Computer Science and Application 计算机科学与应用, 2020, 10(5), 935-943
Published Online May 2020 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2020.105096

Hans汉斯

Generating and Detection Malicious URL Based on Generative Adversarial Networks

Yang Zheng[1], Nurbol[2]

[1]College of Information Science and Engineering, Xinjiang University, Urumqi Xinjiang
[2]Network Center, Xinjiang University, Urumqi Xinjiang
Email: zxdwan717@126.com

Received: Apr. 23rd, 2020; accepted: May 7th, 2020; published: May 14th, 2020

Abstract

Malicious web page recognition based on machine learning is sensitive to data collection and annotation. This paper proposes a method of generating and detecting malicious web pages based on Generative Adversarial Networks (GAN). Design an encoder in order to encode malicious URL at character level. A small number of samples were used to train the model, and the ability of GAN to fit real samples was used to generate malicious web page samples. On the basis of traditional GAN, this paper adds a discriminator to discriminate benign and malignant web pages, and achieves the function of discriminating malicious web pages. Finally, the feasibility of the generated data and the effectiveness of the discriminant model with the currently supervised classifier are verified by vertical and horizontal comparison experiments.

Keywords

Malicious Web Page Detection, Machine Learning, Generative Adversarial Network, Multiple Discriminator, Classification

基于生成式对抗网络的恶意URL数据生成与检测

郑　阳[1]，努尔布力[2]

[1]新疆大学信息科学与工程学院，新疆　乌鲁木齐
[2]新疆大学网络中心，新疆　乌鲁木齐
Email: zxdwan717@126.com

收稿日期：2020年4月23日；录用日期：2020年5月7日；发布日期：2020年5月14日

文章引用：郑阳，努尔布力．基于生成式对抗网络的恶意 URL 数据生成与检测[J]．计算机科学与应用，2020，10(5)：935-943. DOI: 10.12677/csa.2020.105096

摘　要

针对基于机器学习的恶意网页识别中对数据集的收集和标注敏感的问题，提出了一种基于生成式对抗网络(GAN)的检测方法，并且设计了编码器，将恶意URL进行字符级编码。通过使用少量样本训练模型，通过GAN拟合真实样本的能力，生成恶意网页样本。本文在传统GAN的基础上增加了一个判别器用来判别良性和恶性网页，达到了判别恶意网页的作用。最后通过横纵对比实验，分别验证了生成数据的可行以及判别模型可以达到当前有监督分类器相当的效果。

关键词

恶意网页识别，机器学习，生成对抗网络，多判别器，分类

1. 引言

当今社会中人们生活必不可少的一定是互联网，网络占据着越来越重要的位置。伴随着人们对互联网所提供服务的依赖性越来越高，各种类型的网站如雨后春笋般涌现出来。依据我国互联网络信息中心第 44 次《中国互联网络发展状况统计报告》[1]，截止到 2019 年 6 月，我国网络用户的总体数量已攀升至 8.54 亿，如今该数字仍然处于持续上升状态。2019 年瑞星“云安全”[2]系统在全球范围内共截获恶意网址(URL)总量 1.45 亿个，其中挂马类网站 1.2 亿个，钓鱼类网站 2454 万个。其中中国有 417.43 万个。根据赛门铁克 2018 年互联网安全威胁报告[3]分析，10 个 URL 中就有 1 个是恶意链接，而在 2017 年，这一比例为 1/16。但由于个别网站构架中存在的各类安全问题，再加上庞大的用户基数，信息泄露的危险始终存在于我们身边。网络，成为了一把给予便利与风险的双刃剑。

据 Google 安全部门提供的研究数据显示，目前该搜索引擎中直接链接到钓鱼网页与木马网页的搜索结果就要占据搜索的总数 1.3%。因此我们不难发现，以上两种恶意攻击形式是黑客进行不法行为的重要途径[4]。此外，由于互联网极高的传播性，短短几分钟的时间，就会有不计其数的互联网用户对同一个恶意网页进行访问，加之此类恶意网页往往存在着类型变化大、传播面积大、生命周期短等特征，现已成为一种具有最大危害性的攻击来源。如何高效且快速地检测恶意网页，是目前互联网安全面临的重大问题。

本文以恶意 URL 作为分类数据，在其统计特性基础上结合深度神经网络[5]中的生成式对抗网络(Generative Adversarial Network, GAN [6])对恶意 URL 进行生成预测判别，生成数据以扩大训练样本，并通过实验验证了生成数据的有效性以及本文设计的分类器的分类效果较好的结果。

2. 相关工作

2.1. 恶意网页检测

现阶段对恶意 URL 的检测与判别主要依靠以下三种技术手段：黑名单技术、启发式技术、机器学习技术。

所谓黑名单技术，即在互联网用户对可疑网址进行举报后、专业人员将其中的恶意网址进行判别与整合从而生成列表进行发布。因此，只要某网站存在于该黑名单中，用户一旦试图对其进行访问，就会收到浏览器弹出的提示或警告。由于黑名单技术具有较强的检测效率和较高便利性，是当下最经典，最传统的恶意网址算法技术，因此也被众多杀毒软件所应用。但黑名单技术也时常存在漏判、错判等问题[7]。

启发式算法是对黑名单方法的补充，它不使用黑名单的精确匹配方法来完成恶意网址的识别工作，而是通过对现有恶意网址进行分析，借助启发式规则对所有现存的乃至未出现的网址进行判别和检测。但凡其存在足够的相似性，就会被判定为恶意网址。但此种检测方式的弊端，就是不法分子仍可以通过多种手段轻而易举地对其模糊匹配技术进行回避[8]。此外，以相似性作为筛选依据的检测方式带来了较高的恶意网址误报率。且设计的复杂性也为后续的程序优化带来了不小难度。尽管以 Moshchuk 为代表的设计者们提出，通过对目标网页存在的特殊过程创建、频繁地重定向等执行动态进行分析的方法识别恶意网页的签名[9] [10]，以此对启发式算法进行优化，但此种检测方式仍具有其局限性。

鉴于以上两种算法存在的种种不足，将机器学习算法[11] [12]应用于恶意网页识别的技术应运而生。它的特点是转为对目标网址的 URL 以及内容数据进行分析，并提取出该恶意网页域名的主要特征，以此为依据建立模型，并对未知网页的安全性进行预测。在机器学习中有监督算法需要通过对已知的恶意或非恶意网址的域名特征、注册信息、生存时间等信息进行抽取，记录其作为恶意或非恶意的特征，接下来通过支持向量机、决策树、逻辑回归等现有的算法进行数据建模，从而达到识别恶意网站的目的。总的来说，该算法在检测准确率与误报率等方面仍然有着不俗的表现，但由于有监督算法对标注数据的准确率以及网页特征的选取具有较高的敏感性，导致在实际运行过程中其检测结果的准确率和效率存在着一定范围的波动。

综上所述，上述多种方法无法实时监测多变的恶意 URL。基于恶意网页的生存周期较短等问题，所以在只有少量数据的情况下，实时检测恶意网页非常困难，因为机器学习等方法的检测效率往往对数据集的收集和标注等很敏感，特征的选择会严重影响其检测的效率和准确率。本文采用的 GAN 网络可以直接学习恶意 URL 的特征，无须预先对网页进行聚类、特征提取等操作，只需对 URL 进行编码和解码，即可生成与真实恶意 URL 样本具有高相似度的数据。从而达到扩充数据集的目的且可以提高分类器的检测效率。

2.2. 生成式对抗网络

近年来，人工智能和深度学习已经成为人们心中耳熟能详的高频词。在得益于计算机性能的提高以及大数据的发展，现在社会中信息化工具的普及也越来越广泛，从而人工智能方面的研究和发展也取得了长足的进步。目前而言，判别式模型和生成式模型是深度学习的两个基本模型，而其中的判别式模型发现状明显超过了生成式模型，这其中得益于反向传播算法(Back Propagation, BP) [13]等的发明。但这一现状在 Goodfellow 等人在 2014 年提出了 GAN 之后出现了转机，这一发明为生成式模型领域注入了新的生命力。

GAN 的思想来源于博弈论中的纳什均衡[14]，也即零和博弈。其模型包含两个部分，分别是生成器(Generator, G)和判别器(Detector, D)。以生成图片为例，G 是一个生成网络，它的输入是一个随机噪声 z，通过这个噪声生成一个图像 W’；而 D 是一个判别网络，它的任务是判别一张图片是真实的图片还是 G 生成的图片。D 的输入是真实图片 W 和 G 生成的图片 W’，D 通过输出一个概率来判定图片的真实性，如 D 输出结果是 1，那图片则来自真实图片，如果输出结果为 0，则图片来自于 G 成生的图片。在训练过程中，G 的任务是尽可能的学习图片的特征骗过 D，使 D 对 W’的判定值越高越好。而 D 则尽可能的

提高自己的判别能力，以免被 G 欺骗。G 和 D 构成了一个动态的“博弈过程”，最终的平衡点即纳什均衡点。对应的 GAN 生成网络模型如图 1 所示。

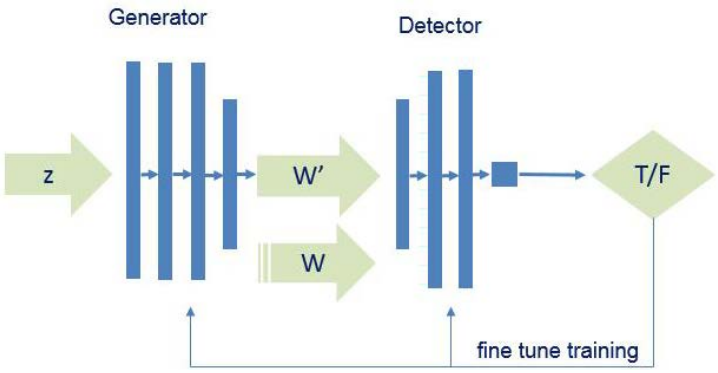

Figure 1. GAN model schematic diagram
图 1. GAN 模型示意图

自从 GAN 提出以来，其论文的发表数量逐年递增，已经成为学术界的热门研究方向，深度学习的创始人之一 Yann LeCun 甚至提出，生成式对抗网络是过去十年里在机器学习、人工智能领域内最有吸引力的想法。目前，GAN 已经广泛应用于图像和视觉领域，并且在自然语言处理、语音、网络安全等多个研究领域也取得了突破性的成果[15]。同时，自 GAN 问世以来其衍生模型也如雨后春笋般涌现出来，这些模型有对原始的框架进行改进，也有通过融合算法等方式来扩展原始模型的理论和应用，逐步解决了 GAN 梯度消失，训练困难和模式崩溃等问题。

本文在传统的 GAN 基础上，增加了一个判别器，改进了该模型的框架和损失函数，使得模型不仅可以生成数据，还可以通过增加的判别器达到检测恶意 URL 的目的。

3. 恶意 URL 字符生成判别模型

3.1. 域名字符分析

URL (Uniform Resource Locator)统一资源定位符，它表示某一网络资源存在于所在计算机网络上的位置，同时也是浏览器用于检索 web 上公布的任何资源的机制。我们对 URL 编码可以采取 One-Hot 编码方式，这是一种基础的一位有效位编码方式。类似一种将词库的所有词和每种状态对应的编码，是一种全映射的关系。但是这种方法存在明显的缺点，其一，该表示方法看不出词和词直接的关联，是一种任意表示，其二，如果词汇表的数量过于庞大时，那么这种表示方法势必会造成维度灾难。

考虑到 URL 的数据是多变的，尤其是我们需要检测的恶意 URL 更是如此，这些 URL 是无法通过分词的方法进行分割处理的，不否认部分 URL 是包含有实际意义的词语如 goodhousekeeping 等，但考虑到大多数的 URL 基本是无意义的，所以 URL 和文本处理还是有区别的，不能完全照搬对于英文单词采用空格来分词的方法处理 URL，所以本文不采用分词的方式处理 URL，而是在 one-hot 编码方式的基础上，采用字符拆分 URL 的方式，设计了字符级别的编码器和解码器，将字符组成的 URL 编码成对应的 URL 向量，从而输入到生成器和判别器中。

3.2. 编码器

为了将 URL 变成可输入的数据，从而输入到生成式模型中进行运算识别，我们需要将字符级的 URL 编码成为对应的 URL 向量。这里定义 URL 为向量 $\vec{u}$，即 $\vec{u}=\left[u_1,u_2,u_i,\cdots,u_n\right]$，其中 $u_i\left(i=1,2,\cdots,n\right)$ 为 URL

中的字符，n 为 URL 所包含的字符的多少，即 URL 的长度。我们这里定义函数 $f(x)=A(x)$ 是将字符转换为 AscaII 码的转换函数，那么 URL 字符向量 $\vec{u}=[u_1,u_2,u_i,\cdots,u_n]$ 可以转化为 $A(\vec{u})=[A(u_1),A(u_2),A(u_i),\cdots,A(u_n)]$，AscaII 码表示的 URL 向量。考虑到 AscaII 码的值域是 $[0,127]$，且在区间 $[0,127]$ 是 URL 中所不包含的字符，所以该映射函数的定义域是 $[33,127]$，现在做归一化处理，将定定义域 $[33,127]$ 映射到 $[0,1]$ 的区间，那么对于 URL 中的每一个字符的映射公式如(1)所示：

$$A'(u_i)=\frac{A(u_i)-MINA}{MAXA-MINA} \tag{1}$$

其中 $MAXA$ 为定义域上限，即 127，$MINA$ 为定义域下限，即 33。则经过上述变换后的 URL 为 $\vec{u}'=[A'(u_1),A'(u_2),A'(u_i),\cdots,A'(u_n)]$。举例来说，以 www.baidu.com 为例：

则其相应的字符向量是：$\vec{u}=[\mathrm{w},\mathrm{w},\mathrm{w},.,\mathrm{b},\mathrm{a},\mathrm{i},\mathrm{d},\mathrm{u},.,\mathrm{c},\mathrm{o},\mathrm{m}]$；

那么其 AscaII 码向量是：

$$A(\vec{u})=[119,119,119,46,98,97,105,100,117,46,99,111,109];$$

则归一化之后的向量是：

$$\vec{u}'=[0.914,0.9140.914,0.138,0.691,0.680,0.765,0.712,0.893,0.138,0.702,0.829,0.808]$$

URL 的长度都是长短不一的，但大部分 URL 的长度都不超过 200 个字符，故在此设计的 URL 向量的维度为 200，对于一些维度不足的 URL，在向量后采取补 0 操作。在所有 URL 映射为 200 维的向量之后，就可以将其作为生成式对抗网络的训练数据了。

3.3. 模型设计

本文提出的 Malicious URL-GAN (MU-GAN)不同于标准 GAN，本文设计的框架中含有三个模型，三种数据类型如图 2 所示：

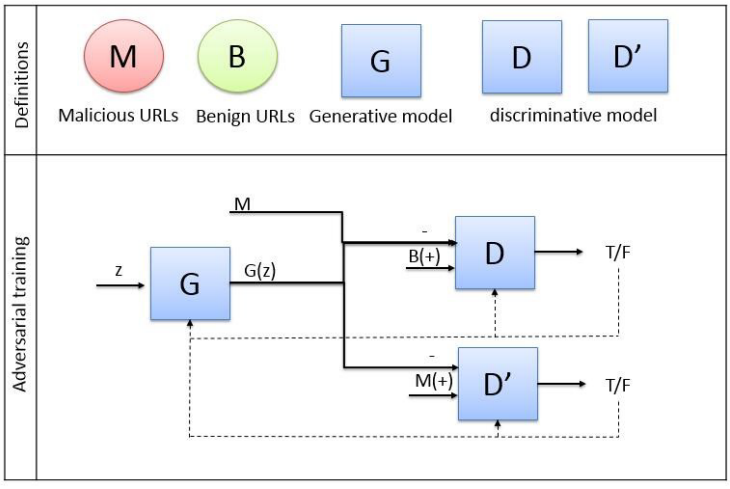

Figure 2. Framework of MU-GAN
图 2. 恶意网页对抗网络算法框架图

本模型中有三个主要组成部分：生成器 G、判别器 D 和判别器 D’，其中 D 是我们最终需要的，可以用来判别恶意和良性 URL 的分类器。在这里 G 网络的输入是一个高斯噪声，D’网络的输入是真实的恶意样本和 G 生成的样本，D 的输入是真实的良性样本，真实的恶意样本，和 G 生成的样本，D 分辨恶

意(包括真实样本中的恶性和 G 生成的样本)和真实数据中的良性样本，它指导 G 生成看起来像真实样本 B 的样本。D’分辨真实的恶意样本和 G 生成的样本，它指导 G 生成的样本更像真实的恶意样本 M，在每轮训练中，通过 D 和 D’的反馈，我们希望 G 成生更像恶意样本的样本来骗过 D’，同时像良性样本的样本来骗过 D。

一般生成式对抗网络的训练重点都在生成器，但本文中的模型将训练重点放在判别器，也即是我们最后所需要的分类器，我们希望得到在训练结束后，保存判别器 D，并通过对比试验验证 D 的效果。

在本模型中，我们有两个判别器，这是在以往的 GAN 模型中极少见到的，也是本模型的特别之处。众所周知，GAN 有模型训练不稳定的问题，我们希望可以通过改变 GAN 结构的方法来解决这个问题。判别器最主要的作用就是为生成器提供下降梯度。如果判别器太差，则无法提供有效的梯度，如果判别器太好，则生成器梯度消失严重。所以希望采用两个判别器解决这个缺点。两个分类器所分的任务不同，这就像让两个分类器分别处理图像中颗粒度和细粒度的分类。在本模型中，D 是用来分辨良性和恶意网页 URL 的分类模型，D’是对应的是标准 GAN 中的判别器，它的功能是要分辨生成数据和真实数据的区别。

3.4. 损失函数

由于本模型中有两个判别器，所以损失函数有三个，在这里我们均使用交叉熵损失函数，我们在训练过程中逐步更新每个模型的参数的损失函数如下：

当我们固定 G 时候，D 的损失函数为：

$$\max_{D} V(D,D',G) = \mathrm{E}_{b\sim P_{data}(b)}\left[\log\left(D(b)\right)\right] + \mathrm{E}_{z\sim P_z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right] + \mathrm{E}_{m\sim P_{data}(m)}\left[\log\left(1-D\left(G(z)\right)\right)\right] \tag{2}$$

当我们固定 G 时候， D' 的损失函数为：

$$\max_{D'} V(D,D',G) = \mathrm{E}_{m\sim P_{data}(m)}\left[\log\left(D'(m)\right)\right] + \mathrm{E}_{z\sim P_z(z)}\left[\log\left(1-D'\left(G(z)\right)\right)\right] \tag{3}$$

当我们固定 D 和 D' 时候， G 的损失函数为：

$$\min_{G} V(D,D',G) = \mathrm{E}_{z\sim P_z(z)}\left[\log\left(1-D'\left(G(z)\right)\right)\right] + \mathrm{E}_{z\sim P_z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right] \tag{4}$$

4. 实验分析

4.1. 实验环境与数据集

本实验的具体软硬件环境配置如表 1 所示：

Table 1. Experimental environment configuration
表 1. 实验环境配置

软硬件条件	参数
操作系统	Windows 10
GPU 规格	GTX 1660Ti
内存	32 G
编程语言	Python 2.7
深度学习框架	TensorFlow 0.12.0
机器学习平台	Anaconda 4.7.10

本文的数据集来自于 Phishtank 网站(https://www.phishtak.com)中的爬取的 2776 恶意网页，和 Yahoo 网站(https://www.yahoo.com)中爬取的 5883 良性网页。本文的数据划分相对复杂，因为本文有三种数据，除了上面的数据还有生成网络生成的恶意网页数据。由于横向对比的实验需要，本文将数据集划分成五种大小的数据，分别是 300、500、1000、1500、2000，其中每组数据都是从真实的恶意和良性样本中混杂后随机抽取的，此外对比用的生成数据每组相应要添加 100、200、300、500、700，都大概原占样本数量的三分之一左右。

4.2. 实验设计

训练模型框架使用 Tensorflow 的标准训练框架，其中主要包括 URL 字符级别编解码器的框实现、一个生成器和两个判别器架构定义、运算流成的定义和框架实现、对抗训练算法定义实现等。

在本文模型中生成模型的参数与判别模型的权重参数，都采用 TensorFlow 中已经封装好的高斯分布进行初始化，偏置部分采用全部置零的方式初始化，在此基础上进行训练来优化参数。其中，三个网络的参数配置如表 2 至表 4 所示：

Table 2. Generative network's parameters configuration
表 2. 生成网络 G 的参数配置

G 模型参数名称	参数值
输入层节点数	100
第一层隐含层数	150
第一层隐含层数	300
输出层节点数	200
学习效率	0.0001
网络输入数据模型	高斯分布
最大迭代次数	500

Table 3. Discriminator network D parameters configuration
表 3. 判别网络 D 的参数配置

D 模型参数名称	参数值
输入层节点数	200
第一层隐含层数	300
第一层隐含层数	150
输出层节点数	1
学习效率	0.0001
网络输入数据模型	恶意和良性 URL

Table 4. Discriminator network D' parameters configuration
表 4. 判别网络 D' 的参数配置

D'模型参数名称	参数值
输入层节点数	200
第一层隐含层数	300
第一层隐含层数	150
输出层节点数	1
学习效率	0.0001
网络输入数据模型	真实和生成 URL

其中激活函数采用 ReLU，因为是二分类，两个判别模型最后一层均用 sigmoid，为了防止过拟合，在隐含层使用 dropout [16]操作。

实验设计部分，我们采取了两条设计思路，其一在训练好 MU-GAN 以后，生成一定量的生成样本，通过不同数量的原始样本和加入生成样本的数据分别对逻辑回归、决策树和支持向量机三个有监督的分类模型进行训练。看再加入生成样本后，是否可以提高分类器的分类效果，用以验证生成样本的真实性。其二，在同等的数据下训练刚才说的三个分类器和本文设计的分类模型，最后验证四个分类器的分类效果。

4.3. 实验结果分析

通过训练后的生成模型将会越来越接近真实恶意 URL 样本的分布，判别器 D 的效果也会越来显著。采用上文所说的准确率来评定实验效果，通过对分类器的和最终训练出的判别器 D 的横纵比较，在五种不同样本和是否使用生成数据后得出对比实验识别率如表 5 所示：

Table 5. The accuracy of three classifiers in five sets of data (%)
表 5. 三种分类器在五组数据下的准确率(%)

	300		500		1000		1500		2000	
	ori	gen	ori	gen	ori	gen	ori	gen	ori	gen
LR	67.91	74.14	80.33	83.93	81.64	84.06	83.35	86.62	85.25	88.62
DT	65.89	71.50	73.19	76.83	75.86	76.12	76.85	78.13	78.05	80.96
SVM	75.86	77.63	80.87	84.82	85.73	86.94	86.41	88.09	89.21	90.54

其中，ori 代表在训练各种分类器时候，只用了原始的没有添加生成样本的情况。gen 是分类器在使用原始样本加生成模型 G 生成的样本之后的对比组。

从表中可以看出，三种分类器在使用了加入了生成器生成的样本以后分类效果都有了一定程度的增加，增幅大概在百分之一到百分之五之间，这既可以证明生成器生成的数据是可以拟合真实的恶意 URL 的，又可以说明这种生成样本的形式在只有少量数据时，是可以通过生成样本的方式提高分类器的效率的。

为了验证本文的判别模型 D 的分类效果，在使用收集的所有样本的情况下，我们分别验证四种分类器的分类效果，实验结果如表 6 所示：

Table 6. The comparison of four classifier (%)
表 6. 各分类器的分类结果对比(%)

	正确率	错误率	精确率	F-measure
GAN	88.38	11.62	87.73	88.48
LR	86.63	13.37	86.26	86.69
DT	81.79	18.21	81.42	81.89
SVM	92.23	7.77	92.54	92.19

从结果看出，本文提出的 MU-GAN 模型中的判别器 D 的正确率达到了 88.38%。这支持了我们的假设，即对抗性训练可以用于检测恶意 URL。更进一步的发现是，尽管本模型依赖于半监督学习，但它的性能与全监督分类算法的相似。

通过以上实验我们不难发现，对抗模型有两方面的借鉴性，其一：在一些需要非常大的真实数据集

的任务中，GAN 可以通过拟合真实样本的分布，生成大量高度接近真实样本的数据看，来提高分类器的分类效果以及其他应用。其二：对前文的描述可知，对抗模型在本文的分类任务中，应用相对较少，通过本实验，这使得我们相信本模型演示了在 GAN 文本分类任务中的良好表现。

5. 结束语

恶意网页的判别是网络安全领域中的重要任务之一，其中数据的收集标注等问题严重影响检测效率，本文在 GAN 网络的基础上，改进了判别器的数量，提出了一种在少样本情况下的恶意网页检测方法，希望通过这种方法解决数据收集和标注困难的问题。并通过实验验证了生成数据的可行性以及判别模型的检测效率，本模型中的 D 可以达到目前最新的有监督训练模型的相当的检测率。本文将 URL 进行编码，通过编码器将 GAN 生成的数据进行限制，通过 GAN 可以高度拟合样本分部的性能，进行训练。本文的下一步工作希望可以更加细化 URL 的特性，通过简单的特征提取生成更加真实的样本。并且希望可以通过将 GAN 和强化学习结合的方法，利用强化学习的奖励机制以及 Policy Gradient 等技术，解决 GAN 面对离散数据时的困境。

参考文献

[1] 中国互联网信息中心. 第 44 次中国互联网发展状况统计报告[EB/OL]. http://www.cac.gov.cn/2019-08/30/c_1124938750.htm, 2019-08-30

[2] 瑞星. 2019 年中国网络安全报告[EB/OL]. http://it.rising.com.cn/dongtai/19692.html, 2020-01-15

[3] 赛门铁克. 2018 年互联网安全威胁报告[EB/OL]. https://www.symantec.com/security-center/threat-report, 2019-04

[4] 沙泓州, 刘庆云, 柳厅文, 周舟, 郭莉, 方滨兴. 恶意网页识别研究综述[J]. 计算机学报, 2016, 39(3): 529-542.

[5] Goodfellow, I.J., Bengio, Y. and Courville, A. (2017) Deep Learning. MIT Press, Cambridge, 1-3.

[6] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., *et al*. (2014) Generative Adversarial Nets. *Advances in Neural Information Processing Systems* 27: *Annual Conference on Neural Information Processing Systems*, Montreal, 8-13 December 2014, 2672-2680.

[7] Prakash, P., Kumar, M., Kompella, R.R., *et al*. (2010) PhishNet: Predictive Blacklisting to Detect Phishing Attacks. *Proceedings of the* 29*th IEEE International Conference on Computer Communications*, San Diego, 15-19 March 2010, 1-5.

[8] Sahoo, D., Liu, C.H. and Hoi, S.C.H. (2017) Malicious URL Detection Using Machine Learning: A Survey.

[9] Moshchuk, A., Bragin, T., Deville, D., *et al*. (2007) Execution-Based Detection of Malicious Web Content. *Proceedings of* 16*th USENIX Security Symposium*, Boston, 3:1-3:16.

[10] Rieck, K., Krueger, T. and Dewald, A. (2010) Efficient Detection and Prevention of Drive-by-Download Attacks. *Proceedings of the* 26*th Annual Computer Security Applications Conference*, Austin, 6-10 December 2010, 31-39.

[11] Tobiyama, S., Yamaguchi, Y., Shimada, H., *et al*. (2016) Malware Detection with Deep Neural Network Using Process Behavior. *Proceedings of the* 40*th Annual Computer Software and Applications Conference*, Atlanta, 10-14 June 2016, 577-582. https://doi.org/10.1109/COMPSAC.2016.151

[12] 张洋, 柳厅文, 沙泓州, 时金桥. 基于多元属性特征的恶意域名检测[J]. 计算机应用, 2016, 36(4): 941-944 + 984.

[13] Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) Learning Representations by Back-Propagating Errors. *Nature*, **323**, 533-536. https://doi.org/10.1038/323533a0

[14] He, D., Chen, W., Wang, L., *et al*. (2013) A Game-Heoretic Machine Learning Approach for Revenue Maximization in Sponsored Search. In: *International Joint Conference on Artificial Intelligence*, AAAI Press, Beijing, 206-212.

[15] 王坤峰, 苟超, 段艳杰, 等. 生成式对抗网络 GAN 的研究进展与展望[J]. 自动化学报, 2017, 43(3): 321-332.

[16] Srivastava, N., Hinton, G., Krizhevsky, A., *et al*. (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, **15**, 1929-1958.