基于关联信息提取的恶意域名检测方法

张斌[1,2]，廖仁杰[1,2]

（1. 信息工程大学密码工程学院，河南 郑州 450001；2. 河南省信息安全重点实验室，河南 郑州 450001）

摘 要：为提高基于域名关联信息的恶意域名检测准确率，提出了一种基于域名解析信息与请求时间相结合的恶意域名检测方法。首先，将域名解析记录表示为异质信息网络中的节点和边，以同时表征异质域名数据获得较高的域名信息利用率；其次，为避免采用稀疏邻接矩阵相乘操作提取关联信息时间复杂度较高的问题，提出了一种基于元路径的广度优先网络遍历算法，提高关联解析信息提取效率；针对弱连接域名由于缺少关联解析信息而漏检的问题，引入请求时间刻画域名之间相关性，提高检测样本覆盖率；最后，设计权重自适应的域名表示学习算法，将域名关联解析信息和关联请求时间信息向量化，通过域名特征向量之间的欧氏距离量化域名之间关联性，进而构建有监督分类器进行恶意域名检测。理论分析和实验结果表明，所提方法具有较高的域名关联信息提取效率，所得检测覆盖率和 F1 分数分别为 97.7%和 0.951。

关键词：恶意域名检测；异质信息网络；域名解析信息；请求时间；表示学习

中图分类号：TP393

文献标识码：A

DOI: 10.11959/j.issn.1000−436x.2021181

Malicious domain name detection method based on associated information extraction

ZHANG Bin[1,2], LIAO Renjie[1,2]

1. Department of Cryptogram Engineering, Information Engineering University, Zhengzhou 450001, China
2. He’nan Province Key Laboratory of Information Security, Zhengzhou 450001, China

Abstract: To improve the accuracy of malicious domain name detection based on the associated information, a detection method combining resolution information and query time was proposed. Firstly, the resolution information was mapped to nodes and edges in a heterogeneous information network, which improved the utilization rate. Secondly, considering the problem of high computational complexity in extracting associated information with matrix multiplication, an efficiency breadth-first network traversal algorithm based on meta-path was proposed. Then, the query time was used to detect the domain names lacking meta-path information, which improved the coverage rate. Finally, domain names were vectorized by representation learning with adaptive weight. The Euclidean distance between domain name feature vectors was used to quantify the correlation between domain names. Based on the vectors learned above, a supervised classifier was constructed to detect malicious domain names. Theoretical analysis and experimental results show that the proposed method preforms well in extraction domain name associated information. The coverage rate and F1 score are 97.7% and 0.951 respectively.

Keywords: malicious domain name detection, heterogeneous information network, domain name resolution information, query time, representation learning

收稿日期：2021−04−12；**修回日期**：2021−07−01

通信作者：廖仁杰，lrj2803@163.com

基金项目：信息保障技术重点实验室开放基金资助项目（No.KJ-15-109）；信息工程大学新兴科研方向培育基金资助项目（No.2016604703）；信息工程大学科研基金资助项目（No.2019f3303）

Foundation Items: The Open Fund Project of Information Assurance Technology Key Laboratory (No. KJ-15-109), The New Research Direction Cultivation Fund of Information Engineering University (No.2016604703), The Research Project of Information Engineering University (No.2019f3303)

1　引言

恶意域名通常采用 IP-Flux、Domain-Flux 等技术动态变换域名字符串构成和 IP 的映射关系，具有较强的欺骗性和隐蔽性[1-2]，如何有效检测恶意域名已成为网络安全领域的研究热点之一。

目前，基于域关联信息的恶意域名检测方法按提取关联信息的不同分为三类。第一类是基于域名请求时间关联的恶意域名检测方法，如基于固定时间窗分析域名请求时间相似性的域名检测方法[3-4]、基于同类别域名请求呈伴随关系的域名检测方法[5]等。此类检测方法的出发点是同类别域名在请求时间上呈聚集出现的特点，对并发访问的恶意域名检测效果较好，并能检测大部分域名，检测样本覆盖率较高，但易受主机产生的合法域名请求和观测时间窗口大小设置的干扰，还需结合域名的其他信息以提高此类方法的稳健性。第二类是基于置信度传播（BP, belief propagation）算法[6]的恶意域名检测方法。此类检测方法基于图模型挖掘域名之间的关联关系，首先，提取域名系统（DNS, domain name system）流量中域名解析 IP 地址、访问域名主机等信息构成图模型，如域名−主机二部图[7]、域名−IP 地址二部图[8-9]、域名传播图[10]、别名图[11]等；然后结合已有黑白名单标记图中部分节点，采用 BP 算法或图聚类方法对图中域名节点进行标记。此类方法基于图中存在边连接的域名节点具有同质性的特点进行节点标记，可在已知标签数据较少的情况下对未知属性域名节点进行检测，但由于仅利用 DNS 流量中单一类型域名信息构成图模型，导致域名信息利用率较低，检测效果不佳。第三类是基于异质信息网络（HIN, heterogeneous information network）[12]的恶意域名检测方法。此类检测方法依据与恶意域名、攻击者掌控 IP 地址存在联系的域名大概率为恶意域名的假设，首先将 DNS 流量中多种信息，如域名、IP 地址、访问主机等映射为 HIN 中的节点，然后采用网络表示学习方法将 HIN 中域名节点间的关联信息向量化，使具有关联的域名向量在特征空间中聚类出现，所得域名向量可结合分类算法实现域名检测，如结合域名请求主机、域名解析 IP 地址和域名请求时间信息，采用 LINE（large-scale information network embedding）[13]进行域名表示学习的域名检测方法[14-15]，结合 HIN 与直推式分类器的域名检测方法[16]，采用图卷积网络[17]进行域名节点表示学习的检测方法[18-19]以及结合 IP 地址信息、被动 DNS 特征和域名字符串特征的检测方法[20]。此类方法采用 HIN 表示域名相关信息，提高了 DNS 流量中域名信息利用率，并通过表示学习方法将 HIN 中域名节点向量化，为恶意域名检测提供区分性强的训练数据，检测准确率较第二类方法有较大提升，但此类方法存在以下不足：1) 由域名信息构造而成的 HIN 中存在弱连接域名，此类节点与其他节点不存在边连接，导致无法从 HIN 中挖掘关联解析信息实现检测，检测样本覆盖率较低；2) 采用矩阵乘法操作提取 HIN 中域名节点之间关联解析信息，时间复杂度较高。

为提高基于域名关联信息检测恶意域名的样本覆盖率和检测准确率，本文考虑结合第一类方法具有较高检测样本覆盖率和第三类检测方法中采用 HIN 表示域名解析信息具有较高检测准确率的特点，提出一种结合域名解析 IP 地址、别名记录和请求时间进行关联信息挖掘的恶意域名检测方法。

本文主要的研究工作如下。

1) 将 DNS 流量中域名解析信息映射为 HIN 中的节点和边，弥补由于采用同质网络无法同时表示域名与 IP 地址之间的解析关系和域名之间别名关系的不足，提高域名信息利用率；给出描述域名之间关联信息的元路径定义，同时提出一种用于提取域名关联信息的网络遍历方法，避免采用矩阵乘法操作提取元路径关联信息计算复杂度较高的问题。

2) 提出基于请求时间的弱连接域名关联信息挖掘方法，根据较小时间窗内发起请求的域名之间属性相似的特点，从请求时间角度挖掘弱连接域名的关联信息，解决弱连接域名因元路径关联信息缺失而无法被检测的问题，提高检测样本覆盖率。

3) 提出一种域名表示学习方法，通过基于元路径的域名关联解析信息与基于请求时间的域名关联信息进行差异学习，将域名映射为特征空间的数值向量，通过向量间欧氏距离反映域名之间关联程度，为用于恶意域名检测的有监督分类器提供区分性较强的训练数据，获得较高的检测准确率。

2　方法设计

基于域名关联信息进行恶意域名检测的依据如下。1) 恶意域名解析信息存在关联关系：由于攻击者掌握的 IP 资源有限，不同恶意程序所使用域名的解析 IP 地址存在交集，可通过分析域名 IP 共享

机制发现恶意域名家族[10,12]。2) 恶意域名在请求时间上存在关联关系：由于感染恶意程序的主机以固定时间周期发送域名请求以验证控制服务器状态和获取攻击命令，安全人员可在 DNS 记录中发现感染主机对恶意域名周期性访问的现象[2,4]。由于攻击者可通过劫持合法域名进行攻击活动，并在对恶意域名发起查询的同时，随机发送大量合法域名请求以隐藏恶意域名请求，若仅依靠单一类型关联信息进行域名检测易产生较多误报与漏报[1]。综上所述，本文结合域名解析信息与请求时间信息进行恶意域名检测，以提高基于域名关联信息的恶意域名检测结果可靠性。

基于解析信息与请求时间相结合的恶意域名检测方法（MDND-RIQT, malicious domain name detection based on resolution information and query time），同时利用域名解析信息与请求时间关联信息进行域名检测：采用异质信息网络挖掘域名解析信息中存在的关联信息；根据固定时间窗提取域名请求时间关联信息；设计域名表示学习算法，将未知域名与已知合法/恶意域名的关联程度量化为向量间欧氏距离，所得数值向量作为域名特征，并结合有监督分类器实现域名检测。MDND-RIQT 整体流程如图 1 所示，包括域名异质信息网络（DN-HIN, domain name heterogeneous information network）构建、基于关联信息的域名对提取、域名表示学习和域名分类器的训练与测试。

DN-HIN 构建是将 DNS 流量中解析记录表征为异质信息网络，为挖掘域名关联解析信息提供数据表示。基于关联信息的域名对提取围绕域名基于元路径的关联解析信息和基于请求时间关联信息展开，将存在关联信息的 2 个域名记为一个域名对。域名表示学习自动融合不同类别的域名关联信息，将域名映射为数值向量，通过向量间欧氏距离量化域名之间关联程度，最后通过已知标签的域名向量训练有监督分类器用于未知标签域名检测。

2.1 DN-HIN 构建

对于网络 $G=(V,E)$，其中 V 和 E 分别代表网络 G 中的节点和边，G 中存在节点类型的映射关系 $\phi: V \to A$，使 $\forall v \in V$，$\phi(v) \in A$，以及边连接的映射关系 $\varphi: E \to R$，使 $\forall e \in E$，$\varphi(e) \in R$，A 和 R 分别代表节点类型集合和边连接类型集合。若 $|A|+|R|>2$，则称 G 为异质信息网络。异质信息网络已广泛应用于信息检索和数据挖掘领域[12]。

在网络通信中，主机发起对某一域名的查询请求后，可通过本地缓存或解析服务器递归查询获得查询结果。DNS 流量中 A 和 AAAA 类型记录包含域名与 IP 地址一对一或一对多的解析关系、CNAME 类型记录包含域名的别名关系。为充分挖掘域名与 IP 地址、域名与域名之间的关联信息用于恶意域名检测，选取 HIN 表示不同类型的域名解析信息，构成 DN-HIN。DN-HIN 包含 2 种节点（即域名节点 N_{D}、IP 地址节点 N_{IP}）和 2 种边连接关系（即 N_{D} 与 N_{IP} 之间的解析关系 R_{Resolve}、别名记录构成的 CNAME 关系 R_{CNAME}）。采用 2 个邻接矩阵存储 DN-HIN 中节点之间 R_{Resolve} 和 R_{CNAME} 边连接关系，分别记为 $\boldsymbol{M}_{\mathrm{Resolve}}$ 和 $\boldsymbol{M}_{\mathrm{CNAME}}$，并根据 DNS 流量中的域名信息对矩阵进行赋值，矩阵赋值如下

$$\boldsymbol{M}_{\mathrm{Resolve}}[i,j]=\begin{cases}1, & (N_{\mathrm{D}}(i),N_{\mathrm{IP}}(j)) \in R_{\mathrm{Resolve}} \\ 0, & \text{其他}\end{cases}$$

$$\boldsymbol{M}_{\mathrm{CNAME}}[i,j]=\begin{cases}1, & (N_{\mathrm{D}}(i),N_{\mathrm{D}}(j)) \in R_{\mathrm{CNAME}} \\ 0, & \text{其他}\end{cases} \tag{1}$$

其中，$\boldsymbol{M}_{\mathrm{Resolve}} \in \mathbb{R}^{|N_{\mathrm{D}}|\times|N_{\mathrm{IP}}|}$，$\boldsymbol{M}_{\mathrm{CNAME}} \in \mathbb{R}^{|N_{\mathrm{D}}|\times|N_{\mathrm{D}}|}$，$|N_{\mathrm{D}}|$ 和 $|N_{\mathrm{IP}}|$ 分别为域名数量与解析 IP 地址数量，i 与 j 为矩阵中索引值。

2.2 基于元路径的域名关联解析信息提取

异质信息网络中 2 个节点可通过不同路径建立

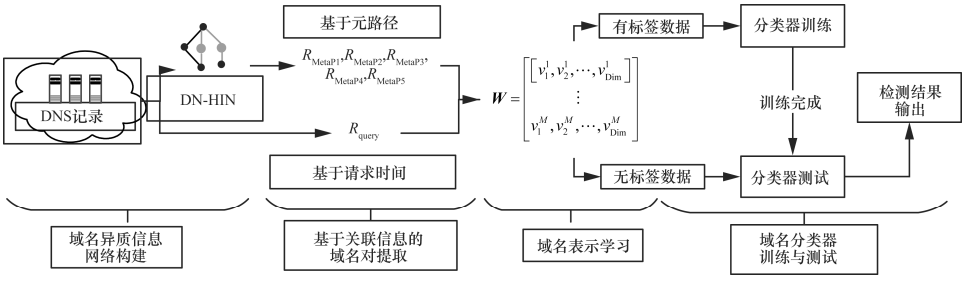


图 1 MDND-RIQT 整体流程

连接，不同的路径代表节点间不同的语义关系，这样的路径称为元路径[12]。为挖掘 DN-HIN 中域名节点之间的连接关系用于恶意域名检测，定义以下 5 种类型的域名元路径（MetaP, meta-path），其中，d 代表域名，IP 代表域名解析 IP 地址。

1) MetaP1：域名(d_1)-IP 地址(IP_1)-域名(d_2)。有限的 IP 地址资源导致不同域名的解析 IP 地址存在交集，并且解析到同一 IP 地址的域名之间具有较强的同质性。例如，同一公司的多个域名通常解析为同一个 IP 地址；恶意程序采用域名生成算法产生的大量恶意域名往往指向同一个 IP 地址，以确保感染主机被攻击者同时管控。

2) MetaP2：域名(d_1)-CNAME-域名(d_2)。CNAME 表示域名的别名记录。若域名 d_1 的查询结果为 CNAME 记录，将继续对别名域名 d_2 发起查询，最终获得域名 d_1 的解析 IP 地址。网络攻击者通过设置 CNAME 记录将感染主机发起的对恶意域名的查询转移到跳板主机，并可灵活更换跳板主机以提高网络攻击事件中的通信隐蔽性[10]。

3) MetaP3：域名(d_1)-IP 地址(IP_1)-域名(d_2)-IP 地址(IP_2)-域名(d_3)。考虑到域名解析的负载均衡问题，在实际设置中通常为同一个域名配置多个解析 IP 地址，并且每个 IP 地址也可作为多个域名的解析地址，从而出现 MetaP3 表示的域名解析 IP 地址共享机制。文献[21]中指出，僵尸网络控制者为寻求更高的经济利益，开始为其他恶意程序提供服务，出现同一恶意域名在不同时间被不同恶意程序家族使用的 Baas（botnet as a service）模式，并且为躲避监管，所使用恶意域名的解析 IP 地址会在不同国家和托管平台之间迁移。

4) MetaP4：域名(d_1)-CNAME-域名(d_2)-CNAME-域名(d_3)。域名 d_1 的查询结果为别名为域名 d_2 的 CNAME 记录，继续发起对域名 d_2 的查询，返回一条别名记录为域名 d_3 的 CNAME 记录，最终由域名 d_3 的查询结果得到解析 IP 地址并将此 IP 地址作为域名 d_1 的解析结果。此类域名利用方式常用于采用动态域名解析服务的钓鱼网站和网络诈骗[10]，具有较高隐蔽性。

5) MetaP5：域名(d_1)-IP 地址(IP_1)-域名(d_2)-CNAME-域名(d_3)。MetaP5 在 MetaP1 和 MetaP2 的基础进行拓展。对于同时存在 A 类型和 CNAME 类型查询结果的域名 d_2，可将域名 d_2 作为中间节点，使域名 d_1 与域名 d_3 建立长距离关联关系。

以上 5 种元路径以合法/恶意网络活动中域名、IP 地址之间的联系为基础，通过不同长度、不同连接关系的元路径提高域名关联解析信息挖掘的全面性，并用于推理域名节点的属性：若域名节点在 DN-HIN 中与已知恶意域名节点或攻击者掌控的 IP 地址节点存在元路径联系，则该域名倾向为恶意。

通过统计 DN-HIN 中域名节点的出度可知，域名节点的出度为 1～6，从而由式(1)所得的邻接矩阵为稀疏矩阵。已有研究采用邻接矩阵相乘操作挖掘域名节点之间不同的元路径关联信息，此过程受元路径种类数、元路径长度、邻接矩阵大小等因素影响，具有较大的计算开销[16]。为提高在 DN-HIN 中提取域名节点之间元路径关联信息的效率，设计基于元路径的网络遍历算法（NTA-M, network traversal algorithm based on meta-path），该算法以 DN-HIN 中域名节点作为遍历起点，以广度优先原则搜寻 DN-HIN 中满足 5 种元路径的下一跳节点，最终输出与元路径匹配的域名节点序列，具体描述如算法 1 所示。

算法 1　基于元路径的网络遍历算法

输入　邻接矩阵 $\boldsymbol{M}_{\text{Resolve}}$ 和 $\boldsymbol{M}_{\text{CNAME}}$，域名集合 DN_Set，元路径遍历匹配项 MetaP3、MetaP4 和 MetaP5

输出　满足元路径关系的域名节点序列集合 Traversal_Result

1) Traversal_Result $\leftarrow \varnothing$ //存放遍历结果

2)for dn in DN_Set do //依次选择域名节点 dn 作为起点

3) Result $\leftarrow$ Traversal(dn) //调用遍历算法，Result 用于存放 dn 的遍历结果

4) Traversal_Result.append(Result) //append 为添加操作

5)end for

Function Traversal(List_NodeSequence) //List_NodeSequence 为节点序列集合

1)for ns in List_NodeSequence do //依次选择节点序列

2) Temp_Record,Record $\leftarrow \varnothing$ //分别存放还需继续遍历的节点序列和已匹配的节点序列

3) NbrIP = Neighbor(ns $[-1]$, $\boldsymbol{M}_{\text{Resolve}}$)

4) NbrCNAME = Neighbor(ns [−1], $\boldsymbol{M}_{\text{CNAME}}$) //寻找节点序列 ns 的末尾节点在 $\boldsymbol{M}_{\text{Resolve}}$ 和 $\boldsymbol{M}_{\text{CNAME}}$ 中的邻居节点

5) for NbrItem in [NbrIP, NbrCNAME] do

6) NewL = ns.append(NbrItem) //将邻居节点加入节点序列

7) if Match(NewL, (MetaP3, MetaP4, MetaP5)) //判断更新后的节点序列是否与 MetaP3、MetaP4 和 MetaP5 匹配

8) Record.append(NewNodeList) //若匹配，加入节点序列结果集合

9) else

10) TempRecord.append(NewNodeList) //否则，加入还需继续遍历的节点序列集合

11) end if

12) end for

13) if TempRecord is not ∅ //若遍历节点序列集合不为空

14) result = Traversal(TempRecord) //调用遍历算法

15) Record.append(result) //加入已匹配的节点序列集合

16) return Record

17) else

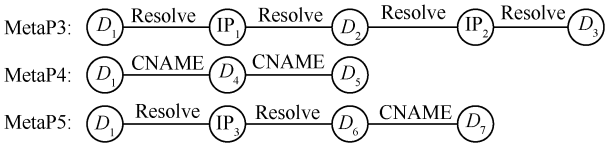
18) return Record //返回结果

19) end if

20) end for

设邻接矩阵 $\boldsymbol{M}_{\text{Resolve}}$ 大小为 $n\times m$、$\boldsymbol{M}_{\text{CNAME}}$ 大小为 $n\times n$，其中，n 为域名数量，m 为 IP 地址数量。若采用矩阵相乘操作提取基于 MetaP3（最长元路径）的域名节点序列，算法复杂度为 $O(n^3m)$。设 DN-HIN 中节点最大出度为 l，则 NTA-M 在最坏情况下（DN-HIN 中所有节点出度均为 l，元路径均为 MetaP3）的算法复杂度为 $O(nl^4)$。由于 l 的数量级远小于 m 或 n 的数量级，则 NTA-M 具有较小的算法复杂度。此外，采用矩阵相乘操作来提取域名关联信息需保存矩阵相乘的结果，该矩阵为 $n\times n$ 的稀疏矩阵，具有较大存储空间开销。NTA-M 所得结果仅需保存与元路径匹配的节点序列。综上，NTA-M 比基于矩阵乘法的元路径信息提取方法具有更小的时间与空间开销。

由于 MetaP3、MetaP4、MetaP5 包含 MetaP1 与 MetaP2，NTA-M 仅考虑 MetaP3、MetaP4 与 MetaP5 用于域名元路径信息提取，所得域名节点序列如图 2(a)所示（均以域名 D_1 作为遍历起点），D 代表域名节点、IP 代表 IP 地址节点、Resolve 和 CNAME 分别对应域名解析关系和域名别名关系。为进一步提取域名之间元路径关联信息用于域名检测，将域名节点序列中的边连接信息和 IP 地址节点删除，并划分为 5 种不同类型的域名对集合，如图 2(b)所示，域名对将用于后续域名表示学习。



(a) 域名节点序列

R_{MetaP1}:$(D_1,D_2),(D_1,D_6),(D_2,D_3)$
R_{MetaP2}:$(D_1,D_4),(D_4,D_5),(D_6,D_7)$
R_{MetaP3}:(D_1,D_3)
R_{MetaP4}:(D_1,D_5)
R_{MetaP4}:(D_1,D_7)

(b) 与不同元路径匹配的域名对

图 2 基于 NTA-M 提取域名对示意

受 DNS 流量采集时长、网络环境等因素影响，弱连接域名在 DN-HIN 中无法提取到任何元路径关联信息用于检测。为此，引入域名请求时间关联信息，用于提取与弱连接域名存在请求时间关联的域名对。

2.3 基于请求时间的弱连接域名关联信息挖掘

当用户浏览合法或恶意网站时，浏览器在较短的时间内会向不同域名发起的 DNS 查询请求，以获取网页中的文字、图片等内容，在此过程中被发起请求的部分域名之间虽不存在解析信息关联，但出现在同一网页浏览事件中，所发起请求的域名具有较大概率为同一类别（合法或恶意）。此外，恶意程序中通过域名生成算法产生的恶意域名在请求时间上呈集中请求的特点。由此，在较小时间窗内发起请求的域名具有较大概率属于同一类别，基于域名请求时间的关联信息可用于恶意域名检测[2-4]。为弥补 DN-HIN 中弱连接域名由于缺少元路径关联信息无法被检测的不足，本文提出基于域名请求时间的关联信息提取方法（AIEM-DNQT, associated information extraction method based on domain name query time），该方法通过提取域名请求时间关联信息有效检测弱连接域名。

设 Dataset 为 N 台主机的域名请求记录集合，$\text{Dateset}=\{D_1,D_2,\cdots,D_N\}$，其中 $D_i=\{(d_1^i,t_1^i),$

$(d_2^i,t_2^i),\cdots,(d_{L_i}^i,t_{L_i}^i)\}$，$i\in\{1,2,\cdots,N\}$，$(d_j^i,t_j^i)$ 代表主机 i 在 t_j^i 时刻发起对域名 d_j^i 的查询，L_i 代表 D_i 的域名请求总数；$\mathrm{WDN\_List}=\{\mathrm{WDN}_1,\mathrm{WDN}_2,\cdots,\mathrm{WDN}_{N'}\}$ 为弱连接域名构成的集合，N' 为弱连接域名数量。令 τ 为请求时间判别阈值，若域名与弱连接域名的请求时间间隔小于 τ，则判别 2 个域名之间存在请求时间关联关系。AIEM-DNQT 具体描述如算法 2 所示。

算法 2　基于域名请求时间的关联信息提取算法

输入　主机域名请求记录集合 Dataset，弱连接域名集合 WDN_List，判别阈值 τ

输出　弱连接域名基于请求时间关联的域名对集合 R_{query}

1) $R_{\mathrm{query}}\leftarrow\varnothing$　//存放结果

2)for WDN in WDN_List do //依次选择弱连接域名进行关联域名对提取

3)　$\mathrm{Temp\_}R_{\mathrm{query}}\leftarrow\varnothing$　//用于存放中间结果

4)　for D in Dataset do　//依次遍历所有域名请求记录

5)　List_Ad = Adjacent(WDN, 2τ, D)　//寻找请求记录 D 中以 WDN 为中心、大小为 2τ 的时间窗内出现的其他域名，并与 WDN 构成域名对保存到 List_Ad 中

6)　$\mathrm{Temp\_}R_{\mathrm{query}}$.append(List_Ad)　//append 为添加操作

7)　end for

8)　R_{query}.append(TopK($\mathrm{Temp\_}R_{\mathrm{query}}$))　//按出现频次降序排列域名对，选择前 K 个域名对作为最终结果

9)end for

算法 2 分为 2 个阶段。1) 弱连接域名请求记录遍历，设 Dataset 中每个弱连接域名请求记录数为 Q，以弱连接域名为中心，大小为 2τ的时间窗内平均域名请求记录数为 N_{av}，则得到所有弱连接域名的域名对时间复杂度为 $O(QWN_{\mathrm{av}})$；2) 域名对排序，设每个弱连接域名的平均域名对数量为 P，则算法 2 的时间复杂度为 $O(QWN_{\mathrm{av}}\mathrm{lb}P)$。算法 2 中步骤 8) 选择出现频次较高的域名对作为最终结果，通过频次统计降低主观设置 τ 值和主机后台程序发起的合法域名请求带来的干扰。R_{query} 将用于后续域名表示学习。

2.4　基于域名关联信息的域名表示学习方法

借鉴 Skip-Gram 模型可在保持字、词语义关系的前提下，基于文本向量化的思想[22]，将通过 NTA-M 和 AIEM-DNQT 分别得到基于元路径和基于请求时间关联信息的域名对理解为自然语言处理的词组，输入 Skip-Gram 模型，将每个域名转化为维度固定的数值向量。域名向量间欧氏距离反映域名之间的关联程度，域名向量之间距离越小说明域名之间关联越紧密。

Skip-Gram 模型训练需建立 2 个大小均为 $M\times\mathrm{Dim}$ 的矩阵，分别记为域名向量矩阵 $\boldsymbol{W}$ 和关联域名向量矩阵 $\boldsymbol{W}'$，其中，M 为域名样本总数，Dim 为域名向量维度，$\mathrm{Dim}\ll M$。通过 Skip-Gram 模型学习域名向量的目标是对于任意存在关联关系的域名对 (d_i,d_j)（$i,j\in\{1,2,\cdots,M\}$，$i\neq j$），使条件概率 $P(d_j\mid d_i,\theta)$ 最大化，$P(d_j\mid d_i,\theta)$ 采用 Softmax 函数进行衡量。

$$P\left(d_j\mid d_i,\theta\right)=\frac{e^{\boldsymbol{v}_j'^{\mathrm{T}}\boldsymbol{v}_i}}{\sum_{k=1}^{M}e^{\boldsymbol{v}_k'^{\mathrm{T}}\boldsymbol{v}_i}}\tag{2}$$

其中，θ 为 $\boldsymbol{W}$ 与 $\boldsymbol{W}'$ 所包含的参数，$\boldsymbol{v}_i$ 为域名 d_i 的数值向量（矩阵 $\boldsymbol{W}$ 中第 i 行对应的数值向量），$\boldsymbol{v}_j'$ 和 $\boldsymbol{v}_k'$ 为域名 d_j 和 d_k 在矩阵 $\boldsymbol{W}'$ 中的数值向量。

推论 1　令 $R=\{R_{\mathrm{MetaP1}},R_{\mathrm{MetaP2}},R_{\mathrm{MetaP3}},R_{\mathrm{MetaP4}},R_{\mathrm{MetaP5}},R_{\mathrm{query}}\}$ 为包含 6 种关联关系域名对的集合，若采用 R 中域名对作为 Skip-Gram 模型的训练数据，那么域名表示学习的目标函数可表示为

$$L_{\mathrm{Weight}}=\arg\min\left(-\sum_{R_r\in R}w_r\sum_{(d_i,d_j)\in R_r}\left(\left(\log\sigma\left(\boldsymbol{v}_j'^{\mathrm{T}}\boldsymbol{v}_i\right)+\sum_{k=1}^{N_k}\mathrm{E}_{v_k\sim P_n(v)}\left[\log\sigma\left(-\boldsymbol{v}_k'^{\mathrm{T}}\boldsymbol{v}_i\right)\right]\right)\right)\right)\tag{3}$$

其中，w_r 为关联关系 R_r 的权值，$\sigma(x)=1/\left(1+\mathrm{e}^{-x}\right)$，$P_n(\boldsymbol{v})$ 为域名负样本（与域名 d_i 不存在任意关联关系的域名）的概率分布，N_k 为负样本采样数。

证明　考虑式(2)的分母项需对所有样本进行计算，计算开销较大的问题，采用负采样技术[22]将式(2)转化为区分 R 中所有域名与 d_i 是否存在关联关系的逻辑回归任务，即

$$P'\left(d_j\mid d_i,\theta\right)=\log\sigma\left(\boldsymbol{v}_j'^{\mathrm{T}}\boldsymbol{v}_i\right)+\sum_{k=1}^{N_k}\mathrm{E}_{v_k\sim P_n(v)}\left[\log\sigma\left(-\boldsymbol{v}_k'^{\mathrm{T}}\boldsymbol{v}_i\right)\right]\tag{4}$$

参数 θ 更新需考虑 R 中不同关联关系的域名对，并且不同关联关系的域名对满足相互独立，通过取最大似然得到域名表示学习的目标函数为

$$O=\arg\max_{\theta}\prod_{R_r\in R}\prod_{(d_i,d_j)\in R_r}P'\left(d_j\mid d_i,\theta\right) \tag{5}$$

考虑式(5)连续乘法操作计算开销较大的问题，对式(5)等号两侧同取 log 函数，令 $L=\log O$，可得

$$L=\arg\min_{\theta}\left(-\sum_{R_r\in R}\sum_{(d_i,d_j)\in R_r}\log P'\left(d_j\mid d_i,\theta\right)\right) \tag{6}$$

考虑到 R 中不同关联关系域名对参数 θ 更新存在差异性影响[16,19]，在式(6)中为不同的关联关系引入权重因子，令 $\text{Weight}=\{w_{\text{MetaP1}},w_{\text{MetaP2}},w_{\text{MetaP3}},w_{\text{MetaP4}},w_{\text{MetaP5}},w_{\text{query}}\}$ 为所有域名关联关系的权重集合，最终可得基于关联信息权重自适应的域名表示学习的优化目标函数

$$L_{\text{Weight}}=\arg\min\left(-\sum_{R_r\in R}w_r\sum_{(d_i,d_j)\in R_r}\log P'\left(d_j\mid d_i,\theta\right)\right) \tag{7}$$

其中，$w_r\in\text{Weight}$，w_r 在训练过程中根据损失值自适应调整，推论 1 证毕。

采用小批量样本的随机梯度下降法对式(7)中参数 θ 和 Weight 进行交替更新，即完成 N_θ 次 θ 参数更新后，进行一次 Weight 参数更新，避免 Weight 中部分权重因频繁调整而取值过大，提高参数更新稳定性。令 r_θ 和 r_W 分别为参数 θ 和 Weight 的更新学习率，N_b 为小批量样本数量，参数更新具体步骤如下。

步骤 1 初始化 θ 与 Weight。

步骤 2 从 R 中每个关联关系域名对集合随机选择 N_b 个域名对，采用式(7)进行误差计算。

步骤 3 若已完成 N_θ 次 θ 参数更新，以学习率为 r_W 的随机梯度上升法对权重 Weight 进行更新；否则，转到步骤 4。

步骤 4 采用学习率为 r_θ 的梯度下降法对 θ 进行更新。

步骤 5 若未达到最大迭代次数，回到步骤 2；否则，输出 θ 与 Weight。

采用梯度上升法更新 Weight 的原因如下。对于任意 $w_r\in\text{Weight}$，若 w_r 的更新梯度值较大则说明式(7)中 R_r 类别的计算结果，即 $-\sum_{d_i\in R_r}\sum_{d_j\in N(d_i)}\log P'\left(d_j\mid d_i,\theta\right)$ 的值较大，从而可通过增大 R_r 类别域名对的权重 w_r 以获得更大的更新梯度，加快参数 θ 更新速度。

考虑所得的域名数值向量在特征空间分布上呈现关联性较强的 2 个域名向量距离较近的同类聚集特点，选取支持向量机（SVM, support vector machine）和随机森林（RF, random forest）作为域名分类器[23]。恶意域名检测器训练与测试流程如下：首先，通过已知域名黑白名单对域名数值向量进行标注；其次，随机选取部分带标签数据用于域名检测器训练；最后，通过训练完成的域名检测器检测未知标签域名。

3 实验与分析

3.1 实验环境与数据

实验环境如下：Windows 7 64 位操作系统，CPU 为 Intel Xeon Silver4114 2.2 GHz ，64 GB RAM，GPU 为 NVIDIA GeForce RTX 2080 SUPER，选取 Python 3.6 实现所提算法。

实验数据来源于 Malware Capture Facility Project，该项目在真实的主机和网络环境中采集僵尸网络、木马等恶意程序运行过程中产生的恶意流量数据和正常用户产生的合法流量数据。为验证所提方法进行恶意域名检测的有效性，筛选数据集中 DNS 流量作为实验数据，并采用恶意软件测试平台 VirusTotal 对所有域名进行标签标注。

3.2 实验参数与评价指标设置

AIEM-DNQT 算法中请求时间判别阈值 τ 为 5 s。域名表示学习参数设置如下：域名数值向量维度设置为 60，r_θ 设置为 5，r_W 设置为 0.05，每轮迭代负样本数 N_k 为 80。SVM 参数设置如下：惩罚系数 C 通过预实验确定为 5，核函数采用径向基函数；RF 中决策树数量通过预实验确定为 100，其余参数为默认设置。

TP（true positive）为被正确判别为恶意域名的样本数；FP（false positive）为被错误判别为恶意域名的样本数；TN（true negative）为被正确判别为合法域名的样本数；FN（false negative）为被错误判别为合法域名的样本数。主要参考的判别标准如下。

查准率为 $\text{Precision}=\dfrac{\text{TP}}{\text{TP}+\text{FP}}$。

查全率为 $\text{Recall}=\dfrac{\text{TP}}{\text{TP}+\text{FN}}$。

准确率为 $\text{Accuracy}=\dfrac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}}$，即所有被正确判别的样本数与样本总数的比值。

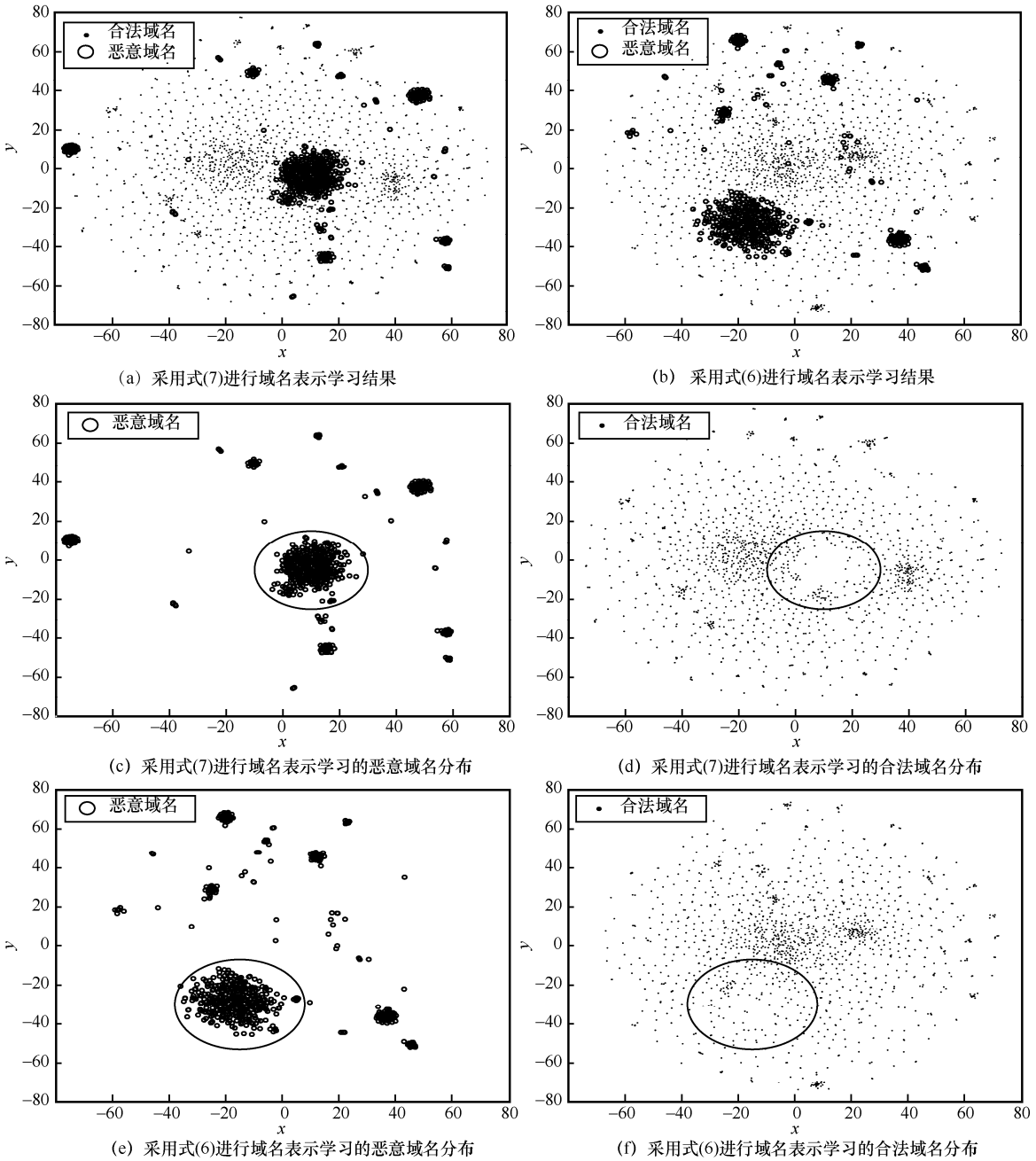
F1 分数为 $\text{F1}-\text{Score}=\dfrac{2\times\text{Precision}\times\text{Recall}}{\text{Precision+Recall}}$，反映查准率和查全率的综合性能，其值越大说明检测效果越好。

检测样本覆盖率（C-rate, coverage rate），即检测方法能检测的样本数与样本总数的比值。

3.3　域名表示学习方法对比

为验证所提域名表示学习方法的有效性，并分析域名数值向量在特征空间中的分布特点，分别采用式(6)和式(7)进行域名表示学习，并采用 t-SNE[24] 将域名向量降至 2 维进行可视化分析，结果如图 3 所示。

图 3(a)为式(7)所得域名向量的 2 维空间分布，域名向量呈现部分聚集的情况，并且位于同一聚集区域中的域名具有相同属性。由 t-SNE 的降维原理



(a) 采用式(7)进行域名表示学习结果　　(b) 采用式(6)进行域名表示学习结果

(c) 采用式(7)进行域名表示学习的恶意域名分布　　(d) 采用式(7)进行域名表示学习的合法域名分布

(e) 采用式(6)进行域名表示学习的恶意域名分布　　(f) 采用式(6)进行域名表示学习的合法域名分布

图 3　域名向量 2 维可视化

可知，在降维后的特征空间中处于同一个类簇的域名在原特征空间中也呈现聚类分布的特点。由图 3(c)与图 3(d)可知，采用式(7)进行域名表示学习可使存在关联关系的域名在高维特征空间中具有较小距离，不存在关联关系的域名间距离较大，并且不同类别的域名数值向量具有较好的可区分性。图 3(b)为采用式(6)进行域名表示学习所得的域名向量可视化结果，仍存在部分同类别的域名向量呈聚类的特点，但通过分析图 3(e)和图 3(f)中黑色框线内的样本分布可知，不同类别域名向量区分性不强，这将降低后续域名检测器性能，产生较高误报率。

3.4 域名检测器检测性能对比与漏报分析

按照不同比例将样本集划分为训练集与测试集，其中训练集用于训练 SVM 和 RF 分类器，测试集用于获得分类评价指标。表 1 给出了不同训练集占比下不同分类器的实验结果，实验结果为进行 10 次实验所得平均值，其中训练集占比为训练集样本数与数据集样本总数的比值。

表 1 不同训练集占比下 SVM、RF 检测性能对比

训练集占比	SVM		RF	
	F1 分数	准确率	F1 分数	准确率
10%	0.783	0.862	0.477	0.798
20%	0.71	0.97	0.65	0.907
30%	0.921	0.978	0.69	0.915
40%	0.936	0.98	0.723	0.922
50%	0.939	0.981	0.742	0.926
60%	0.945	0.981	0.761	0.931
70%	0.951	0.984	0.764	0.933
80%	0.955	0.985	0.781	0.932
90%	0.963	0.987	0.784	0.941

由表 1 可知，在相同的训练集占比下，SVM 的 F1 分数与准确率均优于 RF，主要是因为通过 2.4 节所得的域名向量在特征空间中具有较好的区分性，使通过 SVM 学习得到的支持向量能较好区分不同类别域名向量，在检测效果上优于基于特征选择实现集成决策的 RF。此外，在训练集占比仅为 30%时，SVM 的 F1 分数可达到 0.921，说明 MDND-RIQT 通过学习 DN-HIN 中元路径关联信息和请求时间关联信息得到区分性较好的域名向量，并结合 SVM 的小样本学习能力，取得较好检测效果。

采用 SVM 可获得较优的检测指标，但由于存在漏报，各项指标还有一定提升空间，所提检测方法产生漏报的主要原因是部分恶意域名与合法域名存在关联关系，主要包含以下 2 种情况：1) 攻击者将恶意服务器部署到云/VPS 平台，使恶意域名的解析 IP 地址与部署在同平台的合法域名存在关联，从而造成此类恶意域名的数值向量与合法域名具有相似的数值向量分布，进而被域名检测器误判为合法域名；2) 攻击者通过渗透手段掌握部分站点控制权进行恶意活动，如上传恶意篡改软件供用户下载、在网页中挂载恶意程序等，由于此类攻击事件中的域名只存在与其他合法域名的关联信息，导致所提方法无法检出此类恶意域名利用方式，将此类域名误判为合法。为减少以上两类漏报产生，还需针对恶意域名的利用方式进行分析，以提高检测方法稳健性。

3.5 不同关联信息和表示学习方法的检测性能对比

采用控制变量法设计对比实验，以检验不同关联信息与表示学习方法对检测结果的影响，所得对比结果如表 2 所示，MDND-RI 代表未采用域名请求时间关联信息的 MDND-RIQT 方法，MDND-RIQT-Equal 为采用式(6)进行域名表示学习的 MDND-RIQT 方法。对比实验中域名检测器均为 SVM，训练集占比均为 70%。

表 2 不同实验设置的检测性能对比

方法	F1 分数	准确率	C-Rate
Malshoot[15]	0.933	0.961	0.555
MDND-RI	0.954	0.982	0.819
MDND-RIQT-Equal	0.908	0.969	0.977
本文方法	0.951	0.984	0.977

由表 2 可知，Malshoot 的 C-Rate 和检测准确率最低，其主要原因为 Malshoot 仅提取域名解析 IP 地址的二阶相似度用于域名表示学习，导致大量域名因缺乏基于 IP 地址的关联信息而无法被检测；MDND-RI 采用 2.2 节提出的 5 种元路径关联信息进行域名检测，检测指标较 Malshoot 均有提升，其中 C-Rate 增长明显，但仍有 19.1%的域名由于关联解析信息缺失无法被检测；MDND-RIQT-Equal 方法结合域名解析信息和请求时间两方面的关联信息，C-Rate 达到最高，但在将关联信息转化为数值向量过程中，未能区分不同域名关联关系对目标函数优化的差异性影响，导致部分域名向量更新不足，所

得 F1 分数较低。MDND-RIQT 通过结合域名元路径和请求时间关联信息，并采用域名关联信息权重自适应的域名向量学习方法进行域名检测，各项指标均为最优。

域名数值向量包含域名基于元路径与基于请求时间的关联信息，域名向量维度的设置影响后续域名检测的性能。为说明域名向量维度的设置对检测效果的影响，设置不同维度进行实验，实验结果如图 4 所示。

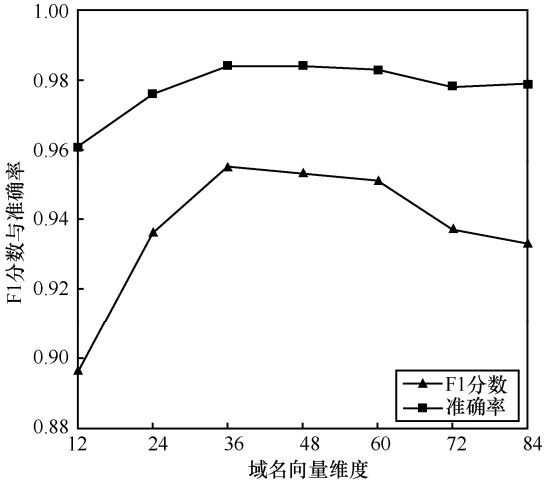


图 4　域名向量维度对检测性能的影响

由图 4 可知，随着域名向量维度增加，F1 分数与准确率均增加并最终稳定在一定范围内。当维度设置为 12，由于向量维度过小，无法有效表征域名之间丰富关联信息，F1 分数和准确率取值最低；当维度设置为 72 或 84 时，检测指标略微降低，说明维度设置过大存在一定过拟合风险；当维度分别设置为 36、48 或 60 时，检测指标受维度调整引起的波动较小，从而在参数调整过程中能较快取得检测指标较优的向量维度设置。

3.6　与基于 BP 算法的恶意域名检测方法对比

基于图模型的恶意域名检测研究通常基于 BP 算法进行恶意域名检测[5-11]，此类方法能在仅有少量域名节点带有标签的情况下，通过节点间消息传递的方式为对未知标签域名节点进行标记，降低恶意域名检测中对大量标签数据的依赖。将所提方法与基于 BP 的恶意域名检测方法[10]进行对比，采用不同训练集占比进行实验，所得 F1 分数对比情况如图 5 所示，其中阈值用于判定域名标签，当 BP 算法迭代收敛后，若域名标签数值大于阈值，判别为恶意域名。由于初始标签设置为 0.5，分别选择 0.49 和 0.51 作为阈值，以检验阈值设置对检测结果的影响。

由图 5 可知，随着训练集占比增加，BP 算法的 F1 分数逐渐增加并最终保持稳定。当训练集占比小于 70%，BP 算法的 F1 分数受阈值设置影响较大，其主要原因如下：1) 当训练集占比较小时，域名图中大量域名节点初始标签为 0.5，导致采用 BP 算法进行节点标签更新后，其标签仍为 0.5；2) 由于样本集中合法域名数量远多于恶意域名数量，当阈值设定为 0.51，标签为 0.5 的域名被判为正常域名，导致样本集中少数恶意域名因标签为 0.5 被误判为合法域名，此时检测误报率较低，从而具有较高的 F1 分数；当阈值设置为 0.49，导致大量标签为 0.5 的合法域名被误判为恶意，产生较多误报，所得 F1 分数较小。与 BP 算法相比，所提方法可在已知标签数据较少的情况取得较高的 F1 分数。

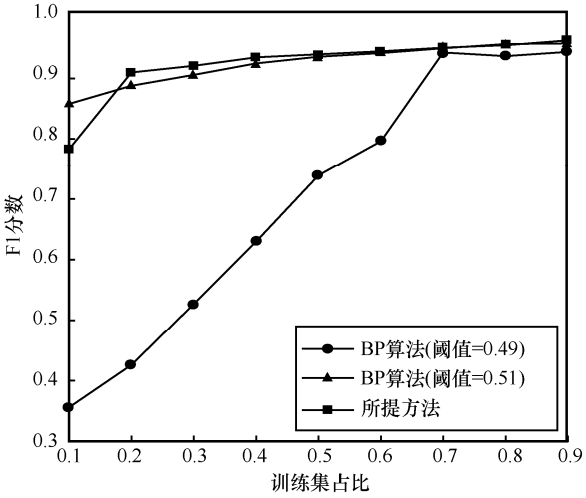


图 5　所提方法与 BP 算法对比

4　结束语

本文提出一种结合域名解析 IP 地址、别名记录和请求时间关联信息的恶意域名检测方法。该方法采用 HIN 表示域名解析信息，设计了基于元路径的网络遍历方法，以提高域名关联解析信息提取效率。引入请求时间关联信息有效检测弱连接域名，提高了检测方法的样本覆盖率。设计了域名表示学习方法融合不同关联信息，通过向量间欧氏距离量化域名关联程度。实验结果表明，所提方法在已知标签数据较少的情况下域名检测效果较优。下一步研究将引入域名注册信息、WHOIS 信息用于域名关联信息挖掘，进一步提高检测精度。

参考文献：

[1] ZHAUNIAROVICH Y, KHALIL I, YU T, et al. A survey on malicious domains detection through DNS data analysis[J]. ACM Computing Surveys, 2018, 51(4): 1-36.

[2] GAO H Y, YEGNESWARAN V, JIANG J, et al. Reexamining DNS from a global recursive resolver perspective[J]. IEEE/ACM Transactions on Networking, 2016, 24(1): 43-57.

[3] WANG X, ZHENG K F, NIU X X, et al. Detection of command and control in advanced persistent threat based on independent access[C]//Proceedings of 2016 IEEE International Conference on Communications (ICC). Piscataway: IEEE Press, 2016: 1-6.

[4] 彭成维, 云晓春, 张永铮, 等. 一种基于域名请求伴随关系的恶意域名检测方法[J]. 计算机研究与发展, 2019, 56(6): 1263-1274.
PENG C W, YUN X C, ZHANG Y Z, et al. Detecting malicious domains using co-occurrence relation between DNS query[J]. Journal of Computer Research and Development, 2019, 56(6): 1263-1274.

[5] YEDIDIA J S, FREEMAN W T, WEISS Y. Understanding belief propagation and its generalizations[J]. Exploring Artificial Intelligence in the New Millennium, 2003, 8: 236-239.

[6] MANADHATA P K, YADAV S, RAO P, et al. Detecting malicious domains via graph inference[M]. Cham: Springer International Publishing, 2014.

[7] KHALIL I, YU T, GUAN B. Discovering malicious domains through passive DNS data graph analysis[C]//Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. New York: ACM Press, 2016: 663-674.

[8] LEE J, LEE H. GMAD: graph-based malware activity detection by DNS traffic analysis[J]. Computer Communications, 2014, 49: 33-47.

[9] 臧小东, 龚俭, 胡晓艳. 基于 AGD 的恶意域名检测[J]. 通信学报, 2018, 39(7): 15-25.
ZANG X D, GONG J, HU X Y. Detecting malicious domain names based on AGD[J]. Journal on Communications, 2018, 39(7): 15-25.

[10] PENG C W, YUN X C, ZHANG Y Z, et al. Discovering malicious domains through alias-canonical graph[C]//Proceedings of 2017 IEEE Trustcom/BigDataSE/ICESS. Piscataway: IEEE Press, 2017: 225-232.

[11] ZOU F T, ZHANG S Y, RAO W X, et al. Detecting malware based on DNS graph mining[J]. International Journal of Distributed Sensor Networks, 2015, 2015: 1-12.

[12] SUN Y Z, HAN J W. Mining heterogeneous information networks: principles and methodologies[J]. Synthesis Lectures on Data Mining and Knowledge Discovery, 2012, 3(2): 1-159.

[13] TANG J, QU M, WANG M Z, et al. LINE: large-scale information network embedding[C]//Proceedings of the 24th International Conference on World Wide Web. New York: ACM Press, 2015: 1067-1077.

[14] LEI K, FU Q A, NI J K, et al. Detecting malicious domains with behavioral modeling and graph embedding[C]//Proceedings of 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). Piscataway: IEEE Press, 2019: 601-611.

[15] PENG C W, YUN X C, ZHANG Y Z, et al. MalShoot: shooting malicious domains through graph embedding on passive DNS data[M]. Cham: Springer International Publishing, 2019.

[16] SUN X Q, TONG M K, YANG J H. HinDom: a robust malicious domain detection system based on heterogeneous information network with transductive classification[C]//Proceeding of the 22nd International Symposium on Research in Attacks, Intrusions and Defenses. Berkley: USENIX Association, 2019: 399-412.

[17] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks[J]. arXiv Preprint, arXiv: 1609.02907, 2016.

[18] LIU Z, LI S, ZHANG Y, et al. Ringer: systematic mining of malicious domains by dynamic graph convolutional network[C]//Proceeding of the International Conference on Computational Science. Berlin: Springer, 2020: 379-398.

[19] SUN X Q, YANG J H, WANG Z L, et al. HGDom: heterogeneous graph convolutional networks for malicious domain detection[C]//Proceedings of 2020 IEEE/IFIP Network Operations and Management Symposium. Piscataway: IEEE Press, 2020: 1-9.

[20] HE W X, GOU G P, KANG C C, et al. Malicious domain detection via domain relationship and graph models[C]//Proceedings of 2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC). Piscataway: IEEE Press, 2019: 1-8.

[21] NSFOCUS. 2019 Botnet trend report[R]. NSFOCUS Security Labs, 2020.

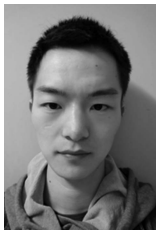
[22] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[C]//Proceeding of the Advances in Neural Information Processing Systems. Massachusetts: MIT Press, 2013: 3111-3119.

[23] SCHÜPPEN S, TEUBERT D, HERRMANN P, et al. FANCI: feature-based automated NXDomain classification and intelligence[C]//Proceeding of the 27th USENIX Security Symposium. Berkley: USENIX Association, 2018: 1165-1181.

[24] VAN D M L, HINTON G. Visualizing data using t-SNE[J]. Journal of machine learning research, 2008, 9(11): 2579-2605.

[作者简介]

张斌（1969− ），男，河南南阳人，博士，信息工程大学教授、博士生导师，主要研究方向为信息系统安全。



廖仁杰（1996− ），男，四川泸州人，信息工程大学硕士生，主要研究方向为基于机器学习的恶意域名检测。