2.1 基于专家知识预处理 URL

视图 1 的构建主要依据 URL 中出现的特殊符号. 在 Canali 等的研究中发现[19], 恶意 URL 中的特殊符号数量普遍多于正常 URL, 以此为基础, 本文设计基于专家知识进行数据预处理后的特征由 3 部分组成. URL 与普通的英文字符串有所不同, 字符之间并非完全独立, 且其中会存在一些特殊字符. 某些特殊字符在正常请求的 URL 中也会出现, 但通常占据字符数少, 所以可以通过观察后从中选择有代表性的特殊字符作为第一部分特征. 另一些特殊字符或关键词在正常的 URL 中不会出现, 一旦检测发现则可以判断为恶意 URL, 这些特殊字符或关键词是特征的重要组成部分. 通过观察发现恶意 URL 的字符数在整体上要多于正常的 URL 字符数, 所以 URL 的总字符数将被作为最后一部分特征.

结合参考资料与所用数据集 URL 条目的特点, 最终确定作为特征的特殊符号与关键词为: “#”、“%”、“&”、“=”、“+”、“−”、“_”、“*”、“. ”、“or”、“NULL”, 计算每个 URL 条目对应特殊符号或者关键词的字符数, 加上 URL 的总字符数构成一共 12 个特征, 完成第一类数据预处理.

2.2.2 使用 Doc2Vec 预处理 URL

视图 2 的构建思想是将 URL 作为文本, 通过文本处理器实现向量化. Ma 等[37] 提出利用 URL 的词汇和主机信息特征训练分类器, 可以更快适应恶意 URL 不断变化的新特征. 徐冬冬等[38] 利用 TF-IDF 将 URL 作为文本处理实现向量化, 并以此训练模型来检测 SQL 注入攻击. 这两篇文献说明了将 URL 作为文本处理的可行性.

文本处理的方法很多, 表 2 给出了目前常用方法的对比. 目前通常使用 bag-of-words、average word vectors、TF-IDF、Word2Vec 等方式把数据投影到向量空间中. Bag-of-words 没有考虑词序 (word order), 且忽略了单词的语义信息; average word vectors 对句子中的所有词向量取平均, 但也没有考虑到单词的顺序; TF-IDF 是一种加权技术, 采用统计方法, 根据字词在文本中出现的次数和在整个语料中出现的文档频率来计算重要程度, 能过滤掉一些常见却无关紧要的词语, 保留影响整个文本的重要词语, 但依然没有考虑到词的顺序; Word2Vec 可以通过训练把对文本内容的处理简化为 K 维向量空间中的向量运算, 而向量空间上的相似度可以用来表示文本语义上的相似度. 在获得词向量后, 对词向量进行平均处理, 最终得到句子向量. Word2Vec 考虑了词序, 但忽视了上下文的联系, 没有对单词的顺序进行特殊处理, 在面对长段落时效果并不理想. 恶意 URL 普遍较长, 只用 Word2Vec 的方式来处理并不理想.

表 2 多种文本处理方式的对比

模型	是否考虑词频	是否考虑语义	是否考虑词序	是否考虑上下文
Bag-of-words	√	×	×	×
Average word vectors	√	√	×	×
TF-IDF	√	√	×	×
n-Gram	√	√	√	×
Word2Vec	√	√	√	×
Doc2Vec	√	√	√	√

为了更少损失文本中的重要信息, 本文中将应用 Doc2Vec 实现 URL 作为文本的处理. Doc2Vec 是 Word2Vec 的拓展, 目前在情感分析等问题上有着广泛

应用. Doc2Vec 不但生成词向量, 每个句子同样被映射到向量空间中, 可以用矩阵的一列来表示. 句向量能和词向量级联或者求平均得到特征, 预测句子中的下一个单词, 这实现了上下文的联系. Doc2Vec 考虑了文本的词频、语义和语序, 还能保留上下文关联信息[39–42]. 我们可将 URL 作为附带情感信息的文本处理, 利用 Doc2Vec 进行数据处理.

首先, 对 URL 条目进行分词, Doc2Vec 会把 URL 每个被分出的词与句都映射到向量空间, 将上下文的词向量与句向量级联或者求平均得到特征, 用于预测 URL 中下一个词. 给定如下训练序列, 目标函数是:

$$\frac{1}{T}\sum_{t=k}^{T-k}\log p(w_t|w_{t-k},\cdots,w_{t+k}) \tag{1}$$

其中, w 代表着 URL 中的词与句. URL 中的下一个词存在多种可能, 即这是一个多分类问题. 我们希望可能性大的词能经常取到, 但可能性小的偶尔也可以被选取, 所以分类器最后一层使用 Softmax 函数来给出各种可能性的评估, 计算公式为:

$$p(w_t|w_{t-k},\cdots,w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} \tag{2}$$

每一个y_i可以理解为预测出每个类别 i 的概率. 在该任务中, 每个词或句可以看成一个类别. 计算的公式为:

$$y = b + Uh(w_{t-k},\cdots,w_{t+k};W) \tag{3}$$

其中, U, b 是 Softmax 函数中的参数, h 由从 W 提取的词向量与句向量的级联或平均构成[41].

Doc2Vec 中有 PV-DM 和 PV-DBOW 两种语言模型[43,44]. 对比两种语言模型, PV-DM 预测行为的次数跟整个文本的词句数相近, 时间复杂度低, 速度快. PV-DBOW 则是通过更多的时间、计算开销来生成更精准的向量. 本文设计的方法基于半监督思想, 初始带标签的数据量基数小, 应用 PV-DBOW 开销并不大. 且由于初始数据量少, 若是用 PV-DM 会降低初始分类器的预测准确度, 在后续多轮学习中这一影响会被放大. 综上两点考虑, 选择 PV-DBOW 来处理向量.

3 实验与分析

3.1 数据处理

本次仿真实验所使用的数据集为西班牙国家研究委员会开发的 CSIC 2012. 该数据集由 Paros 和 W3AF 等工具生成, 异常请求包括 SQL 注入、缓冲区溢出、CRLF 注入、XSS、SSI 等, 涵盖的类别全面, 且使用的所有参数数据都从真实数据库中提取, 含有大量实际攻击数据. CSIC 2012 本身是针对 Web 攻击检测而诞生的, 只需要对该数据集进行一定的处理就能够较好满足恶意 URL 检测仿真实验的需求.

实验需要提取数据集中的 URL 部分, 保留其路径与参数信息. 通过去重、清洗等步骤保留下 20 441 条 URLs 用于本次实验, 划分为训练集以及测试集, 具体的组成情况如表 3 所示.

表 3 数据集划分与组成

数据	良性	恶意	总数据量
测试集	2190	3251	5441
训练集	5985	9015	15 000
总计	8175	12 266	20 441

分类器训练分为两部分, 第一部分基于有监督学习, 用全部打标签的训练集训练分类器. 第二部分基于少量有标签与大量无标签训练集进行的半监督学习. 半监督训练中包含自训练、协同训练以及本文的设计方法, 目的是进行更全面的分类器性能对比, 分析本文提出方法的可行性以及优势. 为了控制变量, 所有方法的训练集 URL 条目相同, 且全部采用训练量小且训练速度快的逻辑回归二分类算法作为基算法.

本次仿真实验中, 得到的所有数据本身已有标签. 如表 4 所示, 有监督学习的训练集保留全部标签, 即有标签数据量为 15 000, 而半监督学习初始只保留 100 个有标签数据.

表 4 不同训练方法的打标签数据量统计

训练方法	打标签数据数量
有监督学习	15 000
自训练	100
协同训练	100
本文方法	100

对于有监督学习, 用 2 种不同预处理得到的训练集训练至收敛, 最终得到两个分类器. 对于 3 个不同的半监督学习算法, 模型训练将进行多轮, 每次有标签数据集更新后需重新训练, 且每轮都需要将分类器训练至收敛, 直至所有的无标签数据被打上伪标签且被用于最终的分类器训练, 每种方法同样会得到两个不同的分类器. 4 种方法全部训练完成后, 进行分类器测试, 根据不同的评估指标对比性能.

3.2 分类器评价标准

对于每个待检测的 URL, 分类器最终可能产生 4 种

不同的结果, 本实验中这 4 种情况分别解释为:

(1) *TP* (True Positive): 恶意 URL 样本, 且模型预测结果为恶意;

(2) *TN* (True Negative): 正常 URL 样本, 且模型预测结果为正常;

(3) *FP* (False Positive): 正常 URL 样本, 模型预测结果为恶意;

(4) *FN* (False Negative): 恶意 URL 样本, 模型预测结果为正常.

基于以上 4 种可能情况, 对于分类器的性能评判引入了精准度、*F*1、AUC 和 *KS* 4 个指标, 精准度和 *F*1 主要判断训练完成的分类器预测结果的准确性, *AUC* 和 *KS* 主要判断分类器对于 URL 是否恶意的区分能力强弱.

精准度即精确率, 在本实验中表示正确判断为恶意的 URL 样本占全部判断为恶意样本的比例:

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

*F*1 值是 *Precision* 和 *Recall* 的调和平均数. 因为 *Precision* 和 *Recall* 有时候会出现矛盾, 所以需要对它们进行综合考虑:

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{6}$$

AUC 值是 ROC 曲线下方的面积. ROC 曲线绘制的横坐标是 *FPR*, 而纵坐标是 *TPR*. 当 *TPR* 越大, 而 *TPR* 越小时, 说明分类结果是较好的. *TPR* 与 *FPR* 的计算如式 (7)、式 (8) 所示.

$$TPR = \frac{TP}{TP+FN} \tag{7}$$

$$FPR = \frac{FP}{FP+TN} \tag{8}$$

KS 值源自 *KS* 曲线, 同样反应分类器的划分能力, 不同的是 *KS* 曲线采取了另一个视角展示模型功效. *KS* 曲线是将概率从小到大进行排序, 取 10% 的值为阈值, 同理将 10%×*k* (*k*=1, ⋯, 9) 处值作为阈值, 计算不同的 *FPR* 和 *TPR*, 以 10%×*k* (*k*=1, ⋯, 9) 为横坐标, 同时分别以 *TPR* 和 *FPR* 为纵坐标画出两条曲线. 两条曲线之间最远的距离就是 *KS* 值, 而此处对应的阈值, 就是划分模型最优异的阈值.

$$KS = |\max(TPR - FPR)| \tag{9}$$

精准度是本次实验中最重要的指标. 我们的目标并不是训练出一个能检测出所有恶意 URL 的分类器, 而是希望能高效、低成本的部署分类器来一定程度上减少恶意 URL 带来的损失, 且分类器不会有过多的误报而降低可用性, 即得到精准度越高越好. 当分类器精准度相近时, 可综合考虑 *F*1、*AUC*、*KS* 3 个指标来判断分类器性能的优劣.

3.3 实验结果与对比分析

4 种不同训练方法最终分别得到的 2 个分类器用设定的实验指标进行对比, 结果如表 5 和表 6 所示.

表 5 分类器 1 在 4 种不同训练方法下的结果

分类器1	*Precision*	*F*1	*AUC*	*KS*
有监督学习	0.9852	0.9300	0.9548	0.8610
自训练	0.9670	0.9289	0.9394	0.8485
协同训练	0.9869	0.9052	0.9213	0.8196
本文方法	0.9942	0.9119	0.9322	0.8349

表 6 分类器 2 在 4 种不同训练方法下的结果

分类器2	*Precision*	*F*1	*AUC*	*KS*
有监督学习	0.9503	0.9271	0.9693	0.8436
自训练	0.9486	0.8720	0.9052	0.7460
协同训练	0.9494	0.8900	0.9246	0.7781
本文方法	0.9523	0.8997	0.9262	0.7977

表 5 显示, 对于分类器 1, 通过本文方法能够得到的精准度明显高于自训练, 且略高于有监督学习以及协同训练, 说明本文方法得到的分类器 1 有最低的误报率. 再比较 *F*1、*AUC*、*KS* 值, 可见本文方法在该 3 项指标的表现上虽稍低于有监督学习, 但在半监督学习中数值与其它 2 种方法得到的分类器 1 相近.

表 6 显示 4 种不同方法得到的分类器 2 的精准度相近, 综合比较下有监督学习的 *F*1、*AUC*、*KS* 3 个评估指标最高, 本文方法 4 项指标与其接近并稍高于自训练与协同训练.

综合表 5 与表 6, 可以分析出本文方法在所用有标签数据远少于有监督学习的情况下得到的 2 个分类器 4 项指标与有监督学习所得分类器相近, 精准度更是分别达到了 99.42% 与 95.23%, 即误报率在 4 种方式中最低, 满足恶意 URL 检测应用中的低误报率要求. 通过其它 3 项指标的对比可知本文方法所的分类器在低误报的同时, 保证了对 URL 良性或恶意的区分能力.

通过表 5 和表 6 的对比我们定义分类器 1 为强分

类器, 分类器 2 为弱分类器. 在协同训练中由于每轮 2 个分类器所新增的伪标签数据来自另一分类器的预测, 则可能导致弱分类器的过多错误预测对强分类器的性能产生负面影响. 如图 4~图 7 所示, 在协同训练的执行过程中强分类器的精准度会有较大起伏, 这表明分类器性能在训练过程中的不稳定性, 对最终的分类器性能产生较大影响. 而本文方法基于共同判断, 强分类器每轮新增的伪标签数据依然全部来自自身预测, 虽然存在因为与弱分类器的判断结果不同而未选择少量高置信度数据的情况, 但却降低了受到弱分类器影响的可能. 从图中可以看到本文方法中强分类器的精确度曲线虽也有起伏, 但相对平稳, 且整体呈上升趋势. 综上得出, 本文方法相较于协同训练牺牲了一定的性能提升速率得到了更高的稳定性.

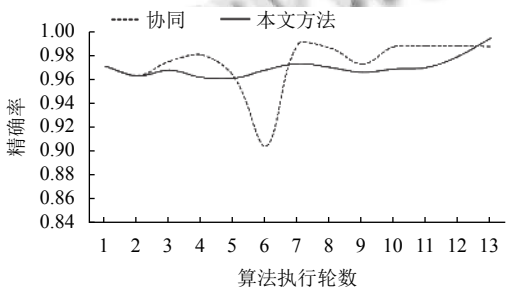

图 4 分类器 1 精准度变化趋势图

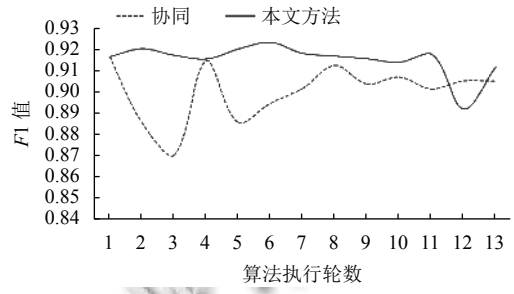

图 5 分类器 1 *F*1 值变化趋势图

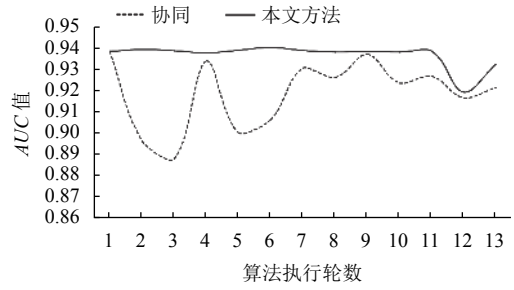

图 6 分类器 1 *AUC* 值变化趋势图

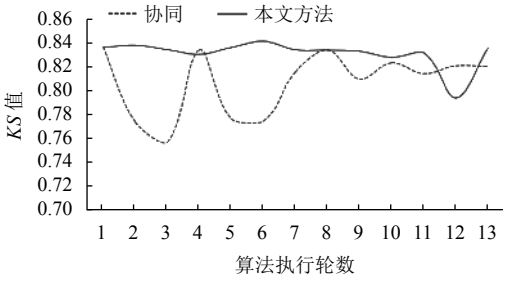

图 7 分类器 1 *KS* 值变化趋势图

4 总结

本文恶意 URL 检测方法结合了特征与文本处理预处理数据, 并对协同训练算法进行了改进, 仅用 0.67% 有标签数据训练出的两个分类器预测精准度分别达到 99.42% 与 95.23%, 低误报率使得该方法训练得到的检测模型有较高的实用性. 这种方式在现实应用中大幅度节约了人为打标签的成本, 减少了时间开销, 且检测效果接近有监督学习得到的分类器, 提供了有效应对新型恶意 URL 的方案. 未来的工作将考虑如何把这种半监督思想应用于恶意 URL 的在线学习中, 在节约开销的同时保证检测模型的定时更新.

参考文献

1 Kaspersky Security Bulletin 2018. https://securelist.com/kaspersky-security-bulletin-2018-statistics/89145/. (2018-09-04).
2 Sahoo D, Liu C, Hoi SCH. Malicious URL detection using machine learning: A survey. arXiv preprint arXiv: 1701.07179, 2017.
3 Prakash P, Kumar M, Kompella RR, *et al*. PhishNet: Predictive blacklisting to detect phishing attacks. 2010 Proceedings IEEE INFOCOM. San Diego, CA, USA. 2010. 1–5.
4 Tsai CF, Hsu YF, Lin CY, *et al*. Intrusion detection by machine learning: A review. Expert Systems with Applications, 2009, 36(10): 11994–12000. [doi: 10.1016/j.eswa.2009.05.029]
5 Le H, Pham Q, Sahoo D, *et al*. URLNet: Learning a URL representation with deep learning for malicious URL detection. arXiv preprint arXiv: 1802.03162, 2018.
6 Sommer R, Paxson V. Outside the closed world: On using machine learning for network intrusion detection. Proceedings of 2010 IEEE Symposium on Security and Privacy. Berkeley/Oakland, CA, USA. 2010. 305–316.
7 Sinclair C, Pierce L, Matzner S. An application of machine

learning to network intrusion detection. Proceedings 15th Annual Computer Security Applications Conference. Phoenix, AZ, USA. 1999. 371–377.

8 Buczak AL, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 2016, 18(2): 1153–1176.

9 吴海滨, 张冬梅. 基于上下文信息的恶意 URL 检测技术. 软件, 2019, 40(1): 63–68. [doi: 10.3969/j.issn.1003-6970.2019.01.013]

10 沙泓州, 刘庆云, 柳厅文, 等. 恶意网页识别研究综述. 计算机学报, 2016, 39(3): 529–542. [doi: 10.11897/SP.J.1016.2016.00529]

11 Warrender C, Forrest S, Pearlmutter B. Detecting intrusions using system calls: Alternative data models. Proceedings of the 1999 IEEE Symposium on Security and Privacy. Oakland, CA, USA. 1999. 133–145.

12 Mao GJ, Wu XD, Zhu XQ, *et al*. Mining maximal frequent itemsets from data streams. Journal of Information Science, 2007, 33(3): 251–262. [doi: 10.1177/0165551506068179]

13 Garera S, Provos N, Chew M, *et al*. A framework for detection and measurement of phishing attacks. Proceedings of the 2007 ACM workshop on Recurring Malcode. Alexandria, VA, USA. 2007. 1–8.

14 Sinha S, Bailey M, Jahanian F. Shades of grey: On the effectiveness of reputation-based “blacklists”. Proceedings of 2008 3rd International Conference on Malicious and Unwanted Software. Fairfax, VA, USA. 2008. 57–64.

15 Xu L, Zhan ZX, Xu SH, *et al*. Cross-layer detection of malicious websites. Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy. San Antonio, TX, USA. 2013. 141–152.

16 Huang HJ, Qian L, Wang YJ. A SVM-based technique to detect phishing URLs. Information Technology Journal, 2012, 11(7): 921–925. [doi: 10.3923/itj.2012.921.925]

17 Hou YT, Chang YM, Chen T, *et al*. Malicious web content detection by machine learning. Expert Systems with Applications, 2010, 37(1): 55–60. [doi: 10.1016/j.eswa.2009.05.023]

18 Canali D, Cova M, Vigna G, *et al*. Prophiler: A fast filter for the large-scale detection of malicious web pages. Proceedings of the 20th International Conference on World Wide Web. Hyderabad, India. 2011. 197–206.

19 Lee S, Kim J. WarningBird: Detecting suspicious URLs in Twitter Stream. NDSS. 2012. 1–13.

20 Zhou ZH, Li M. Semi-supervised regression with co-training. Proceedings of the 19th International Joint Conference on Artificial Intelligence. San Francisco, CA, USA. 2005. 908−913.

21 Zhou ZH, Li M. Semisupervised regression with cotraining-style algorithms. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(11): 1479–1493. [doi: 10.1109/TKDE.2007.190644]

22 梁吉业, 高嘉伟, 常瑜. 半监督学习研究进展. 山西大学学报 (自然科学版), 2009, 32(4): 528–534.

23 周志华. 基于分歧的半监督学习. 自动化学报, 2013, 39(11): 1871–1878.

24 McClosky D, Charniak E, Johnson M. Effective self-training for parsing. Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. Stroudsburg, PA, USA. 2006. 152–159.

25 Rosenberg C, Hebert M, Schneiderman H. Semi-supervised self-training of object detection models. Proceedings of 2005 7th IEEE Workshops on Applications of Computer Vision. Breckenridge, CO, USA. 2005. 29–36.

26 Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. Proceedings of the 11th Annual Conference on Computational Learning Theory. New York, NY, USA. 1998. 92−100.

27 Nigam K, Ghani R. Analyzing the effectiveness and applicability of co-training. Proceedings of the 9th International Conference on Information and Knowledge Management. New York, NY, USA. 2000. 86–93.

28 Zhou ZH. Disagreement-based semi-supervised learning. Acta Automatica Sinica, 2013, 39(11): 1871–1878. [doi: 10.3724/SP.J.1004.2013.01871]

29 Sindhwani V, Niyogi P, Belkin M. A co-regularized approach to semi-supervised learning with multiple views. Proceedings of the 22nd Workshop on Learning with Multiple Views. Cambridge, UK. 2005. 824−831.

30 Brefeld U, Gärtner T, Scheffer T, *et al*. Efficient co-regularised least squares regression. Proceedings of the 23rd International Conference on Machine Learning. New York, NY, USA. 2006. 137−144.

31 Farquhar JDR, Hardoon DR, Meng HY, *et al*. Two view learning: SVM-2K, theory and practice. Proceedings of the 18th International Conference on Neural Information Processing Systems. Cambridge, UK. 2005. 355−362.

32 Sridharan K, Kakade SM. An information theoretic framework for multi-view learning. Proceedings of the 21st Annual Conference on Learning Theory. Helsinki, Finland.

2008. 403−414.

33 Goldman SA, Zhou Y. Enhancing supervised learning with unlabeled data. Proceedings of the 17th International Conference on Machine Learning. San Francisco, CA, USA. 2000. 327−334.

34 Fushiki T. Estimation of prediction error by using K-fold cross-validation. Statistics and Computing, 2011, 21(2): 137–146. [doi: 10.1007/s11222-009-9153-8]

35 Zhou ZH, Li M. Tri-training: Exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(11): 1529. [doi: 10.1109/TKDE.2005.186]

36 Wang W, Zhou ZH. Co-training with insufficient views. Proceedings of the 5th Asian Conference on Machine Learning. Canberra, Australia. 2013. 467−482.

37 Ma J, Saul LK, Savage S, *et al*. Learning to detect malicious URLs. ACM Transactions on Intelligent Systems and Technology, 2011, 2(3): 30.

38 徐冬冬, 谢统义, 万卓昊, 等. 基于 TF-IDF 文本向量化的 SQL 注入攻击检测. 广西大学学报 (自然科学版), 2018, 43(5): 1818–1826.

39 Bengio Y, Ducharme R, Vincent P, *et al*. A neural probabilistic language model. The Journal of Machine Learning Research, 2003, 3: 1137–1155.

40 Collobert R, Weston J, Bottou L, *et al*. Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 2011, 12: 2493–2537.

41 Le Q, Mikolov T. Distributed representations of sentences and documents. Proceedings of the 31st International Conference on International Conference on Machine Learning. Beijing, China. 2014. 1188–1196.

42 Wallach HM. Topic modeling: Beyond bag-of-words. Proceedings of the 23rd International Conference on Machine Learning. New York, NY, USA. 2006. 977–984.

43 Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space. Proceedings of the 1st International Conference on Learning Representations. Scottsdale, AZ, USA. 2013.

44 Mikolov T, Sutskever I, Chen K, *et al*. Distributed representations of words and phrases and their compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems. Red Hook, NY, USA. 2013. 3111–3119.