基于双向 LSTM 的误植域名滥用检测方法

吕 品[1,2],李全刚[1],柳厅文[1],宁振虎[3],王玉斌[1,2],时金桥[1],方滨兴[4,1]
(1. 中国科学院信息工程研究所,北京 100093;2. 中国科学院大学网络空间安全学院,北京 100049;
3. 北京工业大学信息学部,北京 100124;4. 电子科技大学广东电子信息工程研究院,广东东莞 523808)

摘 要: 当前,误植域名检测主要以计算域名对之间的编辑距离为基础,未能充分挖掘域名的上下文信息,且对短域名的检测易产生大量的假阳性结果。采集域名相关信息进行判定虽然有助于提高检测效果,却会引入较大的额外开销. 本文采用了基于域名字符串的轻量级检测策略,并引入双向长短时记忆模型(LSTM,Long Short-Term Memory)来充分利用域名上下文,提升检测效果. 本文还设计了面向域名的局部敏感哈希函数,以提高在大规模域名集合上进行误植域名检测的速度. 在大量真实数据集上的实验结果表明,本文的工作改进了基于编辑距离检测方法的不足,能够有效地进行误植域名滥用检测.

关键词: 误植域名;编辑距离;双向 LSTM;上下文信息;局部敏感哈希
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2018)09-2081-06
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2018.09.006

Towards Typosquatting Abuse Detection using Bi-directional LSTM

LÜ Pin[1,2],LI Quan-gang[1],LIU Ting-wen[1],NING Zhen-hu[3],WANG Yu-bin[1,2],SHI Jin-qiao[1],FANG Bin-xing[4,1]
(1. *School of Cyber Security, University of Chinese Academy of Sciences, Beijing* 100049, *China*;
2. *Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093, *China*;
3. *Faculty of Information Technology, Beijing University of Technology, Beijing* 100124, *China*;
4. *University of Electronic Science and Technology Guangdong Institute of Electronic Information Engineering, Dongguan, Guangdong* 523808, *China*)

Abstract: Prior works on detection of typosquatting abuse are based on the calculation of edit distance between domains. They do not fully utilize the context information of domains, and usually give many false positive results for short domains. Actively crawling much related information of the given domains can help improving the results, but introduce a heavy overhead. Therefore, we design a lightweight detecting strategy based on domain names, and introduce the bi-directional long short-term memory (LSTM) model to make full use of the domain context information. Furthermore, we give a locality sensitive hashing function for domain names, in order to increase the speed of typosquatting abuse detection over large-scale domain sets. Experimental results on a real data set show that the proposed method can overcome the shortcomings of edit distance based methods, and can detect typosquatting abuse efficiently.

Key words: typosquatting domain; edit distance; bi-directional LSTM; context information; locality sensitive hashing

1 引言

当前,各种针对网络通信域名服务系统的攻击层出不穷. 其中,通过注册近似域名(typosquatting),引诱用户访问该域名(也称误植域名,如 baiidu.com、appl1e.com 等)而非目标网站,以发布假广告、售卖假商品,甚至骗取使用者信息并进行身份盗窃等,已经成为威胁互联网安全运行的重要问题之一.

误植域名检测的相关工作主要是面向对误植域名滥用行为的测量[1~7],这些工作在收集误植域名时基本都是以编辑距离为核心进行的,但并没有充分利用域名字符串的上下文信息,并且对短域名的检测易产生大量的假阳性结果. 收集域名的相关信息,如 Whois 信息、网页内容等,可以提高对误植域名判定的精度,但会带来的计算和存储开销,不适合应用在高速网络环境下的检测场景.

同时,也有相关工作采用了深度学习方法. Wood-

收稿日期:2017-06-15;修回日期:2017-11-26;责任编辑:覃怀银
基金项目:国家重点研发计划(2016YFB0801003);东莞市引进创新科研团队计划资助(No. 201636000100038)

bridge 等人提出了基于 LSTM 的 DGA 域名检测方法[8]，以及 DGA 域名的对抗生成和检测[9]. 周昌令等人提出了基于深度学习的域名查询行为向量空间嵌入方法，并进一步基于域名或主机的向量表示发现网络中的僵尸网络等异常行为[10]. Saxe 等人提出了一种基于卷积神经网络的方法，可以用于检测恶意 URL[11]. Kobojek 等人提出了一种基于循环神经网络的用户识别方法，主要利用了用户敲击键盘的一些特点[12]. 但目前都尚未应用于误植域名检测上.

为了解决上述问题，本文设计了基于域名字符串的误植域名检测方法，并利用深度学习技术，首次引入双向 LSTM（Bi-directional Long Short Term Memory，BiLSTM）构建分类器，从而能够充分利用域名上下文信息，解决了传统基于编辑距离设计的检测方法的不足.

2 问题描述与系统框架

本节主要介绍误植域名检测问题的形式化定义，以及针对该问题提出的基于 BiLSTM 的检测系统框架.

2.1 问题描述

在误植域名判定时，通常是相对一个知名域名或者受保护域名（记为 S）而言的. 对待判定域名 d，如果 d 是根据 S 中某个元素仿冒的域名，则认为域名 d 是误植域名. 在实际应用时，集合 S 的规模（$|S|$）决定了域名 d 的相对判定范围. 集合 S 的实际选取与具体应用需求密切相关.

在本文中，针对待判定域名 d 和给定的域名集合 S，误植域名检测问题的形式化定义如下：

$$F(d,S)=\begin{cases}1, & \exists t\in S, f(d,t)=1\\ 0, & \forall t\in S, f(d,t)=0\end{cases} \tag{1}$$

其中，函数 $F(d,S)$ 取值为 1 时表明 d 是 S 中至少一个元素的仿冒域名，否则取值为 0. 在域名集合 S 给定的前提下，函数 $F(d,S)$ 可以简记 $F(d)$. 函数 $f(d,t)$ 用来判定域名 d 是否为域名 t 的仿冒域名.

2.2 基于 BiLSTM 的系统框架

基于 BiLSTM 的误植域名检测框架如图 1 所示，分为离线的模型训练和在线的域名判定两个阶段. 模型训练阶段会构建一个基于 BiLSTM 的神经网络分类器供域名判定阶段使用. 模型训练是有监督学习的，需要大量带标签的训练样本. 每个训练样本是一个三元组 $\langle d1,d2,f(d1,d2)\rangle$，$d1$ 和 $d2$ 是域名，$f(d1,d2)$ 是取值为 0 或 1 的标签，表明 $d1$ 是否为 $d2$ 的误植域名. 本文使用的训练样本构建过程详见第 6 节，训练过程详见第 4 节.

在域名判定阶段，会将基于 BiLSTM 的神经网络分类器作为黑盒使用，分类器的输入为由两个域名构成的域名对，输出的取值为 0 或 1，表明前一个域名是否为后一个域名的仿冒域名. 域名判定阶段会针对待判定域名 d 与域名集合 S 中各元素的分类结果给出 $F(d,S)$ 的取值. $F(d,S)$ 的计算过程见第 3 节.

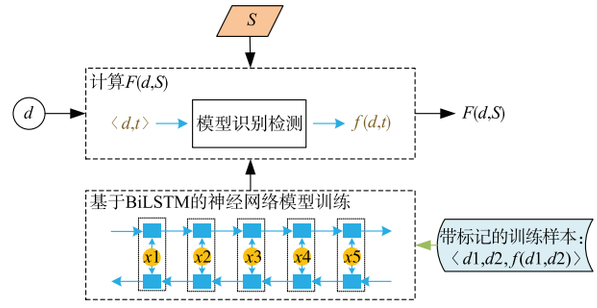

图1 基于BiLSTM的误植域名检测框架

3 $F(d,S)$ 的计算过程

根据式（1）可知，一旦我们利用分类器获知域名 d 是 S 中某个元素的仿冒域名，那么就可知 $F(d,S)$ 的取值必然为 1，此时可以舍弃 d 与 S 中尚未处理的元素的分类判定. 详细计算过程如图 2 所示.

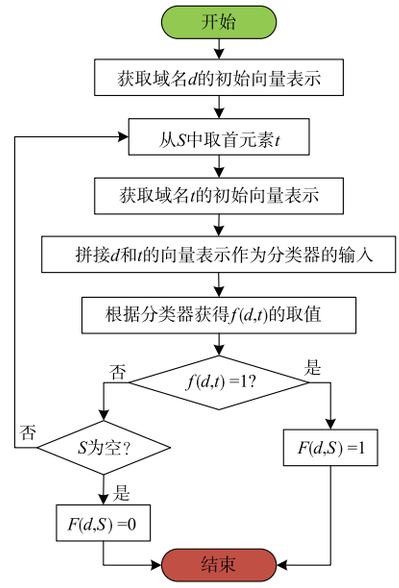

图2 $F(d,S)$的计算过程

从集合 S 中不放回地获取域名，并与待判定域名 d 构成域名对，作为分类器的输入. 直到分类器对某个域名对的输出结果为 1，或者对所有域名对的输出结果都为 0，整个域名判定过程结束. 在实际网络环境中，误植域名所占比例通常是非常低的，大部分情况下需要执行 $|S|$ 次.

4 基于 BiLSTM 的神经网络模型训练

在本节中，我们将仿冒域名的判断问题转化为域

名对的分类问题,提出了一种基于 BiLSTM 的神经网络模型. LSTM 网络是一种循环网络结构[13],在自然语言处理领域应用广泛,其特点是对历史信息的选择性利用与更新,有效实现数据长时特性的特征提取. BiLSTM 进一步结合历史信息与将来信息,即将当前字符的分类特征扩充至整个语句,同时采用 Dropout 等方法避免过拟合,进一步提升了效果.

具体方法是,首先将域名对〈正常域名,测试域名〉根据域名中各字符的字典编号进行 one-hot 编码表示. 将两个域名的 one-hot 编码补齐并拼接,然后输入到神经网络中进行字符向量化表示学习和 BiLSTM 学习;最后通过 Softmax 层转换为 0/1 分类标签. 标签为 1 表示测试域名为仿冒域名,标签为 0 则表示不是仿冒域名. 图 3 给出了基于 BiLSTM 的神经网络模型的结构图.

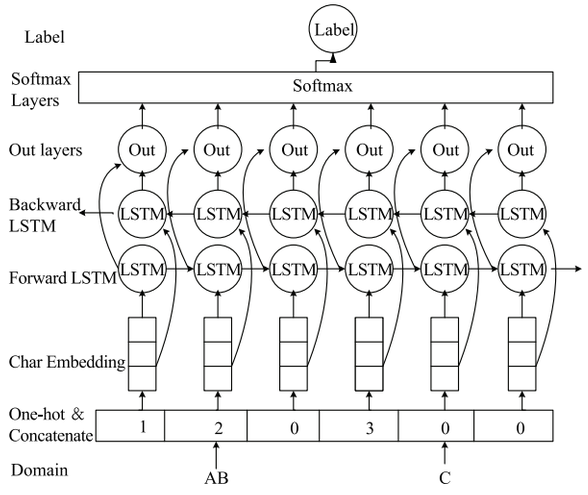

图3　BiLSTM模型训练网络结构图

5　基于局部敏感哈希的加速判定

如上文所述,在真实应用环境中,大部分情况下分类器需要执行$|S|$次. 当域名集合 S 较大时,整个判定过程会非常耗时. 因此,本节引入局部敏感哈希来提高判定速度.

局部敏感哈希(LSH)是在高维空间中解决近似最近邻快速查找的一种算法,通常在大量数据中查找近似数据的应用场景使用,如查找网络上的重复网页、相似图像检索等[14~16]. 对于一些精心设计的 LSH,如果原始空间中相邻的数据落入相同的桶内的话,我们只需要将查询数据进行哈希映射得到其桶号,然后取出该桶号对应桶或者附近桶内的所有数据,再进行线性匹配即可查找到与查询数据相邻的数据.

基于局部敏感哈希的误植域名检测问题的形式化定义如下:

$$F(d,S)=\begin{cases}1, & \exists\, t\in S_d^{\theta}, f(d,t)=1\\ 0, & \forall\, t\in S_d^{\theta}, f(d,t)=0\end{cases} \tag{2}$$

其中

$$S_d^{\theta}=\{t\mid t\in S,\mathrm{distance}(H(t),H(d))\leq\theta\}$$

其中,$H(\cdot)$是局部敏感哈希函数得到的桶号. S_d^{θ} 中每个元素在局部敏感哈希后的桶号,距离域名 d 的桶号不超过阈值 θ.

由于域名数据的长度通常较短,而传统的基于分词的局部敏感哈希函数 SimHash[14] 通常用于处理长文本,不能更好的对域名数据进行快速查找. 因此,本文设计了一种专门针对域名数据的哈希函数.

假设域名字符集为 $C=\{c_1,c_1,\cdots,c_n\}$,首先我们基于词带模型将各个域名 d 表示为字符频率向量 $v(d)=\langle f(d,c_1),f(d,c_2),\cdots,f(d,c_n)\rangle$. 在查找距离不超过 θ 的近似域名时,我们可以将域名的字符频率向量均分为 $\theta+1$ 个连续的向量片段 $\boldsymbol{v}(d)\rightarrow\langle v_1(d),v_2(d),\cdots,v_{\theta+1}(d)\rangle$. 根据鸽巢原理,如果某个域名 d 与待查找域名 d' 的距离不超过 θ,那么至少存在一个向量片段 $\boldsymbol{v}(d)$ 与 $\boldsymbol{v}(d')$ 相等,据此可以借助预先构建已有域名各个向量片段的哈希表来实现对查找过程的加速. 通常 θ 的取值较小,如 2、3,因此可以认为实际构建哈希表的代价与域名数量为线性关系.

6　实验与结果

在本节中,我们使用本文提出的技术构建了一个误植域名检测的原型系统,并且在真实数据集上进行了测试. 本文的神经网络模型是基于 Keras 构建的,运行环境为 16GB 内存、16 核 CPU 的虚拟服务器. 本文所选用的对比实验方案采用了目前最常用的基于编辑距离的判定方案.

6.1　度量标准

对本文的工作,主要从分类器和整体系统两方面来度量. 分类器的评价指标包括精确率、召回率和 $F1$ 值,整体系统的评价指标包括筛选率、覆盖率、精确率、召回率、$F1$ 值和处理速度. 我们定义分类器的精确率、召回率和 $F1$ 值公式如下:

$$\mathrm{Precision}_c=\frac{\sum_{\langle d1,d2\rangle\varepsilon T_c} f(d1,d2)\times f^*(d1,d2)}{\sum_{\langle d1,d2\rangle\varepsilon T_c} f^*(d1,d2)} \tag{3}$$

$$\mathrm{Recall}_c=\frac{\sum_{\langle d1,d2\rangle\varepsilon T_c} f(d1,d2)\times f^*(d1,d2)}{\sum_{\langle d1,d2\rangle\varepsilon T_c} f(d1,d2)} \tag{4}$$

$$F1_c=\frac{2\times\mathrm{Precision}_c\times\mathrm{Recall}_c}{\mathrm{Precision}_c+\mathrm{Recall}_c} \tag{5}$$

其中,T_c是用于评价分类器的测试集,$f(d1,d2)$是 T_c中

测试样本的标签,取值为 0 或 1,$f^*(d1,d2)$是分类器给出的预测标签.

筛选率、覆盖率是用来衡量局部敏感哈希函数在域名集合上进行近似最近邻快速查找效果的两个指标. 我们定义整体系统的筛选率、覆盖率如下:

$$\text{Filter}_s = \frac{\sum_{d\in T_s} |S_d^\theta|}{|T_s| \times |S|} \tag{6}$$

$$\text{Coverage}_s = \frac{|\{d|\langle d,d'\rangle \in T_c, f(d,d') = 1, d' \} \in S_d^\theta|}{|\{d|\langle d,d'\rangle \in T_c, f(d,d') = 1\}|} \tag{7}$$

其中,T_s是用于评价整体系统的测试集. 本文中集合 T_s 是对集合 T_c进行处理得到的:$T_s = \{d|\langle d,d'\rangle \in T_c\}$ $F(d,S)$是 T_s中测试样本 d 针对集合 S 的标签,取值为 0 或 1. $F(d,S) = 1$ 当且仅当 $d \in T_s$并且 $d' \in S$ 使得 $f(d,d') = 1$.

整体系统的精确率、召回率、$F1$ 值和处理速度定义如下:

$$\text{Precision}_s = \frac{\sum_{d\in T_s} F(d,S) \times F^*(d,S)}{\sum_{d\in T_s} F^*(d,S)} \tag{8}$$

$$\text{Recall}_s = \frac{\sum_{d\in T_s} F(d,S) \times F^*(d,S)}{\sum_{d\in T_s} F(d,S)} \tag{9}$$

$$F1_c = \frac{2 \times \text{Precision}_s \times \text{Recall}_s}{\text{Precision}_s + \text{Recall}_s} \tag{10}$$

$$\text{Speed}_s = \frac{|T_s|}{t} \tag{11}$$

其中 $F^*(d,S)$是系统给出的 T_s中测试样本 d 在集合 S 上的预测标签,t 是系统处理测试集 T_s所耗费的时间,本文计时单位为秒.

6.2 数据集

正样本数据构造:我们首先利用 NCC Group 在 Github 上开源的 typofinder 工具获得其误植域名列表(该工具采用已知常见的误植域名构造模式,并主动获取候选域名的相关信息来判定其是否为真实的误植域名,准确率相对较高),对来自 Alexa 排名前 500 的域名,限制域名长度为 20 以内,总计生成了 26273 个误植域名,相应地得到 26273 个标签为 1 的域名对,作为正样本.

负样本数据构造:我们从 Alexa 排名前 10000 个域名中随机选择,域名长度同样限制为不超过 20,构造了编辑距离不超过 3 的 29998 个域名对. Alexa 前 10000 个域名基本都是访问量比较大的知名域名,因此可以基本判定不是误植域名. 这些标签为 0 的 29998 个域名对,构成了负样本.

因此,实验数据集是规模为 56271,其中正样本 26273 个、负样本 29998 个. 其中,选取 70% 用于构建模型,30% 用于评价模型,即作为 T_c. 在构建模型的 70% 数据中,90% 用于训练,10% 用于验证.

本论文中嵌入层 Embedding 的输出维度设为 100,BiLSTM 层输出维度设为 100,Dropout 参数设置为 0.3,优化算法选择为 rmsprop,损失函数选择为 categorical_crossentropy,batch_size 和 epoch 分别设为 100 和 50.

6.3 模型分类器的实验结果

我们将本文的模型分类器与基于编辑距离的判定方法进行了对比. 在实际应用中,编辑距离的阈值 δ 通常取 1 或 2. 表 1 给出了对比实验结果. 本文提出的模型分类器较基于编辑距离的判定方法取 $\delta = 1$ 和 $\delta = 2$ 时,$F1$ 值分别提升了约 2.78% 和 11.2%. 可以看到,基于编辑距离的方法在 $\delta = 2$ 时,精确率较低,这是由于实际存在如 qq.com 及 jd.com 的域名对,虽然编辑距离很小,但显然两者不存在仿冒现象.

表 1 模型分类器的实验结果

方法	精确率	召回率	$F1$ 值
基于编辑距离的方法($\delta = 1$)	0.9851	0.9122	0.9472
基于编辑距离的方法($\delta = 2$)	0.8170	0.9439	0.8758
本文方法	0.9720	0.9751	0.9735

6.4 整体系统的实验结果

我们将本文提出的局部敏感哈希函数与传统的 SimHash 在加速效果上作了对比,如表 2 所示. 实验证明,本文方法的筛选率和覆盖率均远优于 SimHash. 同时,随着阈值 θ 增大,筛选率和覆盖率越高,但 θ 越大,代表集合 S 的域名被约减得越多,引入的误差也就越大. 因此,我们需要根据集合 S 的规模,灵活调整阈值 θ 的值,以达到时间开销与准确率之间的平衡. 这里我们选择 $\theta = 2$ 作为后续实验的阈值 θ 的取值.

表 2 筛选率与覆盖率随阈值的 θ 的变化

θ	本文方法		SimHash	
	筛选率	覆盖率	筛选率	覆盖率
1	0.0063	0.5312	3.584×10^{-5}	0.0162
2	0.0869	0.9863	8.071×10^{-5}	0.0295
3	0.1358	0.9974	3.356×10^{-4}	0.0664

在阈值 θ 确定的情况下,对于不同规模的正常域名集合 S,基于局部敏感哈希的加速判定的方法与线性逐一比较集合 S 中域名的方法的处理速度对比如表 3 所示. 可以看到,通过局部敏感哈希加速判定的方法显著节省了时间开销,加速比随着集合 S 的规模增大而增大,当集合规模为 10000 时加速比已达到 2000 余倍.

表 3　系统处理速度对比

集合 S 规模	线性判定速度	加速判定速度	加速比
500	0.1942 s^{-1}	115.74 s^{-1}	596
1000	0.0971 s^{-1}	79.05 s^{-1}	814
2000	0.0485 s^{-1}	51.74 s^{-1}	1067
5000	0.0194 s^{-1}	30.59 s^{-1}	1577
10000	0.0097 s^{-1}	20.06 s^{-1}	2068

对于仿冒域名判定问题,我们也和基于编辑距离的方法进行了对比,后者同样使用了基于局部敏感哈希的加速判定. 由表 4 可以看出,本文提出的方法相比基于编辑距离的判定方法取 $\delta = 1$ 和 $\delta = 2$ 时,$F1$ 值分别提升了约 9.85% 和 40.1% .

表 4　整体系统的实验结果

方法	精确率	召回率	$F1$ 值
基于编辑距离的方法($\delta = 1$)	0.8662	0.8850	0.8755
基于编辑距离的方法($\delta = 2$)	0.5548	0.9002	0.6865
本文方法	1.0	0.9262	0.9617

7　总结

针对误植域名检测问题,本文提出了一种轻量级的基于域名字符串的检测方法. 该方法引入双向 LSTM 来构建误植域名检测的分类器,从而能够充分利用域名上下文信息,解决了传统基于编辑距离设计的检测方法的不足. 同时,本文还设计了面向域名的局部敏感哈希函数,与 Simhash 相比,该函数在域名数据上能减少在大规模域名集合上的筛选率并提升有效覆盖率. 实验结果表明本文的工作在短域名上也能获得较高的准确率和召回率.

参考文献

[1] Moore T, Edelman B. Measuring the perpetrators and funders of typosquatting[A]. Financial Cryptography and Data Security[C]. New York:Springer Berlin Heidelberg, 2010. 175 – 191.

[2] Wang Y M, Beck D, Wang J, et al. Strider typo-patrol: discovery and analysis of systematic typo-squatting[A]. Conference on Steps To Reducing Unwanted Traffic on the Internet[C]. San Jose, CA: USENIX Association, 2006. 5.

[3] Vissers T, Joosen W, Nikiforakis N. Parking sensors: analyzing and detecting parked domains[A]. The 2015 Network and Distributed System Security Symposium[C]. San Diego, California: Internet Society, 2015. 53.

[4] Liu T, Zhang Y, Shi J, et al. Towards quantifying visual similarity of domain names for combating typosquatting abuse[A]. 2016 IEEE Military Communications Conference [C]. Baltimore, MD, USA: IEEE, 2016. 770 – 775.

[5] Khan M T, Huo X, Li Z, et al. Every second counts: quantifying the negative externalities of cybercrime via typosquatting[J]. Neural Computation, 2015, 6(5): 135 – 150.

[6] Banerjee A, Barman D, Faloutsos M, et al. Cyber-fraud is one typo away[A]. Proceedings of the 27th Conference on Computer Communications[C]. Phoenix, AZ, USA: IEEE, 2008. 1939 – 1947.

[7] Banerjee A, Rahman M S, Faloutsos M. SUT: Quantifying and mitigating url typosquatting[J]. Computer Networks, 2011, 55(13): 3001 – 3014.

[8] Woodbridge J, Anderson H S, Ahuja A. Predicting Domain Generation Algorithms with Long Short-Term Memory Networks[DB/OL]. https://arxiv.org/abs/1611.00791, 2016 – 11 – 02.

[9] Anderson H S, Woodbridge J, Filar B. DeepDGA: adversarially-tuned domain generation and detection[A]. Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security[C]. New York, NY, USA: ACM, 2016. 13 – 21.

[10] 周昌令,栾兴龙,肖建国. 基于深度学习的域名查询行为向量空间嵌入[J]. 通信学报,2016,37(3):165 – 174.

[11] Saxe J, Berlin K. eXpose: A Character-Level Convolutional Neural Network with Embeddings for Detecting Malicious URLs, File Paths and Registry Keys[DB/OL]. https://arxiv.org/abs/1702.08568, 2017 – 02 – 27.

[12] Kobojek P, Saeed K. Application of recurrent neural networks for user verification based on keystroke dynamics [J]. Journal of Telecommunications and Information Technology, 2016 (3): 80.

[13] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735 – 1780.

[14] Manku G S, Jain A, Das Sarma A. Detecting near-duplicates for web crawling[A]. Proceedings of the 16th International Conference on World Wide Web[C]. Banff, Alberta, Canada: ACM, 2007. 141 – 150.

[15] Jing Y, Baluja S. Visualrank: Applying pagerank to large-scale image search[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30 (11): 1877 – 1890.

[16] Slaney M, Casey M. Locality-sensitive hashing for finding nearest neighbors[lecture notes][J]. IEEE Signal Processing Magazine, 2008, 25(2): 128 – 131.

作者简介

吕 品 男,1982 年 9 月出生于河北省曲阳县. 现为中国科学院大学网络空间安全学院、中国科学院信息工程研究所在读博士研究生,高级工程师,主要研究方向为信息安全、网络安全监测。
E-mail:lvpin@ iie. ac. cn

李全刚(通讯作者) 男,1986 年 1 月出生于山东省安丘市. 2015 年毕业于电子科技大学. 现为中科院信息工程研究所助理研究员,主要研究方向为信息内容安全等.
E-mail:liquangang@ iie. ac. cn

柳厅文 男,1986 年 5 月出生于安徽省临泉县. 2013 年博士毕业于中国科学院大学. 现为中国科学院信息工程研究所副研究员,主要研究方向为大数据分析与知识发现等.
E-mail:liutingwen@ iie. ac. cn

宁振虎 男,1983 年 9 月出生于河北省邯郸市. 现为北京工业大学信息学部讲师,主要研究方向为信息安全和可信计算。
E-mail:nzh41034@ 163. com

王玉斌 男,1991 年 8 月出生于河北省保定市. 中国科学院信息工程研究所在读博士生,主要研究方向为大数据分析.
E-mail:wangyubin@ iie. ac. cn

时金桥 男,1978 年 1 月出生于黑龙江省哈尔滨市. 2007 年博士毕业于哈尔滨工业大学. 现为中国科学院信息工程研究所正研级高工,博导,主要研究方向为大数据安全与隐私保护等.
E-mail:shijinqiao@ iie. ac. cn

方滨兴 男,1960 年 7 月出生于黑龙江省哈尔滨市. 现为中国工程院院士、电子科技大学广东电子信息工程研究院教授,主要研究方向计算机系统结构、信息安全等。
E-mail:fangbx@ iie. ac. cn