计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2020.20190860
57(4):746-766,2020

基于深度神经网络 burst 特征分析的网站指纹攻击方法

马陈城[1,2] 杜学绘[1,2] 曹利峰[1,2] 吴 蓓[3]
[1](战略支援部队信息工程大学 郑州 450001)
[2](河南省信息安全重点实验室(战略支援部队信息工程大学) 郑州 450001)
[3](61497 部队 北京 100000)
(machencheng07@foxmail.com)

burst-Analysis Website Fingerprinting Attack Based on Deep Neural Network

Ma Chencheng[1,2], Du Xuehui[1,2], Cao Lifeng[1,2], and Wu Bei[3]
[1](*Strategic Support Force Information Engineering University*, *Zhengzhou* 450001)
[2](*He'nan Province Key Laboratory of Information Security* (*Strategic Support Force Information Engineering University*), *Zhengzhou* 450001)
[3](*Unit* 61497, *Beijing* 100000)

Abstract Anonymous network represented by Tor is a communication intermediary network that hides user data transmission behavior. The criminals use anonymous networks to engage in cyber crimes, which cause great difficulties in network supervision. The website fingerprinting attack technology is a feasible technology for cracking anonymous communication. It can be used to discover the behavior of intranet users who secretly access sensitive websites based on anonymous network, which is an important mean of network supervision. The application of neural network in website fingerprinting attack breaks through the performance bottleneck of traditional methods, but the existing researches have not fully considered to design the neural network structures based on the characteristics of Tor traffic such as burst and the characteristics of website fingerprinting attack technology. There are problems that the neural network is too complicated and the analysis module is redundant, which leads to problems such as incomplete feature extraction and analysis and running slowly. Based on the researches and analysis of Tor traffic characteristics, a lightweight burst feature extraction and analysis module based on one-dimensional convolutional network is designed, and a burst-analysis website fingerprinting attack method based on deep neural network is proposed. Furthermore, aiming at the shortcoming of simply using the threshold method to analyze fingerprinting vectors in open world scenarios, a fingerprint vector analysis model based on random forest algorithm is designed. The classification accuracy of the improved model reaches 99.87% and the model has excellent performance in alleviating concept drift, bypassing defense techniques against website fingerprinting attacks, identifying Tor hidden websites, performance of models trained with a small amount of data, and run time, which improves the practicality of applying website fingerprinting attack technology to real networks.

收稿日期:2019-12-17;**修回日期**:2020-02-09
基金项目:国家重点研发计划项目(2016YFB0501901,2018YFB0803603);国家自然科学基金项目(61502531,61702550,61802436)
This work was supported by the National Key Research and Development Program of China (2016YFB0501901, 2018YFB0803603) and the National Natural Science Foundation of China (61502531, 61702550, 61802436).
通信作者:杜学绘(dxh37139@sina.com)

Key words website fingerprinting attack; deep neural network (DNN); burst analysis; Tor anonymous network; network supervision

摘 要 以 Tor 为代表的匿名网络是一种隐匿用户数据传输行为的通信中介网络.不法分子利用匿名网络从事网络犯罪,对网络监管造成了极大的困难.网站指纹攻击技术是破解匿名通信的可行技术,可用于发现基于匿名网络秘密访问敏感网站的内网用户行为,是网络监管的重要手段.神经网络在网站指纹攻击技术上的应用突破了传统方法的性能瓶颈,但现有的研究未充分考虑根据突发流量(burst)特征等 Tor 流量特征对神经网络结构进行设计,存在网络过于复杂和分析模块冗余导致特征提取和分析不彻底、运行缓慢等问题.在对 Tor 流量特征进行研究和分析的基础上,设计了轻便的基于一维卷积网络的 burst 特征提取和分析模块,提出了基于深度神经网络分析 burst 特征的网站指纹攻击方法.进一步,针对在开放世界场景中仅使用阈值法简单分析指纹向量的不足,设计了基于随机森林算法的指纹向量分析模型.改进后的模型分类准确率达到了 99.87%,在缓解概念漂移、绕过网站指纹攻击防御机制、识别 Tor 隐藏网站、小样本训练模型和运行速度等方面均有优异的性能表现,提高了网站指纹攻击技术应用到真实网络的可实践性.

关键词 网站指纹攻击;深度神经网络;burst 特征分析;Tor 匿名网络;网络监管

中图法分类号 TP393

对于党政军网络及大型企业网络等敏感网络,网络监管是维护网络良好秩序的重要手段.近年来发展迅速的流量加密和匿名网络技术,一方面保护了网络的敏感数据和用户隐私,另一方面也给网络监管带来了巨大的困难和挑战.SSH 和 VPN 等技术通过加密数据包载荷,可绕过基于载荷字段的流量分析和检测,但通过分析数据包的长度分布等规律,加密流量仍能被有效分析[1-3].但随后的 Tor(the onion router)匿名通信技术进一步隐匿了数据包长度信息,给流量分析带来了更大的困难.由于匿名通信系统具有节点发现难、服务定位难、用户监控难、通信关系确认难等特点,利用匿名通信系统隐藏真实身份从事恶意甚至网络犯罪活动的现象层出不穷[4],如利用暗网进行地下交易[5]及国内不法分子翻越中国墙访问不健康网站和发表不正当言论等行为.

Tor 网络[6]是匿名网络的代表之作.目前 Tor 网络在全球拥有 6 000 个志愿者节点,日活跃用户达到了 200 万[7].Tor 基于传输层安全协议(transport layer security, TLS)加密数据包载荷以及随机链路技术来保护用户端的数据隐私.其原理如图 1 所示,用户本地的客户端与 Tor 目录服务器进行协商分配链路节点,由于构成通信链路(circuit)的 3 个 Tor 节点 relay 的随机性和周期更新性,基于链路追溯数据包是困难的.待传输数据在客户端相应地被依次实施 3 道传输层安全协议(TLS)加密,每经过一个 Tor 节点,最外面一层的加密就被相应地解开,因此即使控制了其中一个 Tor 节点,也无法读取用户的数据包内容.Tor 基于一个或多个 512 B 的数据单元(cell)实现数据传输.固定长度的 cell 传输模式使得过去基于数据包长度的分析手段失去了攻击和分析效果.为了对基于 Tor 匿名网络的通信和访问行为进行有效监管,针对 Tor 匿名通信系统的攻击和分析技术研究发展迅速,如流水印技术[8]、流量关联分析技术[9]等.其中,网站指纹(website fingerprinting, WF)攻击技术发展尤为迅速[10-11].相比其他匿名通信攻击技术,WF 攻击技术具有易部署、低成本的特点.面向加密或匿名传输的 WF 攻击技术基于内网用户访问网站产生的流量数据对模型进行训练,模型对新产生的网页流进行分类,分析该网页流是否正在利用加密通道或匿名通信网络秘密访问敏感网站,如非法网站或可能导致内网失泄密的网站,及以暗网为代表的隐藏网站等[12],实现对利用匿名网络访问非法网站行为的攻击与分析.

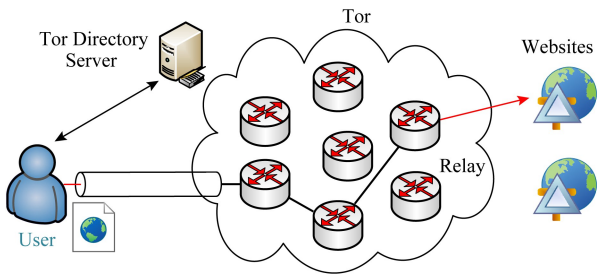

Fig. 1 Schematic diagram of the Tor network

图 1 Tor 网络原理示意图

WF 攻击与分析本质上是一个分类问题[11],机器学习在网络空间安全中的广泛应用[13-14]促进了 WF 技术的快速发展,近几年神经网络方法更是隐隐成为研究 WF 技术的主要利器[15].基于神经网络的 WF 攻击技术通过数据驱动构建模型,使模型自动学习网站指纹特征.相比传统方法[16-17],神经网络方法能够学习到人工经验难以定义的网站指纹特性,实现更好的攻击效果[11].

但目前主流的基于神经网络的 WF 攻击与分析方法仍存在不足之处.WF 攻击技术研究通常基于封闭世界场景(close-world, CW)和开放世界场景(open-world, OW)2 个假设进行分析.CW 场景假设用户仅访问网络管理员定义的被监控的敏感网站,WF 模型需要识别出用户当前访问被监控网站的具体站点域名,是一个 n 分类问题(n 为被监控网站的数量);而 OW 场景假设用户访问任意网站,WF 模型需要识别用户是否正在访问被监控网站集的站点,即识别网页流是否属于被监控网站集,是一个二分类问题.在 CW 和 OW 场景下,当前基于神经网络的 WF 研究都仅直接利用经典的神经网络架构,如 VGG16[18],ResNet[19]等,没有根据 WF 攻击技术的特点对神经网络模型结构进行设计和改进,存在网络过于复杂和分析模块冗余导致特征提取和分析不彻底、模型运行缓慢等问题[20],因此神经网络在 WF 攻击技术上的适应性还有待提高,模型性能还有很大的提升空间.另外,神经网络方法在 OW 场景下通常仅基于阈值判别法分析神经网络输出的指纹向量以实现二分类决策[21].由于神经网络方法输出的指纹向量的高度准确性,阈值法虽然简单但也表现出了较好的分类性能[22].但是阈值法没有分析被监控网站集和非监控网站集的指纹向量在各维度的相关性,也没有学习被监控网站集和非监控网站集的二类别特性.在被监控集网站为天然自成一类的情况下(如被监控集的站点均为 Tor 隐藏网站),阈值法的分类性能表现出较大的不足.

针对上述研究存在的问题,本文通过对 Tor 匿名网络流量序列的特征表现进行研究后,设计了基于深度分析 burst 特征的网站指纹攻击模型(deep burst-analysis based website fingerprinting attack, DBF).强加密性和隐匿性的 Tor 网络流量只有少数特征可分析出有用信息,突发流量特征(burst)是其中的一个重要的上层特征,它反映了访问网站时数据交互过程中的一段持续性的数据传输行为.为对 Tor 匿名网络流量的 burst 特征进行有效发现与分析,本文分别针对 CW 与 OW 场景进行了相关研究.在 CW 场景中,设计了基于 burst 特征提取模块和 burst 特征抽象学习及深度分析模块的 DBF-CW (DBF in Close-World)模型.首先,burst 特征提取模块通过由多个卷积层平行拼接而成的浅层卷积神经网络(convolution neural network, CNN)对不同长度的 burst 特征进行提取;然后,burst 特征抽象学习及深度分析模块对 VGG16 架构的基本区块(由 2 层卷积层及一层池化层组成)和含残差连接的密集神经网络(dense neural network, DNN)进行融合,对 burst 特征进行深度的抽象学习,由此提取并输出网页流的指纹向量,并通过对指纹向量做反向最大值函数计算实现对被监控网页流的网站标记识别;在 OW 场景中,基于 DBF-CW 输出的指纹向量结果,进一步设计了基于随机森林算法的二分类模型 DBF-OW(DBF in Open-World),通过对指纹向量进行向量维度相关性分析,模型可以学习二分类特性,实现了比阈值法更好的分类效果.

本文的主要贡献有 3 个方面:

1) 在封闭世界场景中设计了一个基于 CNN 和 DNN 的 WF 攻击模型 DBF-CW,通过对浅层卷积网络、VGG16 基本区块和含残差连接的密集神经网络进行连接与结合,形成多层深度神经网络结构,对 Tor 流量序列的 burst 特征进行提取和深度分析,提高了 burst 特征发现的成功率和准确率,模型对 Tor 流量的分析和分类性能得到很大的提高;

2) 在开放世界场景中设计了一个基于随机森林算法的 WF 模型 DBF-OW,改进了基于阈值法的决策思路,通过分析 DBF-CW 输出的指纹向量间各维度相关性与被监控网站集和非监控集二类别的映射规律,实现了更有效的二分类决策;

3) 使用了多个数据集对方法进行评估,从实践的角度验证了本文所提出的 DBF 模型在缓解概念漂移、绕过网站指纹攻击防御机制、识别 Tor 网络隐藏网站、小样本训练模型和运行速度等方面优异的性能表现.

1 相关工作

1.1 针对匿名通信的攻击与分析技术对比

从对流量的干扰程度及流量的采集点 2 个维度进行分析[23],匿名通信攻击技术主要可分为被动端到端流量分析[9]、主动端到端流量分析[8,24-25]、被动单端流量分析[1,12,26]和主动单端流量分析[27-29],

它们的区别如表 1 所示.端到端分析在实际网络环境中难以实施完备的攻击,因为需要在被监控站点近端进行系统部署,而站点数量往往是非常庞大的.主动单端攻击通过向用户端注入恶意代码,通过分析用户机器物理特征(如内存)与访问不同网站时的映射关系来实现攻击,操作性要求较高.相比之下,以网站指纹攻击为代表的被动单端流量分析的实现成本最低,通过监听并分析用户近端流量即可建模,是当前实现全面的敏感站点检测的最可行方法.

Table 1 Comparison of Four Anonymous Network Communication Attack Technologies

表 1 4 种匿名网络通信攻击技术对比

Attack Type	Deployment	Interference	Assumption	Representative Technology	Characteristics
Passive End-to-end	Hard	None	Middle	Flow Correlation	Imperceptibility
Active End-to-end	Very Hard	Big	Weak	Flow Watermarking	Strong Traceability
Passive Single-end	Easy	None	Strong	Website Fingerprinting	Low Cost, Cost-effectiveness
Active Single-end	Middle	Middle	Middle	Malicious Code Injection	Strong Operability

1.2 网站指纹攻击技术发展现状

网站指纹(WF)攻击是一个本地的、被动地获取用户进出流量、不主动干预流量状态的一种流量窃听攻击.如图 2 所示,WF 攻击的发起者可以是用户与 Tor 入口节点之间链路上的本地管理员(local administrator)、服务提供商(Internet server provider, ISP)、自治系统(auto-nomous system, AS)或者控制了 Tor 入口节点的攻击者.网络管理员首先定义需要监控的敏感网站集,通过前期获取用户端近端流量样本和网站标记形成训练数据,完成训练的模型部署在用户端近端的链路上.基于被动监听用户的进出流量判断用户当前是否正在访问被监控网站,以达到网络监管的目的.

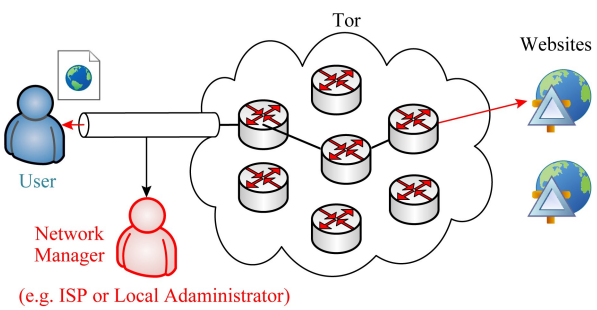

Fig. 2 Schematic diagram of WF attack

图 2 WF 攻击原理示意图

WF 攻击通常基于 3 种模型假设:

1) 用户访问行为单一.假设用户在同一时间只浏览一个网页,攻击者可以简单获取到网页流的开始和结束.

2) 无噪声流量.假设网页流无背景流量,不需要处理噪声流量.

3) 特殊网页可代表网站.假设用户访问某个具体网站时必将访问某个特殊网页(如网站首页),因此网站指纹分析可转化为网页指纹分析.

WF 攻击技术由于初期所基于的安全假设过于理想化而没有被广泛认可[30],近年来有许多研究围绕放松其基于的安全假设展开[31].Gu 等人[1]在 2015 年在用户同时访问 2 个网站的复杂情况下成功实施了 WF 攻击;Wang 等人[32]在 2016 年提出了模型更新算法以应对数据概念漂移问题,提出了多网页流分割算法以应对用户同时浏览多个网页的情况,还提出了处理流量噪声的手段等;Cui 等人[33]在 2019 年提出了 2 个针对连续和重合网页流的分割算法;针对网站指纹攻击可转换为网页指纹攻击的理想假设,Cai 等人[34]在 2012 年基于隐 Markov 链对网站链接的点击关系进行分析,基于多网页训练形成网站指纹;Zhuo 等人[35]在 2017 年提出了一种面向分析网站链接的隐 Markov 链模型.

上述对模型基础性安全假设进行分析和放松的研究工作,为在理想条件下建模的 WF 攻击技术提供了数据清洗等基础性的支撑工作,大大提高了 WF 模型应用到真实网络中的可行性.这些基础性的工作同样适用于本文模型,因此本文不涉及对安全假设的研究,旨在在理想条件下,提高 WF 模型在 2 个场景下对 Tor 匿名通信的攻击和分析能力,从提升分类性能的角度提高 WF 攻击技术应用到实际的可行性.

依据数据封装协议的不同,WF 攻击主要分为 3 类[36].在早期网站还使用 HTTP1.0 进行数据传输时,攻击者通过分析资源(如图片、文字等)长度可实现 WF 攻击[37-38].而后 HTTP1.1,VPN 和 SSH 通过

加密和混淆的方式使攻击者无法获取网站资源长度特征,基于数据包长度的分析可构建网站的指纹信息[26,39].Tor 匿名网络通过填充和固定传输单元的大小进一步隐匿了长度特征,针对 Tor 网络的网站指纹攻击在当前仍是一个难点.

作为 WF 模型的信息源,流量特征的提取是决定模型性能的关键一环.Tor 流量可以在数据包、TLS 和 cell 层次上进行提取,实验证明在 cell 层次上提取特征最有利于对 Tor 流量的分析[40].由于只有方向特征和数量特征可利用,对 cell 的分析通常基于 cell 方向序列的形式.方向序列中的 burst 特征被 WF 研究广泛使用[16,36],是实现 WF 攻击的一个重要的上层特征.

当前主流的面向 Tor 网络的 WF 模型主要分为基于人工设计指纹的一般机器学习方法和指纹(半)自动学习的神经网络方法.如表 2 所示,序列号 1～7 为一般机器学习方法,其基于流量特征直接形成或者通过形态变换形成网站指纹;而序列号 8～14 为神经网络方法,它通过深度挖掘流量特征的方式自动学习形成网站指纹.表 2 还对各研究所采用的基础模型算法、所利用的流量基础特征的层次、类型和表现形式进行了总结和描述.

Table 2 Comparison of Website Fingerprint Attack Methods for Tor

表 2 面向 Tor 网络的网站指纹攻击方法对比

No.	Model	Year	Method	Tor Traffic Basic Feature Extraction							Performance/%	
				Level		Type			Form		CW	OW
				TCP/IP	Cell	Time	Count	Direction	Sequence	Statistics	*Acc*	*TPR*
1	Panchenko et al.[10]	2011	SVM	√			√	√		√	54	73
2	Cai et al.[34]	2012	DLSVM+OSAD	√				√	√		80	80
3	Wang et al.[40]	2013	SVM+OSAD	√	√			√	√		91	90
4	*k*-NN[36]	2014	Variant KNN	√			√	√	√	√		90
5	CUMUL[17]	2016	CUMUL+SVM	√				√	√	√	92	96
6	*k*-FP[16]	2016	RF+KNN	√		√	√	√	√	√	91	93
7	Jahani et al.[41]	2016	FFT+SVM	√				√	√		97	
8	Abe et al.[42]	2016	SDAE+MLP		√			√	√		87	87
9	AWF[11]	2018	CNN, LSTM		√	√		√	√		97	80
10	DF[22]	2018	CNN		√	√		√	√		98	97
11	Oh et al.[15]	2018	CNN, DNN		√			√	√			98
12	He et al.[20]	2018	CNN+GRU		√			√	√		99	
13	Var-CNN[21]	2019	CNN	√		√		√	√	√	99	89
14	Rahman[43]	2019	CNN+DNN	√		√		√	√		98	98

Notes: *Acc* means accuracy; *TPR* means true positive rate; "√" means the item is selected.

对于一般机器学习方法,由于模型分析能力有限,指纹向量通常基于人工设计的规则进行提取,模型算法只进行指纹向量的距离对比、相似性计算等,因此模型所分析的特征一般需要包含丰富的表层信息,如通过增加特征维度、扩大特征的涵盖范围(如通过统计计算的方式)等,特征提取一般较为复杂.Wang 等人[36]在 2014 年通过对传统 KNN 算法进行加权改进,并基于改进后的 *k*-NN 算法分析高维特征集实施 WF 攻击,在封闭世界环境下取得了 91%的准确率.Panchenko 等人[17]在 2016 年对网页流实例使用累加和(cumulative representation, CUMUL)的方式表达序列特征,并使用基于 RBF(radial basis function)核函数的改进 SVM 进行分类,得到较好的效果.Hayes 等人[16]在 2016 年使用随机森林(random forest, RF)模型分析网页流的包计数、包间隔等共 150 维统计特征,并基于各叶子节点的标识形成网页指纹,通过传统 KNN 算法和汉明距离(Hamming distance)实现分类.然而,一般机器学习方法基于人工设计的指纹是不稳健的,匿名网络通过改进协议即可破坏这些指纹的提取[11].

对于指纹(半)自动学习的神经网络方法,由于模型具备强大的分析能力,指纹向量通常由模型自行分析得到,因此模型所分析的特征一般为不加处理的原始流量特征(如网页流的包方向序列、时间序列

等),较少通过统计的方式对原始数据进行加工.Abe 等人[42]在 2016 年提出了一种基于自编码神经网络和多层感知机分析 Tor cell 方向序列的 WF 方法,在开放世界场景中的准确度要高于此前的一般机器学习方法.Rimmer 等人[11]在 2018 年提出了利用深度学习的思想分析 Tor cell 方向序列并自动提取流量特征,以实现网站指纹建模.他们采用了 SDAE(stacked denoising autoencoder),CNN 和 LSTM(long short term memory)这 3 种神经网络进行模型构建.实验结果表明,基于神经网络的网站指纹攻击方法在性能上比当前人工提取指纹的传统方法要好.Sirinam 等人[22]在 2018 年基于 CNN 的 VGG 框架[18]分析 Tor 网页流 cell 序列特征,在封闭世界情景下达到 98% 的准确率,并成功攻破了 WTD-PAD 防御机制[44].Oh 等人[15]基于 CNN 分析 cell 序列和人工提取的 burst 长度特征实施 WF 攻击,在封闭世界情景得到了较高的准确率.He 等人[20]利用残差网络思想分析 cell 序列特征和包时间戳特征,基于 CNN 的 ResNets 架构[19]和 GRU 网络实施 WF 攻击,在封闭世界场景下得到了 99% 的准确率.Bhat 等人[21]在 2019 年同样基于 ResNets 架构训练 WF 模型,并且还引入了时间类特征,通过集成的方法综合分析了方向和时间类特征,也取得了 99%的分类准确率.Rahman 等人[43]在 2019 年通过实验证明了在一般机器学习算法中无法被有效使用的时间特征,在神经网络中能被有效挖掘出有用的信息.以上方法从特征设计和提取的角度对 WF 攻击技术进行改进,或利用已有的神经网络架构直接应用到 WF 攻击上,但都没有根据 Tor 流量和 WF 攻击技术的特点对神经网络结构进行改进,网络结构存在指纹分析不彻底或结构冗余的问题,前者导致分类准确率较低,后者导致模型运行速度缓慢.

burst 特征是方向(direction)特征的序列形式表现,是流量中的一种上层特征表现,在人工设计指纹的一般机器学习方法被广泛使用[16,36],但通过人工提取的 burst 特征只有长度信息,而位置抽象信息及潜藏的深度规律难以被人工设计的规则所提取和分析.同时,当前的神经网络方法[11,20,22]大多仅利用深度学习泛性地挖掘原始流量特征的规律,而没有从流量本身潜藏的特性分析出发设计模型,因此目前还没有针对 burst 特征进行分析的神经网络方法.对于数据加密、链路随机、传输时延不稳定、隐匿了数据传输单元长度特征的 Tor 流量,burst 特征无疑是一个非常重要的上层特征表现,而本文是该领域首个针对 Tor 流量 burst 特征进行分析的神经网络方法.

由于 WF 攻击的蓬勃发展,相应的防御手段也应运而生[45],但大多数防御技术的实用性较差[46-47],或仅针对某一个具体的 WF 攻击模型进行防御,应用范围不广[48].BuFLO 家族(BuFLO[49],CS-BuFLO[50],Tamaraw[51])对 WF 进行了有效的阻截,但是消耗过多的网络带宽和增加较多的传输延迟.近年来基于神经网络方法提出了对抗样本模型,基于误导攻击者将该网页流误导分类至另一个网站的思想实施防御[52-53],但是该方法的假设前提过强,实际可操作性较低.目前相对可用的 WF 防御机制是 WTF-PAD[54]和 Walkie-Talkie(W-T)[55],但本文在实验部分会验证模型可以有效攻破这 2 个防御机制.

2 基于 burst 深度分析的网站指纹攻击模型

基于当前神经网络方法与面向 Tor 匿名网络的 WF 攻击技术结合不足的问题,根据 burst 特征在基于 Tor 网络的网站访问流量中具有强显性的特点,设计了基于深度分析 burst 特征的网站指纹攻击模型(DBF).本节首先对模型的重要元素进行定义,然后给出模型的整体框架,最后对 DBF 模型的 2 个重要部分 DBF-CW 和 DBF-OW 进行阐述和分析.

2.1 模型基本元素的定义

在对本文提出的 DBF 模型进行分析前,需要对网站指纹(WF)攻击技术的重要元素进行介绍,符号定义如表 3 所示,其中 4 个重要的定义如下:

定义 1. 网站集(website set).网站集分为被监控网站集和非监控网站集.被监控网站集是由网络管理员定义的禁止用户访问的网站集,以 MW 表示;而非监控集则为真实网络中除监控集以外的所有网站,以 UW 表示.

WF 模型的任务是分析内网中是否存在用户正在利用匿名网络访问被监控网站,甚至进一步分析用户访问的是哪一个被监控网站,2 个目的分别对应于 WF 模型验证及测试阶段的开放世界场景(OW)和封闭世界场景(CW).如表 3 所示,MW 的大小为 N_s,UW 的大小在真实网络中为无限大,而在模型实验阶段是有限的,实验会采集一个尽可能大的数据集以模拟真实环境,至少保证 UW 的大小远大于 MW 的大小.

Table 3 Concepts and Symbol Definitions of WF Model

表 3 WF 模型的相关概念及符号定义

No.	Notion	Symbol	Description
1	Monitored Website Set	MW	$MW=\{mw_1,mw_2,\cdots,mw_{N_s}\}$, mw_i is the i-th monitored website.
2	Size of MW	N_s	The size of the monitored website set
3	Unmonitored Website Set	UW	$UW=\{uw_1,uw_2,\cdots\}$, uw_i is the i-th unmonitored website and the size of the set is infinite in reality.
4	Website Instance	I	I is the website trace instance set. $I_{(i)}$ is the trace instance set of i-th website. $\boldsymbol{I}_{(i)}^{(j)}$ is the j-th trace instance of $I_{(i)}$. $\boldsymbol{I}_i$ is the i-th instance in I.
5	Instance Feature	$\boldsymbol{F}$	$\boldsymbol{F}_{(i)}^{(j)}$ is the feature vector of $\boldsymbol{I}_{(i)}^{(j)}$. $\boldsymbol{F}_i$ is the feature vector of $\boldsymbol{I}_i$.
6	Instance Label	L	L is the website label set. l_i is the i-th label of L. $L^{(\mathrm{CW})}=\{l_1^{(\mathrm{CW})},l_2^{(\mathrm{CW})},\cdots,l_{N_s}^{(\mathrm{CW})}\}$ is the close-world website label set. $L^{(\mathrm{OW})}=\{l_0^{(\mathrm{OW})},l_1^{(\mathrm{OW})}\}$ is the open-world website label set.
7	Result Vector	$\boldsymbol{R}$	$\boldsymbol{R}$ is the set of vectors processed from I through neural network. $\boldsymbol{R}_{(i)}^{(j)}$ is the vector corresponding to $\boldsymbol{I}_{(i)}^{(j)}$. $R_{(i)}^{(j)}[k]$ is the value of the k-th dimension of $\boldsymbol{R}_{(i)}^{(j)}$. $\boldsymbol{R}_i$ is the vector corresponding to $\boldsymbol{I}_i$. $R_i[k]$ is the value of the k-th dimension of $\boldsymbol{R}_i$. The dimension of $\boldsymbol{R}_i$ is N_s.

定义 2. 网页流实例(instance).是用户对单个网站访问一次所产生的流量,是 WF 模型训练和分析的数据基本单元,$\boldsymbol{I}_i$ 表示实例集 I 中的第 i 个实例,$\boldsymbol{F}_i$ 表示实例 $\boldsymbol{I}_i$ 用于模型输入的特征向量.

由于在不同条件下(如网络状态、随机的广告内容、数据更新、不可预测的资源序列等)产生的流量序列不同,同一网站可以产生多个不同的实例[36],因此一个网站有多个不同的实例供模型训练使用,第 i 个被监控网站的第 j 个实例数记为 $\boldsymbol{I}_{(i)}^{(j)}$,相应的特征向量记为 $\boldsymbol{F}_{(i)}^{(j)}$.

定义 3. 网站标记(website label).是网站类别的标识,是 WF 模型的分类标记.其中封闭世界场景标记(CW)集中的每一个标记分别对应于被监控网站集中的一个网站,为 N_s 类标记;开放世界场景标记(OW)集为二类标记,即被监控网站类标记和非监控网站类标记.实例 $\boldsymbol{I}_i$ 的 2 种标记分别记为 $l^{(\mathrm{CW})}(\boldsymbol{I}_i)$ 和 $l^{(\mathrm{OW})}(\boldsymbol{I}_i)$,以 $l(\boldsymbol{I}_i)$ 泛指 $\boldsymbol{I}_i$ 的 2 种标记.

定义 4. 指纹向量(fingerprinting vector).即神经网络的结果向量(result vector),由神经网络方法自动学习特征形成并输出,用于识别网站标记.实例 $\boldsymbol{I}_i$ 的指纹向量记为 $\boldsymbol{R}_i$,$R_i[k]$为向量第 k 维的值.

2.2 DBF 模型框架

封闭世界场景假设(CW)和开放世界场景假设(OW)是 WF 攻击技术研究中 2 个重要的场景验证.DBF 模型由 DBF-CW 和 DBF-OW 这 2 个子模型构成,如图 3 所示.DBF-CW 基于深度神经网络对被监控网站的网页流 burst 特征进行深度分析和学习,输出网页流的指纹向量,若网页流属于被监控网站集,则利用指纹向量可直接得到该被监控流的网站域名 CW 标记.CW 标记为多分类标记,每一类为一个具体的网站域名.以往的研究通常仅训练一个 WF 模型同时用于 2 个场景,在 OW 场景中对模型输出的指纹向量基于阈值判断的方式实现二分类决策.DBF-OW 同样是基于 DBF-CW 输出的指纹向量进行再分析,但放弃了阈值法的使用,而是利用随机森林(RF)算法对被监控网站流和非监控流进行二分类特性学习以构建模型,在 OW 场景下实现二分类获取流的 OW 标记,即识别该网页流是否属于被监控网站集,OW 是二类标记,即被监控网站标记和非监控网站标记.

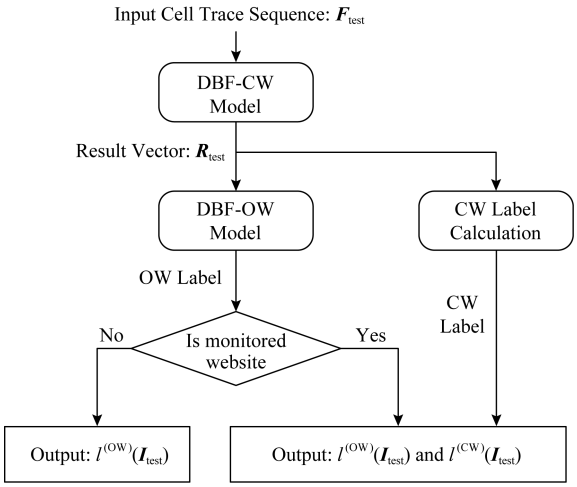

Fig. 3 The framework of DBF

图 3 DBF 模型框架

在模型训练阶段,对于 CW 场景,DBF-CW 与常规 WF 模型相同,使用被监控网站流的训练数据特征和标记对模型进行训练,即给定训练实例集 I 和标记集 L,基于特征设计提取各实例 $\boldsymbol{I}_{(i)}^{(j)}$ 的特征值 $\boldsymbol{F}_{(i)}^{(j)}$,对初始神经网络 NN 进行训练.如式(1)所

示,m_{N_s} 是标记为 $l_{N_s}^{(CW)}$ 的被监控网站训练实例数.而在 OW 场景中,不同于一般 WF 模型仅需要被监控网站集进行模型训练,DBF 还需要非监控网站集数据训练模型.DBF-OW 首先依照 CW 场景训练 DBF-CW 模型,然后使用 DBF-CW 输出被监控集和非监控集训练数据的指纹向量,$\boldsymbol{R}_{(i)}^{(j)} \leftarrow DBF\_CW(\boldsymbol{F}_{(i)}^{(j)})$,再基于这些指纹向量和对应的二分类标记训练初始的随机森林模型,如式(2)所示,m_0' 和 m_1' 分别是标记为 $l_0^{(OW)}$ 和 $l_1^{(OW)}$ 的训练实例数.

$$DBF\_CW \leftarrow NN(\boldsymbol{F}_{(1)}^{(1)}, l_1^{(CW)}; \cdots; \boldsymbol{F}_{(i)}^{(j)}, l_i^{(CW)}; \cdots; \boldsymbol{F}_{(N_s)}^{(m_{N_s})}, l_{N_s}^{(CW)}), \tag{1}$$

$$DBF\_OW \leftarrow RF(\boldsymbol{R}_{(0)}^{(1)}, l_0^{(OW)}; \cdots; \boldsymbol{R}_{(0)}^{(m_1')}, l_0^{(OW)}). \tag{2}$$

在模型验证和测试阶段,对于 CW 场景,DBF-CW 与常规基于神经网络的 WF 模型相同,输入待测试的被监控网页流实例 $\boldsymbol{I}_{test}$ 的特征向量 $\boldsymbol{F}_{test}$,提取指纹向量 $\boldsymbol{R}_{test}$,$\boldsymbol{R}_{test} \leftarrow DBF\_CW(\boldsymbol{F}_{test})$,进一步得到被监控网站 CW 标记 $l^{(CW)}(\boldsymbol{I}_{test}) = \arg\max(\boldsymbol{R}_{test})$,即实例标记 $l^{(CW)}(\boldsymbol{I}_{test})$ 为 $\boldsymbol{R}_{test}$ 中向量值最大对应的维度位序.对于 OW 场景,区别于一般神经网络方法人工设定一个阈值 Th,只有当 $\boldsymbol{R}_{test}[\arg\max(\boldsymbol{R}_{test})] > Th$ 时,实例 $\boldsymbol{I}_{test}$ 才被判定为被监控网页流,否则为非监控网页流的思路,DBF 在 DBF-CW 提取出指纹向量的基础上,DBF 的子模型 DBF-OW 基于随机森林算法分析指纹向量 $\boldsymbol{R}_{test}$ 各维度值的关联性和潜在规律得到实例 $\boldsymbol{I}_{test}$ 的 OW 标记,即 $l^{(OW)}(\boldsymbol{I}_{test}) \leftarrow DBF\_OW(\boldsymbol{R}_{test})$.

在 WF 模型应用到实际中时,模型首先基于 OW 场景分析网页流是否属于被监控网站集,若是则进一步基于 CW 场景分析网页流所属的具体网站域名.具体而言,模型首先基于 DBF-CW 计算获取指纹向量,并基于 DBF-OW 对指纹向量的分析得到网页流的 OW 标记,若流的 OW 标记为被监控网站,则进一步基于指纹向量分析流的 CW 标记,即识别流的具体网站域名,如图 3 所示.

2.3 封闭世界场景模型 DBF-CW

2.3.1 burst 特征

Tor 网络流量对数据包进行了加密,且载荷被封装成固定的 512 B 的 cell 进行传输,因此 Tor 网络流量可利用的主要特征只有时间戳、数据传输方向以及数据包数.基于对传输方向的分析,一次网页访问所产生的流量实例 $\boldsymbol{I}_{(i)}^{(j)}$ 可构建出一个 cell 流序列特征 $\boldsymbol{F}_{(i)}^{(j)} = (1, -1, -1, \cdots, -1)$,其中 1 表示出包方向,−1 表示入包方向.burst 特征即为方向相同的连续流序列的长度,即对于序列(1,−1,−1,−1,1),burst 特征值为 3(连续 3 个−1),位置序列坐标为(2,4).burst 在网站指纹攻击技术中是重要的上层特征表现,其揭示了网页资源加载等数据传输关系.在无法提取其他有用特征信息的 Tor 网络中,burst 特征显得更为重要.

2.3.2 burst 特征深度分析的神经网络原理

一维卷积神经网络对序列具有较好的分析效果,而且相比循环网络,运行速度更快.卷积网络基于卷积层和池化层的叠加,使得卷积窗口能覆盖到越来越多的局部序列信息,并提取到越来越深度抽象的序列特性,其卷积原理如图 4 所示.卷积网络的卷积核可用于提取网页流序列的 burst 特征,并通过更深层的卷积和池化运算得到序列中 burst 位置的抽象相关特性.Tor 流量的 burst 特征有长有短,利用卷积核大小不同的卷积层对不同长度的 burst 特征进行提取,进而利用深层网络对不同长度 burst 的位置分布进行分析,能较有效地分析 Tor 流量的 burst 特征,解构 Tor 流量特性.深度神经网络对高维向量具有较好的分析效果,基于卷积网络输出的高维向量,DNN 可以实现对向量各维度间复杂的相关性分析,如图 5 所示.

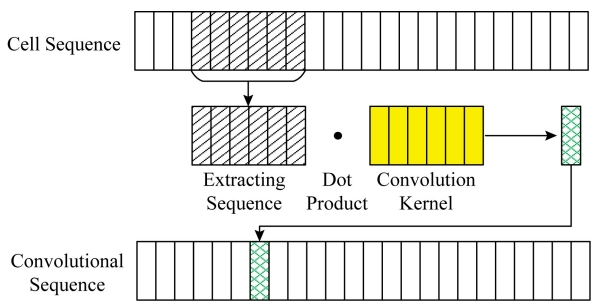

Fig. 4 Schematic diagram of one-dimensional convolution operation

图 4 一维卷积运算示意图

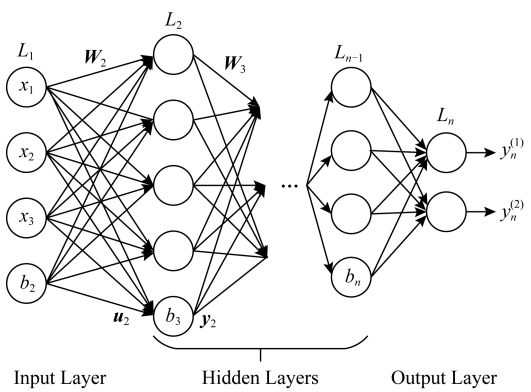

Fig. 5 Schematic diagram of dense neural network

图 5 密集神经网络示意图

2.3.3 DBF-CW 的神经网络结构设计

DBF-CW 由 burst 提取模块、burst 抽象学习模块和 burst 深度分析模块三大模块构成，主要由卷积层(convolution layer, Conv)、最大池化层(max pooling layer)、密集层(dense layer)、批标准化处理(batch normalization)和 Dropout 处理这 5 个基本层件组成，如图 6 所示.批标准化处理有助于神经网络参数的快速训练；Dropout 处理则有利于提高模型的泛化性，丢失率越高，模型越不容易过拟合，但丢失率过高会大大降低模型的性能.

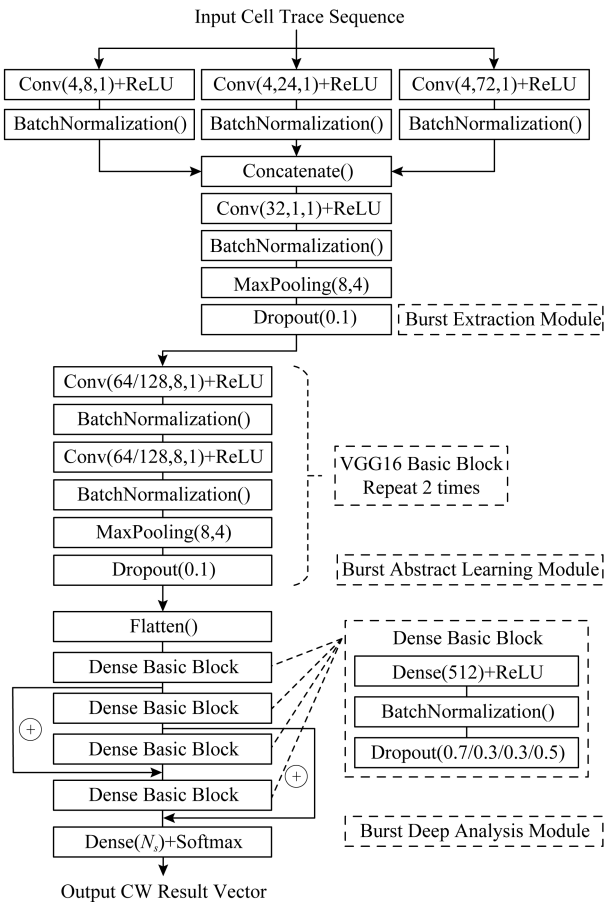

Fig. 6 The neural network structure of DBF-CW

图 6 DBF-CW 神经网络结构

模块 1 为 burst 特征提取模块，主要作用和功能是利用不同大小的卷积核对短、中、长 burst 进行提取，并对 burst 在序列中的位置进行简单的定位和浅层分析.定义短、中、长 burst 长度依次为 8，24 和 72，后者依次为前者的 3 倍长度.基于该定义，模型对不同长度的 burst 分别采用了 4 个与其长度对应大小(即 8，24 和 72)的卷积核进行提取，然后将得到的 3 个卷积张量在通道维度轴上进行拼接(concatenate)，形成通道轴为 12 维的卷积张量.拼接后的张量进入有 32 个大小为 1 的卷积核的卷积层中进行学习，大小为 1 的卷积核的主要作用是学习卷积张量在通道维度轴上的通道向量各维度之间的规律和相关性，分析定位 burst 在序列上可能出现的单点位置.最后采用一层最大池化层加快卷积网络对局部特征的学习效率.DBF-CW 使用的池化层均为最大池化层，且池化窗口大小与短 burst 长度一致，步进长度为短 burst 长度的一半.

模块 2 为 burst 抽象学习模块，主要作用是对第 1 模块输出的浅层卷积张量实施更加抽象和深度的学习，从局部特征的学习逐渐过渡到全局概念的学习，以挖掘不同类网页流序列 burst 特征的深层抽象特性和概念.该模块由经典 CNN 架构 VGG16 的 2 个基本区块构成，该基本区块由 2 层卷积层和一层最大池化层组成，在充分利用卷积运算对特征规律学习的同时，保证了网络的学习效率.第 1 个 VGG16 基本区块的卷积核数为 64，是模块 1 卷积层的 2 倍；第 2 个 VGG16 基本区块的卷积核数为 128，是上一个基本区块的 2 倍.随着卷积网络层的深入，卷积核数的增加有助于学习到不同类网页流 burst 特征的深层概念.burst 抽象学习模块的卷积窗口大小均与定义的短 burst 长度一致，步进长度均为 1.

模块 3 为 burst 深度分析模块，主要作用是将上一模块输出的具有 burst 特性深度和全局概念意义的卷积张量铺平形成向量，并基于密集神经网络对该向量的各维度相关性和特征规律进行分析，以进一步挖掘上一模块所提取出的各个全局特征的关系.模块 3 由 4 个密集基本区块构成，密集基本区块由一层全连接层、一层批标准化层和一层 Dropout 层组成，全连接层的神经元数均为 512.同时，burst 深度分析模块还基于残差连接的思想，将第 1 和第 3、第 2 和第 4 基本区块的输出进行残差相加，以缓解特征向量信息随着网络层的增加而丢失和遗忘的问题.

模型采用 RMSProp 算法训练网络，批处理大小 batch 为 128，采用交叉熵计算分类损失，模型评估指标为准确率(accuracy, Acc).

2.4 开放世界场景模型 DBF-OW

DBF-OW 模型基于随机森林(RF)算法，对 DBF-CW 输出的指纹向量 $\boldsymbol{R}_i$ 进行分析.随机森林是基于结构和参数简单的决策树等弱分类器的集成模型，对中低维的特征向量具有良好的分析效果.如图 7 所示，DBF-CW 结果向量在进入 RF 模型训练前，DBF-OW 先计算向量 $\boldsymbol{R}_i$ 各维度值的 3 个统计特征.

结果向量各维度值的统计分布是反映向量属性的重要特征,对模型的分类决策具有影响力.3 个统计特征如式(3)~(5)所示,DBF-OW 通过计算 $\boldsymbol{R}_i$ 的最大维度值、熵和标准差,得到 $\boldsymbol{R}_i$ 各维度值的分布情况,并将这 3 个统计特征添加到 $\boldsymbol{R}_i$ 中,形成 N_s+3 维的特征向量.新的特征向量与其对应的二分类标记输入到 RF 模型中进行规律学习,最终得到一个可识别未知网页流实例的二分类标记的开放世界模型.

$$\max(\boldsymbol{R}_i)=R_i[k],\ R_i[k]\geqslant R_i[j],\ 0\leqslant j\leqslant N_s-1, \tag{3}$$

$$\mathrm{std}(\boldsymbol{R}_i)=\sqrt{\frac{1}{N_s}\sum_{k=0}^{N_s-1}(R_i[k]-\overline{\boldsymbol{R}_i})^2},\quad \overline{\boldsymbol{R}_i}=\frac{1}{N_s}\sum_{k=0}^{N_s-1}R_i[k], \tag{4}$$

$$Ent(\boldsymbol{R}_i)=-\sum_{k=0}^{N_s-1}R_i[k]\times \mathrm{lb}\,R_i[k]. \tag{5}$$

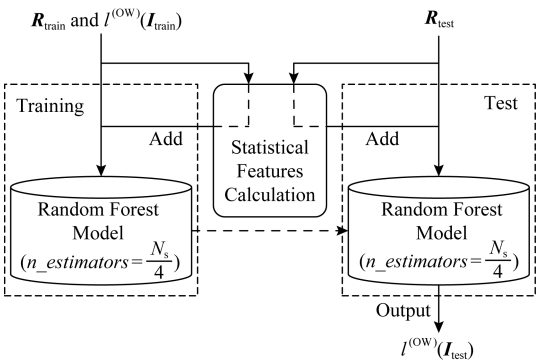

Fig.7 The structure of DBF-OW

图 7 DBF-OW 模型结构

DBF-OW 所基于的随机森林由若干决策树构成,每个决策树的训练、结构和参数相互独立、各不相同.每个决策树在分析训练数据时,以指纹向量某一维度的属性值作为基准对数据进行划分,计算分类前和分类后的信息熵差值,以此得到以不同维度轴作为划分基准的各数据划分方法的信息增益,以信息增益最大的分类方法作为该"树支"的分类逻辑.训练数据被划分成多个部分后,决策树对各部分数据分别继续分析,形成新的分支逻辑,以此类推,最终形成一个有若干分支的决策树.信息熵、信息增益及划分基准选择的计算如式(6)~(8)所示:

$$Ent(D)=-\sum_{i=1}^{|y|}p(x_i)\times \mathrm{lb}\,p(x_i), \tag{6}$$

$$G(D,a)=Ent(D)-\sum_{j=1}^{J}\frac{|D^j|}{D}\times Ent(D^j), \tag{7}$$

$$a_*=\arg\max_{a\in A}G(D,a), \tag{8}$$

其中,$Ent(D)$表示原始数据集 D 的信息熵,$|y|$是数据的类别数,$p(x_i)$表示第 i 类数据占整个数据集的比例;$G(D,a)$表示以指纹向量第 a 维度作为划分基准时的信息增益,J 表示此时的分支数,D^j 表示被划分到第 j 个分支的数据;a_* 表示被选择的基准维度,即信息增益最大的指纹向量维度.

在各决策树训练完毕后,决策树的所有叶结点由该结点训练数据的大多数类作为该结点的类别.决策树在对新的数据点指纹向量进行分析时,新向量依照决策树的逻辑分支分配到某个叶结点,该叶结点对应的类别即决策树对该向量的类别预测.在所有决策树都对新数据点的指纹向量进行类别预测后,随机森林对各决策树的预测结果进行集成和综合分析,以投票的方式决定数据点的类别,如式(9)所示:

$$L(x)=\arg\max_{y\in Y}\sum_{t=1}^{T}\prod(c_t(x)==y), \tag{9}$$

其中,$c_t(x)$表示第 t 个决策树对 x 的预测结果;T 是随机森林模型中决策树的个数;Y 是标签集;派函数 $\prod()$ 表示当括号内条件为真时函数值为 1,否则为 0.因此式(9)的含义是对于标签集 Y 中的每一个元素标记 y,将随机森林模型 T 中的每一棵树 t 的预测结果 $c_t(x)$与 y 进行比较,当结果为真时对 y 的预测值加 1,最后通过反向最大值函数输出具有最大预测值的 y 值,即为随机森林模型对数据 x 的标记预测结果.随机森林以决策树为基础,通过各决策树对指纹向量的学习,分析向量各维度的相关性和潜在规律,获取指纹向量的属性逻辑规则,对应于决策树的每一条路径.

随机森林作为一个集成模型,子分类器的个数是一个重要的参数.由于结果向量的维度会随着被监控网站集的大小而变化,DBF-OW 设定子分类器数为 $N_s/4$,即被监控网站集大小的四分之一.RF 子分类器数随着被监控网站集的大小而变化,有利于 RF 模型对数据进行充分的拟合,避免欠拟合的情况发生.

3 实验与结果

3.1 实验设置

实验主要分为 2 个部分,分别在封闭世界场景和开放世界场景下对模型性能进行评估.采用了微星(MSI)GT63 作为实验机器,包含了 6 个 Intel® Core™ i7-8750H@2.2 GHz 的 CPU 和一个 NVDIA

GeForce GTX 1070 的 GPU,机器内存为 32 GB.实验中的算法代码均基于 Keras 实现,DF[22] 和 AWF[11] 作为实验的对比模型.由于实验所使用的数据集只有包方向序列特征,k-FP[16],k-NN[36] 和 CUMUL[17] 等需要分析时间特征的算法无法在该实验条件下执行,这些模型的实验对比结果来源于与数据集或模型相关的论文.

3.2 评估指标

封闭世界场景是一个多分类任务,在该场景下模型的分类性能主要体现在对不同网页流的分类能力上,因此采用准确率(Acc)对模型性能进行评估:

$$Acc=\frac{1}{N}\sum_{i=1}^{N_s}TP_i, \tag{10}$$

其中,TP_i 表示第 i 类网页流被正确分类的实例数,N 表示参与评估的实例总数.

开放世界场景是一个二分类任务,在该场景下模型的分类性能不仅体现在能正确识别出受监控网页,还体现在尽可能少地将非监控网页误识别成监控网页.实验采用了真阳性率(true positive rate, TPR)、假阳性率(false positive rate, FPR)和多类真阳性率(multi-TPR, $MTPR$)对模型性能进行评估:

$$TPR=\frac{TP}{TP+FN}, \tag{11}$$

$$FPR=\frac{FP}{TN+FP}, \tag{12}$$

$$MTPR=\frac{\sum_{i=1}^{N_s}TP_i}{TP+FN}, \tag{13}$$

其中,TP 表示被监控网页流被正确分类的实例数,TN 表示非监控网页流被正确分类的实例数,FN 表示受监控网页流被错误分类为非监控网页流的实例数,FP 表示非监控网页流被错误分类为受监控网页流的实例数.在真实网络中非监控网页流要远多于被监控网页流,准确率和精度(precision)指标不能准确衡量模型性能,因此实验不采用这 2 个指标.

3.3 实验数据集

针对不同的实验目的,实验采用了多个基于 Tor 网络访问网站的数据集,数据集的每一条数据表示一个网页流实例的数据包方向序列,即(1,−1,−1,…,−1)的数据形式,序列长度均为 5 000 维,不足 5 000 维的部分以 0 补足.如表 4 所示,前缀为 CW 的数据集表示封闭世界数据集,前缀为 OW 的数据集表示封闭世界数据集;$N(MW)$ 表示被监控网站集的大小;$N(I_i)$ 表示各被监控网站的网页流

Table 4 Datasets Used in the Experiments

表 4 实验使用的数据集情况

Reference	Number	Dataset	$N(MW)$	$N(I_{(i)})(N_{\text{train-val}}+N_{\text{test}})$	$N(UW)(N_{\text{train-val}}+N_{\text{test}})$
Ref[11]	1	CW100	100	2 500(2 375+125)	
	2	CW200	200	2 500(2 375+125)	
	3	CW500	500	2 500(2 375+125)	
	4	CW900	900	2 500(2 375+125)	
	5	CW200-Time	200	2 500(1 900+100×6)	
	6	OW200	200	2 000(1 900+100)	400 000(380 000+20 000)
	7	OW200-Time	200	2 500(1 900+100×6)	400 000(380 000+20 000)
Ref[22]	8	CW-NoDef	95	1 000(900+100)	
	9	CW-W-T	100	900(850+50)	
	10	CW-WTFPAD	95	1 000(900+100)	
	11	OW-NoDef	95	1 000(900+100)	40 100(20 100+20 000)
	12	OW-W-T	100	900(810+90)	40 000(20000+20 000)
	13	OW-WTFPAD	95	1 000(900+100)	40 100(20 100+20 000)
Ref[16]	14	CW-Normal	55	100(70+30)	
	15	CW-HS	30	80(60+20)	
	16	OW-Normal	55	100(70+30)	100 000(70 000+30 000)
	17	OW-HS	30	80(60+20)	100 000(75 000+25 000)

实例数;$N(UW)$表示非监控网站集的大小,每个非监控网站的实例数均为 1;数据括号中的第 1 个数表示训练-验证集(train-val)的大小,第 2 个数表示测试集(test)的大小,训练-验证集和测试集的划分与源论文保持一致.所有数据的测试集仅用于模型最后的结果对比;在参数验证的实验中,验证集的大小始终保持为训练-验证集的 10%.

不同数据集的用处不尽相同.CW100-CW900 数据集的被监控网站集大小不同,可用于验证被监控网站集 *MW* 的大小对模型性能的影响.CW200-Time 和 OW200-Time 数据集采集了与训练数据间隔 3 d、10 d、2 周、4 周、6 周的被监控网站实例,可用于测试模型的抗概念漂移性能.Sirinam 数据集[22]用于验证模型对 W-T 和 WTFPAD 这 2 个相对成熟的 WF 防御机制的突破能力,CW-NoDef,CW-W-T,CW-WTFPAD 分别是在无 WF 防御、有 W-T 防御和有 WTFPAD 防御机制下采集的封闭世界数据集,OW-NoDef,OW-W-T,OW-WTFPAD 同理.Haye 数据集[1]可用于验证模型对 Tor 隐藏网站的检测能力,CW-Normal 和 CW-HS 是用户通过 Tor 网络分别访问普通网站和 Tor 隐藏网站所采集到的数据集,OW-Normal 和 OW-HS 同理.

3.4 封闭世界场景实验

封闭世界场景的实验目的,是检验 WF 攻击模型是否能正确分类被监控网页流实例所对应的被监控网站集标记,检验的是模型的多分类性能.实验主要分为参数验证和性能测试 2 部分.参数验证阶段主要探讨训练轮次 epoch、神经网络的输入序列长度、训练实例数对模型性能的影响;性能测试阶段主要分析被监控网站集 *MW* 的大小对性能的影响、模型的抗概念漂移能力、绕过 WF 攻击防御机制的能力以及检测 Tor 隐藏网站的能力.DBF-CW 与 DF 的默认参数是 epoch 为 30,输入序列长度为 5 000.AWF 的默认参数是 epoch 为 30,输入序列长度为 3 000.

3.4.1 epoch 对模型准确率的影响

实验在 CW100 和 CW-NoDef 数据集上对训练不同 epoch 下的模型准确率进行验证,训练集为训练-验证集的 90%,验证集为 10%.如图 8 和图 9 所示,图 8 为 DBF-CW 模型分别在 CW100 和 CW-NoDef 数据集上运行 60 个 epoch 的结果,图 9 为 DBF-CW,DF 和 AWF 模型在 CW100 数据集运行 30 个 epoch 的结果.尽管 CW100 和 CW-NoDef 数据集的大小不同,但当 epoch 为 15～20 时,DBF-CW 在 2 个数据集上均达到了拟合的状态,验证了 DBF-CW 训练的稳定性.同时,相比 AWF 模型,DBF-CW 和 DF 训练速度更快且更稳健,仅经过前 5 轮的训练,整体准确率已经稳定在 97%以上.

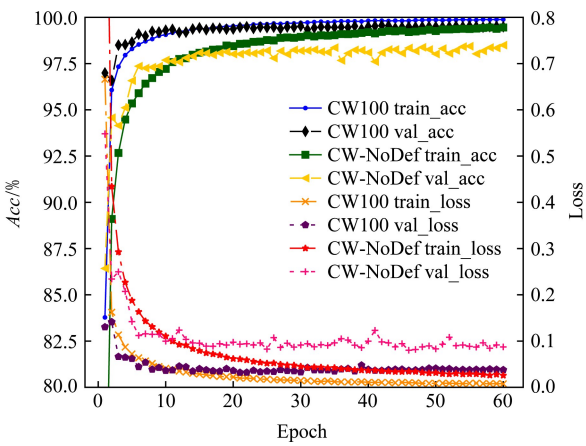

Fig. 8 Performance of DBF-CW under different epochs

图 8 DBF-CW 训练不同 epoch 时的性能

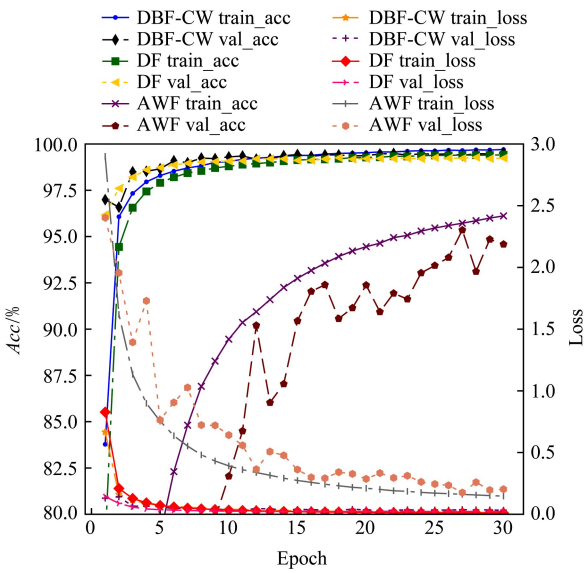

Fig. 9 Performance under different epochs on the CW100 dataset

图 9 各算法在 CW100 数据集上训练不同 epoch 的性能

3.4.2 网页流序列长度对模型准确率的影响

实验在 CW100 和 CW-NoDef 数据集上验证模型在输入的网页流序列长度不同时的准确率变化,训练集为训练-验证集的 90%,验证集为 10%.如图 10 所示,DBF-CW 和 DF 模型的准确率均随着输入序列长度的增大而增大,且在输入长度为 1 000 时,模型的验证准确率在 98%以上.相比 AWF 模型,DBF-CW 和 DF 模型对输入长度不敏感,准确率

变化幅度较小,表明模型对输入的长度依赖性不强,有较好的健壮性.

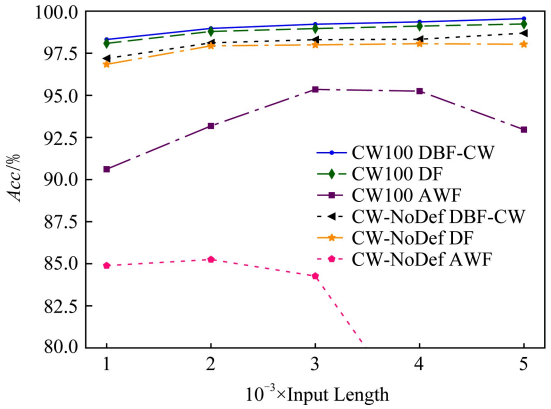

Fig. 10 Accuracy of the algorithms with different input lengths

图 10 各算法在输入序列长度不同时的准确率

3.4.3 训练实例数对模型准确率的影响

实验在 CW100 和 CW-NoDef 数据集上对模型的训练实例数与模型准确率之间的关系进行验证,验证集大小为训练-验证集的 10%,训练集大小依次为 10%～90%,间隔 10%,取 9 个点.实验结果如图 11 和图 12 所示,随着每类被监控网站的训练实例数增加,3 个算法模型的分类准确率均随之增大,但 DBF-CW 相比 AWF 的变化幅度小得多.在小样本训练的情况下,DBF-CW 和 DF 算法仍能保持 96%以上的分类准确率,表明算法对样本的规律学习和泛化性能比较好,在小样本训练的情况下同样可以成功实施 WF 攻击.

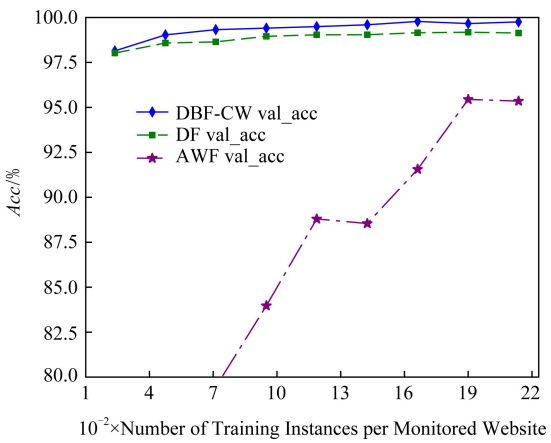

Fig. 11 Accuracy of the algorithms with different training instances on the CW100 dataset

图 11 各算法在 CW100 数据集上训练不同实例数的准确率

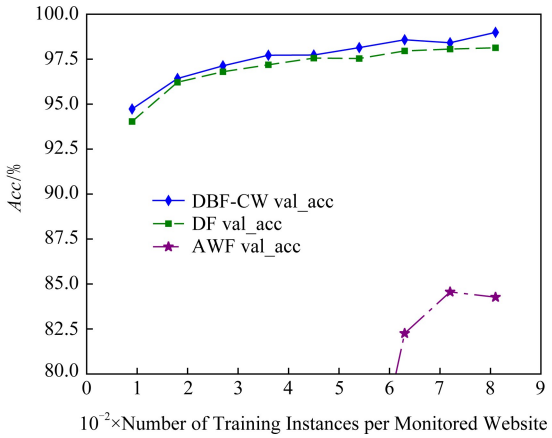

Fig. 12 Accuracy of the algorithms with different training instances on the CW-NoDef dataset

图 12 各算法在 CW-NoDef 集上训练不同实例数的准确率

3.4.4 被监控网站集大小对模型准确率的影响

实验在 CW100-CW900 四个数据集上验证被监控网站集的大小对模型准确率的影响,这 4 个数据集的网站集大小分别为 100,200,500 和 900.如图 13 和表 5 所示,随着被监控网站集的增大,DBF-CW

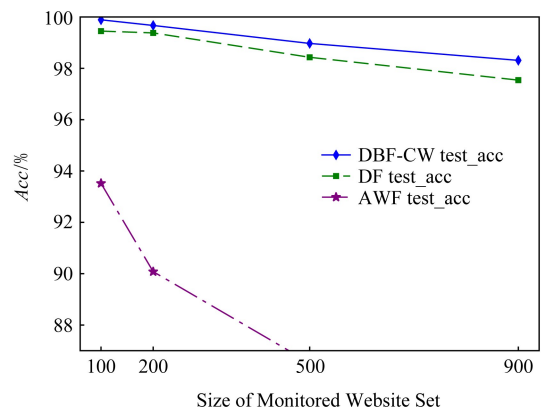

Fig. 13 Test accuracy of the algorithms on CW100-900 dataset

图 13 各算法在 CW100-900 数据集上的测试准确率对比

Table 5 Test Accuracy of the Algorithms on CW100-900 Dataset

表 5 各算法在 CW100-900 数据集上的测试准确率 %

Dataset	DBF-CW	AWF[11]	DF[22]	CUMUL[17]
CW100	99.89	93.52	99.45	97.72
CW200	99.67	90.08	99.38	97.23
CW500	98.97	86.53	98.43	95.74
CW900	98.31	82.82	97.54	92.76

和 DF 的准确率有略微下降,而 AWF 模型准确率下降较快.DBF-CW 的分类准确率始终保持在最高位,且均在 98%以上.实验表明 DBF-CW 是健壮的,对 WF 技术适应性较好,受被监控集网站大小的变化影响较小.

3.4.5 模型的抗概念漂移能力验证

实验采用 CW200-Time 数据集验证模型缓解概念漂移(concept drift)的能力.概念漂移是指在实际网络环境中,数据模式会随时间的推移而出现变化,模型训练使用的数据与测试数据的间隔越长,模型通过"旧"数据训练得到的概念与测试数据实际的概念模式的偏差就会越大,导致模型分类性能下降.

图 14 和表 6 是 DBF-CW 与对比算法在 CW-Time 数据集上的准确率对比,CW-Time 数据集包含 1 个训练集和 6 个测试集,各测试集的采集时间与训练集分别相隔了 0 d,3 d,10 d、2 周、4 周和 6 周.从图 14 可以看到,DBF-CW,DF 和 AWF 模型的分类准确率随着时间间隔增大而均有所下降,但 DBF-CW 的下降速度是最慢的,验证了 DBF-CW 能较有

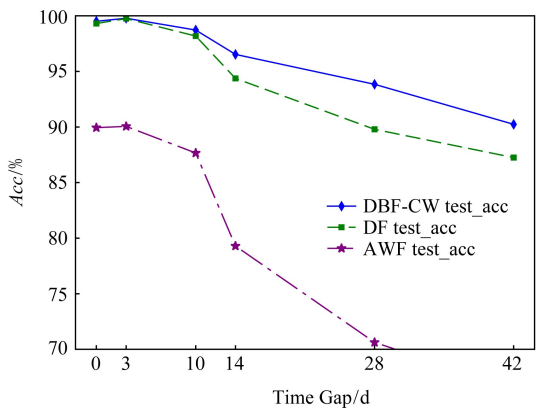

Fig. 14 Test accuracy of the algorithms on CW-Time dataset

图 14 各算法在 CW-Time 数据集上的测试准确率

Table 6 Test Accuracy of the Algorithms on CW-Time Dataset

表 6 各算法在 CW-Time 数据集上的测试准确率 %

Time Gap/d	DBF-CW	AWF[11]	DF[22]	CUMUL[17]
0	99.51	89.94	99.31	97.21
3	99.78	90.55	99.75	91.17
10	98.72	87.66	98.18	86.43
14	96.53	79.29	94.37	83.45
28	93.84	70.62	89.79	74.92
42	90.24	66.48	87.25	70.88

效地缓解概念漂移问题.概念漂移是实际应用中模型随着时间推移而性能下降的一个无法避免的问题,但如果模型能有效减缓性能下降的速度,就有更充分的时间准备新的训练数据以训练出新的模型,以真正解决实际应用场景中的概念漂移问题.

3.4.6 模型对 Tor 隐藏网站的检测性能

实验在 Tor 隐藏网站数据集上对模型的 Tor 隐藏网站检测能力进行测试.如表 7 所示,DBF-CW 在正常集 CW-Normal 和隐藏网站集 CW-HS 的准确率表现一般,分别为 70.6%和 80.66%.这可能是因为该数据集的训练实例数和序列长度过短导致的,各类被监控网站的训练实例数仅为 70 和 60,远远少于其他 2 个数据集的 900 训练实例和 2375 训练实例;另一方面,该数据集的序列为数据包序列,而不是其他 2 个数据集的 cell 序列,这会导致模型对 burst 特征的提取和分析不足.相比之下,基于一般机器学习方法的 k-FP[16] 在小样本情况下表现出了较强的学习能力.从纵向看,DBF-CW 在隐藏网站数据集上的分类准确率高于正常数据集约 10%,说明 DBF-CW 对 Tor 隐藏网站是有检测能力的.从横向上看,DBF-CW 相比其他 2 个神经网络模型的准确率是最高的,体现了 DBF-CW 的神经网络结构在 WF 领域有更强的适应性.

Table 7 Test Accuracy on Tor Hidden Website Dataset

表 7 各算法在 Tor 隐藏网站数据集上的测试准确率 %

Dataset	DBF-CW	AWF[11]	DF[22]	k-FP[16]
CW-Normal	70.60	43.45	64.06	93.97
CW-HS	80.66	48.66	75.45	81.91

3.4.7 模型对 WF 攻击防御机制的突破能力验证

实验在无针对 WF 攻击的防御机制、有 W-T 机制和有 WTFPAD 机制这 3 个数据集上进行.如表 8 所示,WTFPAD 和 W-T 防御机制牺牲了一定的带宽,分别为 31%和 64%,WTFPAD 机制还有 34%的传输延迟.从横向比较上看,DBF-CW 在 CW-NoDef,CW-W-T 和 CW-WTFPAD 这 3 个数据集上的准确率均为最高.对于 WTFPAD 防御机制,DBF-CW 对各被监控网站的识别准确率达到了 96.25%,表明 WTFPAD 对 DBF-CW 几乎没有防御能力.虽然 DBF-CW 在 W-T 防御机制数据集上的准确率只有 52.06%,但考虑到该数据集的被监控集大小为 100,该准确率仍能说明 DBF-CW 在一定程度上能够突破 W-T 防御机制.

Table 8 Test Accuracy of the Algorithms on Defense Against WF Attack Dataset

表 8 各算法在抵御 WF 攻击数据集上的测试准确率对比 %

Dataset	Overhead		DBF-CW	AWF[11]	DF[22]	k-FP[16]	k-NN[36]	CUMUL[17]
	Bandwidth	Latency						
CW-NoDef	0	0	98.77	81.69	98.00	95.52	95.03	97.37
CW-W-T	64	0	52.06	38.00	49.66	7.04	20.21	38.43
CW-WTFPAD	31	34	96.25	71.87	92.83	69.07	16.09	60.29

3.5 开放世界场景实验

开放世界场景的实验目的,是检验 WF 攻击模型是否能正确识别未知网页流实例为被监控网站流或非监控网站流,检验的是模型的二分类性能.实验主要分为参数验证和性能测试 2 部分.参数验证阶段主要探讨基于随机森林算法的 DBF-OW 子分类器数和非监控网站训练实例数对 DBF 模型性能的影响;性能测试阶段主要分析模型的抗概念漂移能力、绕过 WF 攻击防御机制能力以及对 Tor 隐藏网站的检测能力.实验中,DBF-OW 的子分类器为被监控集大小的 1/4,其余参数与封闭世界场景实验的设置保持一致.

3.5.1 DBF-OW 子分类器数对模型准确率的影响

实验在 OW-NoDef 和 OW200 数据集上对由不同子分类器构建的 DBF-OW 模型性能进行验证,训练集为训练-验证集的 90%,验证集为 10%.如图 15 和图 16 所示,图 15 为 DBF-OW 模型在 OW-NoDef 数据集上运行的结果,实验选取了子分类器数分别为 10～210(间隔为 20)的 11 个模型进行评估;图 16 为 DBF-OW 模型在 OW200 数据集上运行的结果,选取了子分类器数分别为 10～410(间隔为 40)的 11 个模型进行评估.从对比结果上看,2 个实验分别在分类器数为 30 和 50 时性能达到相对最优,此后模型性能几乎没有增长.需要注意的是,30 和 50 个分类器分别约是各自所使用数据集的被监控网站集大小($N(MW)$)的 31%和 25%.因此,该实验验证了 DBF-OW 模型在分类器数取为被监控网站集大小的 1/4 时,性能能够达到一个相对较好的水平.

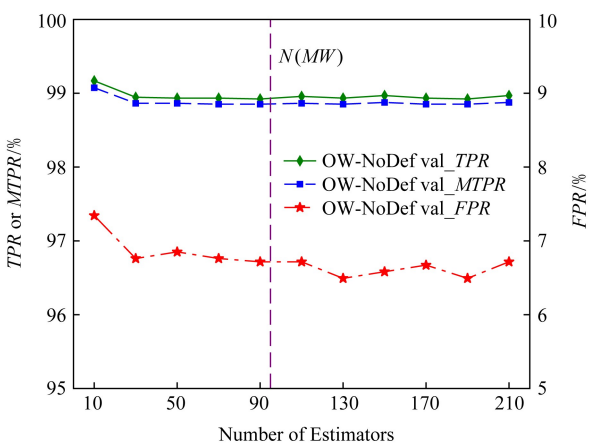

Fig. 15 Performance of DBF with different number of estimators on OW-NoDef

图 15 DBF 在 OW-NoDef 上子分类器个数不同时的性能

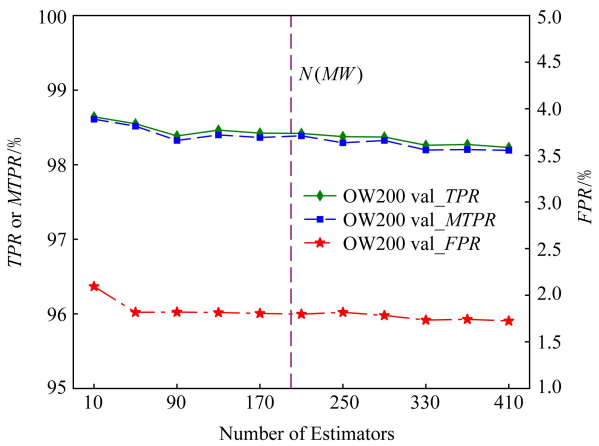

Fig. 16 Performance of DBF with different number of estimators on OW200

图 16 DBF 在 OW200 上子分类器个数不同时的性能

3.5.2 DBF-OW 有效性验证

实验在 OW-NoDef 和 OW200 数据集上通过比较 DBF-OW 和阈值法的性能以验证 DBF-OW 模型思想的有效性,训练集为训练-验证集的 90%,验证集为 10%.如表 9 所示,DBF-OW 模型在 OW-NoDef 数据集上的 TPR 与 $MTPR$ 值要优于阈值法,而在 OW200 数据集上的 TPR 与 $MTPR$ 值与阈值法持平,表明 DBF-OW 相比阈值法对正类的检测率有所提高,但提升水平有限.而对于 FPR 值,阈值法在 2 个数据集上的表现均大于 15%,表明阈值法将反类误分类为正类的问题较为严重,而 DBF-OW 的 FPR 分别仅为传统阈值法的 43%和 11%,表明 DBF-OW 有效缓解了该问题的出现,改进了阈值法的缺陷.

3.5.3 非监控网站训练实例数对模型准确率的影响

实验在 OW200 数据集上对模型的非监控网站训练实例数与模型性能之间的关系进行验证.实验使用数量固定的被监控网站实例数和数量不定的非监控网站实例数对 DBF 模型进行训练.被监控网站训练实例数为训练-验证集中被监控集的一半,即 190 000 条数据,非监控网站训练实例数依次取训练-验证集中非监控集的 10%～90%,间隔 10%,共 9 个点.实验使用 10%的训练-验证集(含监控集和非监控集,且与训练数据不重复)作为验证数据.如图 17 所示,随着非监控网站训练实例的增多,模型的 *TPR*,*MTPR* 和 *FPR* 均有所下降.但整体上看,DBF 在训练数据不平衡的情况下,性能依旧是稳健的:在非监控网站训练实例数约为被监控数的 20%时,*FPR* 只有 4.5%;而在非监控数为被监控数 1.8 倍时,DBF 的 *TPR* 和 *MTPR* 仍旧保持在 97%以上.

Table 9 Performance of DBF-OW and Threshold Method

表 9 DBF-OW 与阈值法的性能对比 %

Dataset	Method	*TPR*	*MTPR*	*FPR*
OW-NoDef	DBF-OW	99.00	98.88	6.76
	Threshold	98.02	98.02	15.57
OW200	DBF-OW	98.44	98.35	1.70
	Threshold	98.38	98.37	15.52

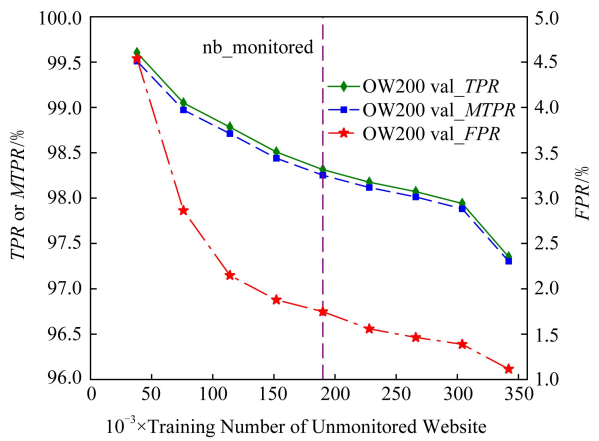

Fig. 17 Performance comparison of DBF with different number of training instances of unmonitored website

图 17 DBF 在非监控网站训练实例数不相同时的性能对比

3.5.4 模型的抗概念漂移能力验证

实验采用 OW200-Time 数据集验证模型在开放世界场景下缓解概念漂移的能力.OW200-Time 的被监控网站集部分与 3.4.5 节中使用的 CW200-Time 数据集完全相同,非监控网站集部分与 OW200 完全相同.由于实验重点关注的是模型对被监控网站类的学习是否会随着时间的变化与实际的类概念发生偏差,而不关心非监控网站是否出现概念漂移(各非监控网站实例只有一个,实际上构不成概念),因此实验的非监控网站集没有和被监控集一样间隔多天采集一次,所以测试集中的非监控集部分没有变化,如表 10 所示 *FPR* 始终为 1.63%.如图 18 和表 10 所示,模型性能随着时间间隔的增大,有较明显的下降.相比 3.4.5 节在封闭世界场景下验证模型抗概念漂移能力的实验,模型在开放世界场景下的性能下降得更快.但总的来说,模型在使用 42 d 前的数据进行训练时仍能达到 80%的 *TPR*,表明模型具有较强的抗概念漂移能力.从实践的角度分析,6 周的时间足够网络管理员采集新的数据训练模型.

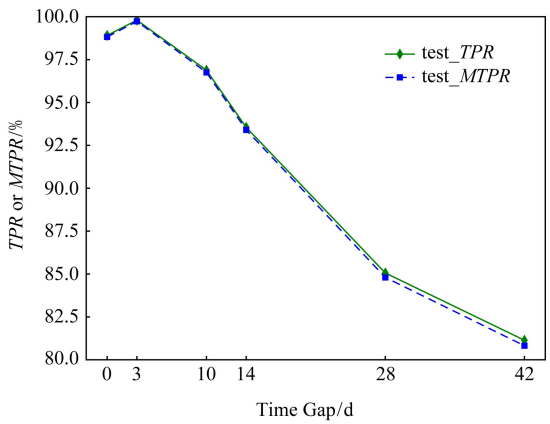

Fig. 18 DBF performance on the OW-Time dataset

图 18 DBF 在 OW-Time 数据集上的性能表现

Table 10 DBF Performance on the OW-Time Dataset

表 10 DBF 在 OW-Time 数据集上的性能表现 %

Time Gap/d	*TPR*	*MTPR*	*FPR*
0	98.91	98.82	1.63
3	99.80	99.73	1.63
10	96.88	96.75	1.63
14	93.55	93.40	1.63
28	85.06	84.80	1.63
42	81.14	80.82	1.63

3.5.5 模型对 Tor 隐藏网站的检测能力

实验在 Tor 隐藏网站数据集上对模型的 Tor 隐藏网站检测能力进行测试,该数据集的被监控网站集部分与 CW-Normal 和 CW-HS 一致.如表 11 所示:

Table 11 Performance on Tor Hidden Website Dataset

表 11 各算法在 Tor 隐藏网站数据集上的性能测试对比 %

Dataset	Metric	DBF	AWF[11]	DF[22]	k-FP[16]
OW-Normal	*TPR*	68.42	5.57	38.31	84.27
	FPR	0.57	6.59	0.33	0.66
OW-HS	*TPR*	85.33	11.98	77.99	73.31
	FPR	0.13	35.04	0	0.03

DBF 对 Tor 隐藏网站的检测效果是最好的,在 *FPR* 只有 0.13 的情况下 *FPR* 达到了 85.33%,在各类监控网站训练实例只有不到 100 的情况下,其性能比一般机器学习 k-FP[16] 还要出色.相比 3.4.6 节在封闭世界场景下 DBF-CW 检测 Tor 隐藏网站较弱的准确率表现,DBF 在开放世界场景下对 Tor 隐藏网站的识别有了很大的提高,而 2 个实验的被监控集是相同的.出现这种的可能原因是 DBF-OW 起到了重要的作用.不同于 AWF[11] 仅使用被监控网站集训练以及 DF[22] 同时使用被监控集和非监控集及相应的多分类标记同时训练模型,DBF 的子模型 DBF-OW 使用二分类标记训练模型,使得 DBF-OW 能够学习隐藏网站及非隐藏网站的二类特性.另外,不同于人为随机定义的被监控网站集,其整体的规律性比较弱,Tor 隐藏网站作为一种特殊的网页流天然地自成一类网页流,因此 Tor 隐藏网站和非 Tor 隐藏网站具有可以学习的网页流规律.实际上,在该实验中 DBF 的 *MTPR* 只有 66%,远低于 *TPR* 值 85.33%,从反向的角度也证明了 DBF-OW 在识别 Tor 隐藏网站中起到的重要作用.

3.5.6 模型对 WF 攻击防御机制的突破能力验证

实验在无防御机制、有 W-T 机制和有 WTFPAD 机制这 3 个开放世界数据集上进行.如表 12 所示,DBF 在 WTFPAD 数据集上对各被监控网站的 *MTPR* 和 *TPR* 分别达到了 92.16% 和 93.66%,WTFPAD 对 DBF 几乎没有防御能力,与 3.4.7 节在封闭世界场景下的结果相呼应;DBF-CW 在 W-T 数据集上的 *TPR* 到达了 93.92%,但 *MTPR* 为 64.11%,超高的 *TPR* 值与 3.5.5 节中的实验结果类似,这同样归功于 DBF-OW 对二类特性的学习能力.综合来看,DBF 在一定程度上绕过了 W-T 防御机制.与其他算法对比,DBF 在 3 个数据集上的 *MTPR* 和 *TPR* 均为最高,且有较高的性能优势.但 DBF 在 2 个数据集上的 *FPR* 均超过了 15%,在非监控网页流远远少于被监控网页流的真实网络中,这个 *FPR* 值是过高的,其主要原因是非监控网站集的训练实例数(20 000)较少于监控集训练数(90 000)且防御机制对模型起到了干扰作用.但在与对比算法的横向比较上,DBF 的 *FPR* 性能也不具备太大优势,说明 DBF-OW 在分析经过防御机制加持的 Tor 流量时还存在一定问题,仍需要继续改进.

Table 12 Performances on Defense Against WF Attack Datasets

表 12 各算法在抵御 WF 攻击数据集上的性能对比 %

Dataset	Metrics	DBF	AWF[11]	DF[22]
OW-NoDef	*TPR*	98.74	74.26	98.52
	MTPR	98.67	71.23	97.96
	FPR	7.12	22.97	6.57
OW-W-T	*TPR*	93.92	1.67	66.17
	MTPR	64.11	1.06	53.21
	FPR	34.61	1.48	54.02
OW-WTFPAD	*TPR*	93.66	40.67	91.84
	MTPR	92.16	38.08	87.55
	FPR	18.13	9.03	15.71

3.6 模型复杂度分析

DBF 相比其他 2 个神经网络方法要更加轻便、运行速度更快,其神经网络结构简化对比如图 19 所示.DBF 的简化结构与 DF 相似(DBF 的具体参数在 2.3.3 节已有描述;DF 的 4 轮卷积网络参数为:卷积窗口均为 8,卷积步进均为 1,卷积核数依次为 32,64,128,256,池化步进均为 4,池化窗口均为 8),但 DBF 运算速度更快.一方面,DBF 仅有 3 轮基本卷积网络运算(即 2 层卷积层一层最大池化层),而 DF 有 4 轮.另一方面,DBF 的第 1 轮卷积网络用于 burst 特征提取,其结构远比 DF 的第 1 轮卷积网络要简单,如第 1 层卷积层的核数仅为 4(DF 的卷积核数为 32),第 2 层卷积层的卷积窗口大小仅为 1(DF 的卷积窗口大小为 8).DBF 由于深度分析 burst 特征的需要,密集连接网络运算有 4 轮,要多于 DF 的 2 轮,但密集连接网络的运算速度很快,时间消耗远远少于卷积网络.DBF 在简化网络结构的同时提高了模型性能,关键在于 DBF 充分结合了流量 burst 特征分析的需要和网站指纹攻击技术的特点设计神经网络结构,并且摒弃了以往研究中冗余的神经网络结构.其中最具特色的是 DBF 的第 1 轮卷积网络的第 1 层卷积层运算实际上包含了 3 个平行的卷积层,用于提取 burst 特征(DBF 的具体结构如 2.3.3 节图 6 所示),而这 3 个平行的卷积层是可以并行计算

的,因此没有增加时间消耗.AWF 神经网络结构虽然仅有 7 层,但长短时记忆网络层(LSTM)属于循环网络层的一种,运算非常耗时,因此 AWF 的时间消耗要大于 DBF 和 DF.

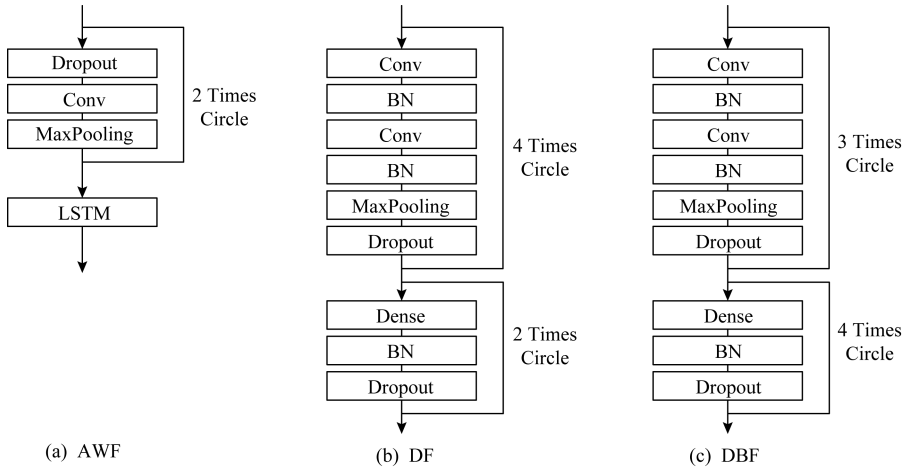

Fig. 19 Simplified neural network structures of the algorithms

图 19 各算法的神经网络结构简化图

DBF 与对比算法具体的时间消耗如表 13 和表 14 所示,DBF 每个 epoch 的训练时间只有 86.30 s,远低于对比算法,可知 DBF 在模型效率上同样优于对比算法.实际上 DBF 的训练并不需要多达 30 个 epoch,3.4.1 节验证了模型在 15～20 个 epoch 时就基本能达到最佳的性能效果.在减少训练 epoch 的情况下,模型的训练时间能进一步缩短.

Table 13 Running Time of DBF on OW-NoDef Dataset

表 13 DBF 在 OW-NoDef 数据集的运行时间 s

Sub-model	Training Time			Testing Time
	Per Epoch	Epochs	Total	
DBF-CW	86.30	30	2 589.11	14.36
DBF-OW			96.28	0.41
All(DBF)	86.30	30	2 685.39	14.77

Table 14 Comparison of Running Time of the Algorithms on OW-NoDef Dataset

表 14 各算法在 OW-NoDef 数据集的运行时间对比 s

Method	Training Time			Testing Time
	Per Epoch	Epochs	Total	
DBF	86.30	30	2 685.39	14.77
DF[22]	102.17	30	3 065.13	15.72
AWF[11]	133.95	30	4 018.71	107.75

3.7 实验讨论

从场景的设置上看,实验从封闭世界场景和开放世界场景 2 个角度对 DBF 进行了分析,模型均表现出了良好的性能.从功能性验证上看,DBF 在受被监控网站集大小影响、缓解真实网络环境存在的概念漂移问题、绕过 WF 攻击防御机制以及对 Tor 隐藏网站的检测上有较好的性能表现,这些模型性能对 WF 攻击技术应用到真实网络环境中有很大帮助;同时 DBF 在 3.5.2 节的开放世界场景实验验证中,表现出对传统阈值法的极大改进,相较传统方法明显降低了 FPR 值,但在 3.5.6 节的实验出现了 FPR 值过高的情况,表明抵御 WF 攻击的防御机制对带宽的扰乱,在误导 WF 模型将非监控网页流误分类为监控流方面起到了明显的作用.虽然 DBF 一定程度上突破了防御机制,并表现出了较高的 $MTPR$,但较高的 FPR 表示 DBF-OW 受防御机制加持的影响较大,说明模型在训练阶段对指纹向量的学习能力还有所欠缺.从模型自身的参数验证上看,DBF 对训练轮次 epoch、输入的特征序列长度、被监控网站的训练实例数、随机森林算法的子分类器数等参数敏感度不高,说明模型本身的结构是健壮的,模型性能不容易受参数变化而影响.从模型对比上看,DBF 模型在各方面的性能表现都要优于 DF 模型,但是在个别方面的优势不明显,如小样本训练下的模型准确率、输入序列长度对模型的准确率影响等;而 AWF 模型的性能与 DBF 和 DF 模型相差较大,证明了神经网络方法虽然是一个利器,但是如果没有对经典架构做出改进以适应 WF 的特点,神经网络的

优势也无法发挥出来.另外,DBF 相比其他 2 个神经网络方法要更加轻便、运行速度更快.综上,DBF 在保证模型运行效率的同时,全方位地提高了模型的分类性能.

4 结 论

本文提出了一个基于神经网络深度分析 burst 特征的网站指纹攻击模型 DBF,提高了神经网络应用到 WF 攻击技术上的适应性.DBF 有效缓解了概念漂移问题和提高了小样本训练下模型的分类准确率等,相比已有研究的方法要更加轻便、运行速度更快,从提升性能的角度提高了 WF 攻击技术应用到实际的可行性.但在 OW 场景下验证模型对 WF 攻击防御机制的突破能力实验中,DBF 出现了 FPR 过高的情况,这将对实际中的网络管理带来一定困难,也表明了 DBF 对 WF 攻击防御机制的突破还有很大的提升空间.该问题的出现与 DBF-OW 的设计是相关的,因此下一步将研究对 DBF-OW 作出改进,使 DBF-OW 的设计更加精细,以更加有效地应对 WF 攻击防御机制,有效降低在加持了防御机制下的 FPR 值,进一步提高 WF 攻击技术在 WF 攻击防御机制下的性能表现.

参 考 文 献

[1] Gu Xiaodan, Yang Ming, Luo Junzhou. A novel website fingerprinting attack against multi-tab browsing behavior [C]//Proc of the 19th IEEE Int Conf on Computer Supported Cooperative Work in Design. Piscataway, NJ: IEEE, 2015: 234-239

[2] Gu Xiaodan, Yang Ming, Luo Junzhou, et al. Website fingerprinting attack based on hyperlink relations [J]. Chinese Journal of Computers, 2015, 38(4): 833-845 (in Chinese)
(顾晓丹, 杨明, 罗军舟,等. 针对 SSH 匿名流量的网站指纹攻击方法[J]. 计算机学报, 2015, 38(4): 833-845)

[3] Herrmann D, Wendolsky R, Federrath H. Website Fingerprinting: Attacking popular privacy enhancing technologies with the multinomial Naïve-Bayes classifier [C] //Proc of the ACM Workshop on Cloud Computing Security. New York: ACM, 2009: 31-42

[4] Zeng Xudong, Kang Cuicui, Shi Junzheng, et al. A novel website fingerprinting method for malicious websites detection [C] //Proc of the Information and Communication Technology for Intelligent Systems. Berlin: Springer, 2019: 723-730

[5] Luo Junzhou, Yang Ming, Ling Zhen, et al. Anomynous communication and darknet: A survey [J]. Journal of Computer Research and Development, 2019, 56(1): 103-130 (in Chinese)
(罗军舟, 杨明, 凌振, 等. 匿名通信与暗网研究综述[J]. 计算机研究与发展, 2019, 56(1): 103-130)

[6] Dingledine R, Mathewson N, Syverson P. Tor: The second-generation onion router [C] //Proc of the 13th USENIX Security Symp. Berkley, CA: USENIX Association, 2004: 303-320

[7] Tor Developers. Tor metrics portal [OL]. [2019-09-12]. https://metrics.torproject.org.

[8] Wang Ran, Xu Guangquan, Liu Bin, et al. Flow watermarking for antinoise and multistream tracing in anonymous networks [J]. IEEE MultiMedia, 2017, 24(4): 38-47

[9] Nasr M, Bahramali A, Houmansadr A. Deepcorr: Strong flow correlation attacks on Tor using deep learning [C] // Proc of ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2018: 1962-1976

[10] Panchenko A, Niessen L, Zinnen A, et al. Website fingerprinting in onion routing based anonymization networks [C] //Proc of the 10th ACM Workshop on Privacy in the Electronic Society. New York: ACM, 2011: 103-114

[11] Rimmer V, Preuveneers D, Juarez M, et al. Automated website fingerprinting through deep learning [C/OL] //Proc of the 25th Network and Distributed System Security Symp. Rosten: The Internet Society, 2018 [2019-08-16]. https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_03A-1_Rimmer_paper.pdf

[12] Kwon A, AlSabah M, Lazar D, et al. Circuit fingerprinting attacks: Passive deanonymization of Tor hidden services [C] //Proc of the 24th USENIX Security Symp. Berkeley, CA: USENIX Association, 2015: 287-302

[13] Zhang Lei, Cui Yong, Liu Jing, et al. Application of machine learning in cyberspace security research [J]. Chinese Journal of Computers, 2018, 41(9): 1943-1975 (in Chinese)
(张蕾, 崔勇, 刘静, 等. 机器学习在网络空间安全研究中的应用[J]. 计算机学报, 2018, 41(9): 1943-1975)

[14] Wang Zhanyi. The applications of deep learning on traffic identification [EB/OL]. [2019-09-23]. https://www.blackhat.com/docs/us-15/materials/us-15-Wang-The-Applications-Of-Deep-Learning-On-Traffic-Identification-wp.pdf

[15] Oh S E, Sunkam S, Hopper N. p-FP: Extraction, classification, and prediction of website fingerprints with deep learning [EB/OL]. [2019-10-31]. https://arxiv.org/pdf/1711.03656.pdf

[16] Hayes J, Danezis G. k-fingerprinting: A robust scalable website fingerprinting technique [C] //Proc of the 25th USENIX Security Symp. Berkley, CA: USENIX Association, 2016: 1187-1203

[17] Panchenko A, Lanze F, Pennekamp J, et al. Website fingerprinting at Internet scale [C] //Proc of the 23rd Network and Distributed System Security Symp. Rosten: The Internet Society, 2016: 1-15

[18] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [EB/OL]. [2019-10-18]. https://arxiv.org/pdf/1409.1556.pdf

[19] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition [C] //Proc of the 26th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 770-778

[20] He Xiaomin, Wang Jing, He Yueying, et al. A deep learning approach for website fingerprinting attack [C] //Proc of the 4th IEEE Int Conf on Computer and Communications. Piscataway, NJ: IEEE, 2018: 1419-1423

[21] Bhat S, Lu D, Kwon A, et al. Var-CNN: A data-efficient website fingerprinting attack based on deep learning [J]. Proceedings on Privacy Enhancing Technologies, 2019, 2019 (4): 292-310

[22] Sirinam P, Imani M, Juarez M, et al. Deep fingerprinting: Undermining website fingerprinting defenses with deep learning [C] //Proc of ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2018: 1928-1943

[23] Yang Ming, Luo Junzhou, Ling Zhen, et al. De-anonymizing and countermeasures in anonymous communication networks [J]. IEEE Communications Magazine, 2015, 53(4): 60-66

[24] Yao Zhongjiang, Zhang Lei, Ge Jingguo, et al. An invisible flow watermarking for traffic tracking: A hidden Markov model approach [C] //Proc of the 53rd IEEE Int Conf on Communications. Piscataway, NJ: IEEE, 2019: 1-6

[25] Rezaei F, Houmansadr A. Tagit: Tagging network flows using blind fingerprints [J]. Proceedings on Privacy Enhancing Technologies, 2017, 2017(4): 290-307

[26] Liberatore M, Levine B N. Inferring the source of encrypted HTTP connections [C]//Proc of the 13th ACM Conf on Computer and Communications Security. New York: ACM, 2006: 255-263

[27] Yang Ming, Gu Xiaodan, Ling Zhen, et al. An active de-anonymizing attack against Tor Web traffic [J]. Tsinghua Science and Technology, 2017, 22(6): 702-713

[28] Winter P, Ensafi R, Loesing K, et al. Identifying and characterizing sybils in the Tor network [C] //Proc of the 25th USENIX Security Symp. Berkeley, CA: USENIX Association, 2016: 1169-1185

[29] Shusterman A, Kang L, Haskal Y, et al. Robust website fingerprinting through the cache occupancy channel [C] // Proc of the 28th USENIX Security Symp. Berkeley, CA: USENIX Association, 2019: 639-656

[30] Juarez M, Afroz S, Acar G, et al. A critical evaluation of website fingerprinting attacks [C] //Proc of the 21st ACM Conf on Computer and Communications Security. New York: ACM, 2014: 263-274

[31] Lazarenko A, Avdoshin S. Anonymity of Tor: Myth and reality [C] //Proc of the 12th Central and Eastern European Software Engineering Conf in Russia. New York: ACM, 2016: 10-14

[32] Wang Tao, Goldberg I. On realistically attacking Tor with website fingerprinting [J]. Proceedings on Privacy Enhancing Technologies, 2016, 2016 (4): 21-36

[33] Cui Weiqi, Chen Tao, Fields C, et al. Revisiting assumptions for website fingerprinting attacks [C] //Proc of the 14th ACM Asia Conf on Computer and Communications Security. New York: ACM, 2019: 328-339

[34] Cai Xiang, Zhang Xincheng, Joshi B, et al. Touching from a distance: Website fingerprinting attacks and defenses [C] // Proc of the 19th ACM Conf on Computer and Communications Security. New York: ACM, 2012: 605-616

[35] Zhuo Zhongliu, Zhang Yang, Zhang Zhili, et al. Website fingerprinting attack on anonymity networks based on profile hidden Markov model [J]. IEEE Transactions on Information Forensics and Security, 2017, 13(5): 1081-1095

[36] Wang Tao, Cai Xiang, Nithyanand R, et al. Effective attacks and provable defenses for website fingerprinting [C] //Proc of the 23rd USENIX Security Symp. Berkeley, CA: USENIX Association, 2014: 143-157

[37] Hintz A. Fingerprinting websites using traffic analysis [C]// Proc of the Int Workshop on Privacy Enhancing Technologies. Berlin: Springer, 2003: 171-178

[38] Sun Qixiang, Simon D R, Wang Yimin, et al. Statistical identification of encrypted Web browsing traffic [C] //Proc of the 23rd IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2002: 19-30

[39] Lu Liming, Chang E C, Chan M C. Website fingerprinting and identification using ordered feature sequences [C] //Proc of European Symp on Research in Computer Security. Berlin: Springer, 2010:199-214

[40] Wang Tao, Goldberg I. Improved website fingerprinting on Tor [C] //Proc of the 12th ACM Workshop on Privacy in the Electronic Society. New York: ACM, 2013: 201-212

[41] Jahani H, Jalili S. A novel passive website fingerprinting attack on Tor using fast fourier transform [J]. Computer Communications, 2016, 96: 43-51

[42] Abe K, Goto S. Fingerprinting attack on Tor anonymity using deep learning [J]. Proceedings of the Asia-Pacific Advanced Network, 2016, 42: 15-20

[43] Rahman M S, Sirinam P, Matthews M, et al. Tik-Tok: The utility of packet timing in website fingerprinting attacks [EB/OL]. [2019-11-01]. https://arxiv.org/pdf/1902.06421.pdf

[44] Juarez M, Imani M, Perry M, et al. Toward an efficient website fingerprinting defense [C] //Proc of the European Symp on Research in Computer Security. Berlin: Springer, 2016: 27-46

[45] Cherubin G, Hayes J, Juarez M. Website fingerprinting defenses at the application layer [J]. Proceedings on Privacy Enhancing Technologies, 2017, 2017 (2): 186-203

[46] De la Cadena W, Mitseva A, Pennekamp J, et al. POSTER: Traffic splitting to counter website fingerprinting [C] //Proc of the 26th ACM SIGSAC Conf on Computer and Communication Society. New York: ACM, 2019: 2533-2535

[47] Al-Naami K, Gharmy A E I, Islam M S, et al. BiMorphing: A bi-directional bursting defense against website fingerprinting attacks [EB/OL]. [2019-11-03]. http://dx.doi.org/10.1109/TDSC.2019.2907240

[48] Liu Xiaolei, Zhuo Zhongliu, Du Xiaojiang, et al. Adversarial attacks against profile HMM website fingerprinting detection model [J]. Cognitive Systems Research, 2019, 54: 83-89

[49] Cai Xiang, Zhang Xincheng, Joshi B, et al. Touching from a distance: Website fingerprinting attacks and defenses [C] // Proc of the 19th Conf on Computer and Communications Security. New York: ACM, 2012: 605-616

[50] Cai Xiang, Nithyanand R, Johnson R. CS-BuFLO: A congestion sensitive website fingerprinting defense [C] //Proc of the 13th Workshop on Privacy in the Electronic Society. New York: ACM, 2014: 121-130

[51] Cai Xiang, Nithyanand R, Wang Tao, et al. A systematic approach to developing and evaluating website fingerprinting defenses [C] // Proc of the 21st ACM Conf on Computer and Communications Security. New York: ACM, 2014: 227-238

[52] Imani M, Rahman M S, Wright M. Adversarial traces for website fingerprinting defense [C] //Proc of 2018 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2018: 2225-2227

[53] Imani M, Rahman M S, Matthews M, et al. Mockingbird: Defending against deep-learning-based website fingerprinting attacks with adversarial traces [EB/OL]. [2019-10-27]. https://arxiv.org/pdf/1902.06626.pdf

[54] Juárez M, Imani M, Perry M, et al. Toward an efficient website fingerprinting defense [C] //Proc of the European Symp on Research in Computer Security. Berlin: Springer, 2016: 27-46

[55] Wang Tao, Goldberg I. Walkie-talkie: An efficient defense against passive website fingerprinting attacks [C] //Proc of the 26th USENIX Security Symp. Berkley, CA: USENIX Association, 2017: 1375-1390

Ma Chencheng, born in 1994. Master candidate. His main research interests include network security and machine learning.

Du Xuehui, born in 1968. PhD, professor, PhD supervisor. Her main research interests include cloud computing and big data security.

Cao Lifeng, born in 1981. PhD, associate professor. His main research interests include cloud computing and information security.

Wu Bei, born in 1980. Engineer. Her main research interest is network security.