DOI: 10.11992/tis.201712006
网络出版地址: http://kns.cnki.net/kcms/detail/23.1538.TP.20180410.1436.020.html

PG-RNN: 一种基于递归神经网络的密码猜测模型

滕南君[1,2], 鲁华祥[1,3,4], 金敏[1], 叶俊彬[1,2], 李志远[1,2]

（1. 中国科学院 半导体研究所, 北京 100083; 2. 中国科学院大学, 北京 100089; 3. 中国科学院 脑科学与智能技术卓越创新中心, 上海 200031; 4. 半导体神经网络智能感知与计算技术北京市重点实验室, 北京 100083）

摘　要: 用户名—密码 (口令) 是目前最流行的用户身份认证方式, 鉴于获取真实的大规模密码明文非常困难, 利用密码猜测技术来生成大规模密码集, 可以评估密码猜测算法效率、检测现有用户密码保护机制的缺陷等, 是研究密码安全性的主要方法。本文提出了一种基于递归神经网络的密码猜测概率模型 (password guessing RNN, PG-RNN), 区别于传统的基于人为设计规则的密码生成方法, 递归神经网络能够自动地学习到密码集本身的分布特征和字符规律。因此, 在泄露的真实用户密码集上训练后的递归神经网络, 能够生成非常接近训练集真实数据的密码, 避免了人为设定规则来破译密码的局限性。实验结果表明, PG-RNN 生成的密码在结构字符类型、密码长度分布上比 Markov 模型更好地接近原始训练数据的分布特征, 同时在真实密码匹配度上, 本文提出的 PG-RNN 模型比目前较好的基于生成对抗网络的 PassGAN 模型提高了 1.2%。

关键词: 密码生成; 深度学习; 递归神经网络; Markov; 密码猜测
中图分类号: TP391　　**文献标志码:** A　　**文章编号:** 1673−4785(2018)06−0889−08

中文引用格式: 滕南君, 鲁华祥, 金敏, 等. PG-RNN: 一种基于递归神经网络的密码猜测模型[J]. 智能系统学报, 2018, 13(6): 889–896.
英文引用格式: TENG Nanjun, LU Huaxiang, JIN Min, et al. PG-RNN: a password-guessing model based on recurrent neural networks[J]. CAAI transactions on intelligent systems, 2018, 13(6): 889–896.

PG-RNN: a password-guessing model based on recurrent neural networks

TENG Nanjun[1,2], LU Huaxiang[1,3,4], JIN Min[1], YE Junbin[1,2], LI Zhiyuan[1,2]

(1. Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China; 2. University of Chinese Academy of Sciences, Beijing 100089, China; 3. Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai 200031, China; 4. Semiconductor Neural Network Intelligent Perception and Computing Technology Beijing Key Lab, Beijing 100083, China)

Abstract: Passwords are the most popular way of user ID authentication. However, it is rather difficult to obtain large-scale real text passwords. Generating large-scale password sets based on password-guessing techniques is a principal method to research password security, which can be applied to evaluate the efficiency of password-guessing algorithm and detect the defects of existing user-password protective mechanisms. In this paper, we propose a password guessing-based recurrent neural network (PG-RNN) model. Our model can directly and automatically infer the distribution characteristics and character rules from the data of password sets, which is different from the traditional password generating method based on manual design rule. Therefore, an RNN model that has been trained on a disclosed real user password set can generate passwords very close to the real data of the training set, which avoids the limitations of manual setting for password guessing. The results of our experiments show that PG-RNN can generate passwords closer to primitive data distribution more than Markov in password length and character structure categories. When evaluating on large password dataset, the proposed PG-RNN model matching outperforms that of PassGAN, which is based on generative adversarial networks, by more than 1.2%.

Keywords: password generation; deep learning; recurrent neural networks; Markov; password guessing

收稿日期: 2017−12−05.　**网络出版日期:** 2018−04−10.
基金项目: 北京市科技计划课题 (Z171100002217094); 中科院战略性先导科技专项 (A 类)(XDA18040400).
通信作者: 金敏. E-mail: jinmin08@semi.ac.cn.

在网络时代普及的今天, 密码是一种被广泛使用的用户验证方法。主要原因在于, 一方面密码方便理解、使用, 另一方面较容易实现。然而,

让人担忧的是，密码的使用者总是倾向于设置一些强度低、易猜测的弱密码，例如：abcdefg，1234567 等。实际上，密码的安全性和方便性之间，总是存在某种程度上的折中：即强密码不容易被攻击破解，但是对于用户来说，很难记忆；而弱密码虽然方便记忆和使用，但却容易被猜到。现阶段大部分网站在用户设定密码时，都会加入密码强度测试机制 (一般分为“弱、中等、强”3 个级别) 这样的预防措施能够在一定程度上提醒用户避免设定过于简单的密码。这些机制通常都是基于规则的，比如：要求密码必须包含一个数字、一个小写字母或者一个特殊字符[1]，密码长度在 6~18 位之间等。

如何更快、更有效地找到有效的用户密码，一直以来都是一个活跃的研究领域。目前流行的基于规则的密码猜测工具 hash-cat, John the ripper (JTR)[2-3]，主要通过原有的密码字典或泄露的密码数据集，加上密码规则的模糊化和变形来生成新的大量近似的密码。文献[4]开发了一种基于模板结构的密码模型 PCFGs，采用了上下文无关法，这种方法背后的思想是将密码切分成不同的模板结构 (e.g., 5 个小写字母加 3 个数字)，让终端产生的密码符合这样的密码结构。每个生成的密码 P 概率等于该密码结构类型的概率 P_T 与各子结构的概率乘积，例如，如果一个密码由两部分组成：字母+数字，那么该密码的生成概率则为 $P=P_{\text{letter}}P_{\text{digit}}P_T$，值得一提的是，PCFGs 模型在针对长密码时有着较好的效果。文献[1]采用一种基于马尔可夫的模型，该模型通过评估 n 元概率的原理，在衡量密码强度上性能要优于基于规则的方法。文献[6]系统地比较和实现了目前流行的几种密码猜测的技术来评估密码强度，发现字典攻击在发现弱密码时最有效，它们能够快速地以哈希校验的方法快速检验大量规则相似的密码，而马尔可夫链模型则在强密码时表现更加突出。所有的这些攻击方法随着搜索空间的不断扩大，有效性会出现指数型的下降[7]。

尽管上述的这些方法，都能够在一定程度上弥补人为设定密码规则的一些不足，但是这些方法往往也包含大量非真实用户设置密码[5]；此外，密码规则的确立和启发式探索依然需要大量密码专家的参与。对于人为设定的密码，在一定程度上，可以将其看成语言的延伸，因此，明文密码的设置习惯依然符合人类的表达习惯；在本文中我们希望能够直接、有效地挖掘出密码的一些内在的规律或特征。文献[8-9]中，展示了递归神经网络能够很好地学习到文本数据特征，并且生成一些之前从未出现过的新字符组合。这表明，递归神经网络并不仅仅只是简单的复刻、重现训练数据，而是通过内部的特征表示不同的训练样本，在高维度中综合重构出新的数据。我们的 PG-RNN 模型很大程度上是基于之前的这些方法，旨在通过小规模泄露密码样本数据，生成更多符合真实用户密码样本分布空间特征的密码，提高密码猜测算法效率；同时，通过端到端的小模型生成方式，能够有效地扩充密码攻击字典，缩小密码猜测空间。

预测是一个概率问题，对于一个训练好的 RNN 网络，给定一串输入字符序列，然后计算出下一个字符的概率分布并且根据概率生成下一个出现的字符，并将当前时刻的字符作为下一步网络的输入。由于密码本身就是一串字符串，因此，密码的生成和文本生成之间有着非常相似的特点。最早尝试使用递归神经网络来做密码猜测攻击的是一篇博客[10]，它的想法是通过一大堆已经被破解的密码，产生新的、有效的密码，来预测那些还没有被破解的密码。但是遗憾的是，作者只是简单地搭建了个 RNN 模型，并没有对模型进行调整和修改，每个模型只生成了很少的密码数量，而且匹配上的密码数量也非常有限，以至于作者对这种方法可行性表示怀疑。最近，文献[11]第 1 次尝试了使用生成对抗网络[12](generative adversarial networks, GAN) 来进行密码猜测攻击。在生成对抗网络 PassGAN 中，生成网络 G 和对抗网络 D 采用的都是卷积神经网络，生成网络 G 接受输入作为噪声向量，前向传播经过卷积层后输出一个长度为 10 的 one-hot 编码的字符序列。这些字符序列经过 Softmax 非线性函数之后，进入对抗网络 D 中进行判别。在测试中，文献[11]通过两个网站公开泄露的密码数据集来训练 PassGAN 模型，然后生成不同数量级别的密码数量，结果显示他们的模型能够在测试密码数据中匹配上一定数量的密码。Melicher 等[13]提出了一种快速的密码猜测方法，他们采用了复杂的 3 层长短时记忆 (long-short term memory, LSTM) 递归层和两层全连接层的网络来产生新的密码字符序列。在测评中，文献[13]基于蒙特卡罗仿真的方法：在一个非常大的数量范围内 (10^{10}~10^{25})，对模型在 5 组密码长度、字符类型都不同的测试数据上进行测试，结果表明他们的方法性能要优于基于字典和规则的 Hash-cat 与 JTR，以及基于概率的 PCFGs、Markov 模型。

1　递归神经网络和 PG-RNN 模型参数设置

1.1　递归神经网络

递归神经网络 (RNN) 是一种基于时间序列的网络结构, 因而能够对具有时间顺序特性的数据进行建模。对于字符级别的 RNN 网络, 在每个时间步上, 输入值为 one-hot 编码的一维向量 (其中, 向量维度由数据集包含的字符种类数决定), 输入数据信息传递到隐层, 并更新隐层状态, 经过非线性函数后最后达到输出层, 输出一个预测概率分布, 并通过概率分布的值, 确定输出字符的种类。RNN 网络可以具有多层隐含层, 并且每一层包含若干个神经元, 加上非线性激活函数; 因而整个网络具有非常强大的特征表达能力。在连续多个时间步上, RNN 网络能够组合、记录大量的信息, 从而能够用来进行准确地预测工作。对于某一个特定时间步 T 的输出, 它不仅仅依赖于当前的输入值, 还与 T 之前的若干步输入有关。举个例子, 一个 RNN 网络要输出“Beijing”这个字符串, 我们可能会给该网络输入 Beijin, 而对于接下来网络要输出的这个字符, 根据输出概率分布, 输出字符“g”的概率要显著高于其他候选字符。另外, 在输出一串字符之后, 我们依赖一个特殊的换行符作为单个密码结束的标志。

RNN 网络的整个计算流程如下, 给定输入向量序列 $\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_T)$, 其中, $\boldsymbol{x}_i$ 代表的是 i 时刻的输入向量; 通过输入−隐层之间的权重矩阵传入网络隐层, 加上从上一个时刻隐层传入的状态信息, 经过非线性函数后计算出隐层向量序列 $\boldsymbol{H}=(\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_T)$ 和输出序列 $\boldsymbol{Y}=(y_1,y_2,y_3,\cdots,y_T)$。具体迭代过程如下:

$$\begin{aligned}&\text{for } i=1 \text{ to } T:\\ &\quad \boldsymbol{h}_i=\tanh(\boldsymbol{W}_{hx}\boldsymbol{x}_i+\boldsymbol{W}_{hh}\boldsymbol{h}_{i-1}+\boldsymbol{b}_h)\\ &\quad \boldsymbol{y}_i=\boldsymbol{W}_{yh}\boldsymbol{h}_i+\boldsymbol{b}_y\end{aligned}\tag{1}$$

式中: $\boldsymbol{W}$ 表示的是权重矩阵, 大小与连接的神经元个数有关; $\boldsymbol{W}_{hx}$ 表示隐层与输入层之间的权重矩阵; $\boldsymbol{W}_{hh}$ 表示隐层与隐层之间权重矩阵; $\boldsymbol{W}_{yh}$ 则表示的是隐层与输出层之间的权重矩阵; $\boldsymbol{b}_h$、$\boldsymbol{b}_y$ 分别表示的是隐层和输出层的偏置矩阵; tanh 是隐层输出值必须经过的非线性函数。

递归神经网络的误差通过反向梯度传播算法按照时间步从后往前传递。但是, 由于梯度在传递过程中需要经过连续地相乘, 因此这样的参数关系使得 RNN 的梯度传播会存在一定的难度。Bengio 等[16-17]证明了梯度在反向传播中, 会随着时间步的推移呈指数级的衰减或者爆炸问题, 给递归神经网络的训练增加难度。梯度爆炸会带来 RNN 网络训练的不稳定性, 在实际训练中, 梯度爆炸的情况可以通过对梯度进行裁剪 (将梯度限制在一定数值范围内) 来有效地控制。后来出现的 long short term memory (LSTM)[14], GRU[15]则是解决了 RNN 梯度衰减问题。通过改变神经元内部的结构方式, 并且加入中间信息的存储单元, 使得梯度可以在很长的时间步上传播, 输出与输入之间依赖的时间跨度变大。对于密码猜测任务来说, 单个密码的长度是有限的 (绝大部分 $\leqslant 15$)。因此, 长时间序列上的可依赖性或许并不是我们所需要的, 因为对于一个长度有限的密码来说, 当前字符可能仅仅取决于之前的几个字符, 而不是很多个。出于这样的考虑, 本文中的 PG-RNN 模型采用的是之前没有人尝试过的 RNN 网络结构, 从而能够搭建一个轻量化但非常有效地密码猜测模型 (整个网络模型参数约 0.12 M)。

1.2　PG-RNN 模型参数设置

本文提出的 PG-RNN 模型, 参数设置如下: 出于对训练数据中绝大部分的密码长度的考虑, 时间序列长度为 20; 模型采用单层递归神经网络, 隐层神经元数量为 256, 两个全连接层; 学习率初始化为 0.01, 采用了 Adagrad 梯度更新算法。

2　密码数据集分析

本文采用的是从公开互联网上收集到的一些网站泄露的真实密码数据集合, 这些公开的密码集合都是以纯文本 txt 或者 sql 格式存在。我们仅仅使用这些数据集中的密码部分, 而滤除掉其他非相关信息 (包括用户注册邮箱或者用户名等)。我们在实验中使用了如下的密码数据集, 它们分别是 Rockyou、Yahoo、CSDN、RenRen 和 Myspace[18-20]。Rockyou 密码集包含了 2009 年 12 月由于 SQL 漏洞遭到了黑客攻击, 导致约 3 200 万用户密码, 我们收集到大约 1 400 万无重复的密码; 2012 年, Yahoo 公司的 Voices 泄露了大约 40 万个账号信息, CSDN(Chinese software developer network) 是目前国内最大的 IT 开发者社区, 它在 2011 年发生的数据库泄露事件, 有大约 600 万用户账号和明文密码被公开。同样是在 2011 年, 国内著名的社交平台人人网也被曝遭到黑客攻击, 将近 500 万用户账号和密码泄露. 此外, 还有 Myspace 网站泄露的部分数据, 大约 37 000 个存在于 txt 的明文密码。

我们对这些数据进行了以下清洗工作。1) 剔除掉了除密码之外的其他信息; 2) 考虑到编码问

题, 只保留了那些只包含 95 个可打印 ASCII 字符的密码 (出于用户使用习惯考虑), 这一步滤除掉了少量的密码; 3) 我们对这些密码进行了长度的统计分析, 如图 1 所示。对于以上提到的密码数据集, 我们发现任何一个密码数据集来说, 大部分的密码长度都集中在[5, 15]的范围内 (对于本文中采用到的密码数据集来说, 密码长度分布在[5, 15]区段内的数量都占据了总数的 95% 以上)。这是因为一方面大部分网站在要求用户输入密码时, 都有最短长度限制, 另一方面, 对于大多数用户在设定密码时, 为了方便自己记忆和输入, 也不会选择长的密码。因此我们进一步只选取了长度 (不包括换行符) 在[5, 15]的密码作为我们的实验数据。最终的密码集细节情况如表 1 所示。

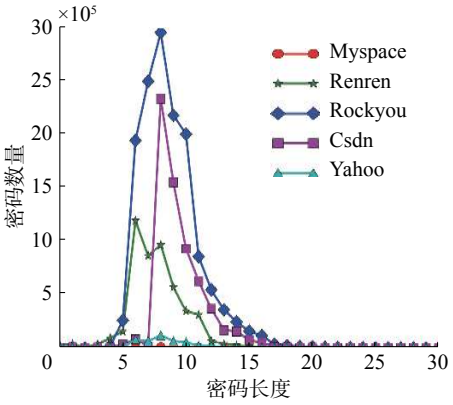


图 1　5 个公开泄露的密码数据集的密码长度分布情况

Fig. 1　The password length distribution of the five leaked passwords dataset

表 1　密码数据集的统计以及数据清理情况

Table 1　Statistic of password datasets and data clean

密码集	原始密码数	密码过滤	非 ASCII 数量	长度在[5, 15]	全部被移除的百分比/%
CSDN	6 428 632	6 427 077	1 555	6 349 908	1.20
RenRen	4 768 599	4 766 815	1 784	4 549 974	4.55
Rockyou	14 344 297	14 259 461	84 836	14 006 368	1.77
Yahoo	453 492	453 346	146	438 212	3.34
Myspace	37 144	36 874	270	36 215	1.24

3　实验及结果分析

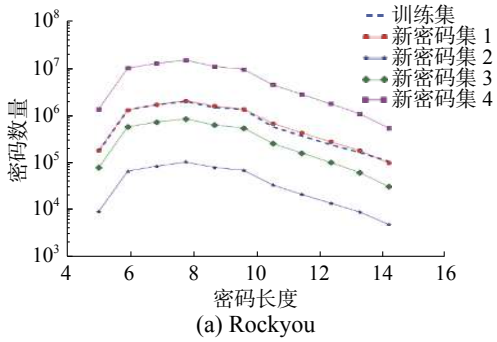
3.1　PG-RNN 的训练与数据切分

为评估 PG-RNN 模型效果, 我们对密码数据集进行了随机切分: 70% 密码用于训练, 30% 用于测试。以 Rockyou 密码数据集为例, 70% 的训练数据 (一共有 9 804 818 个无重复密码), 30% 的测试数据 (一共 4 201 550 个无重复密码), 对于其他数据集, 我们也做了同样的处理。

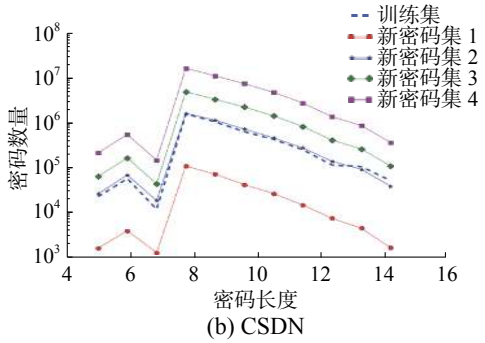
神经网络通过在训练过程中不断地迭代, 逐步学习到数据特征。考虑到我们收集到的数据集之间大小差异巨大, 而数据集的大小对于网络的训练次数是有着至关重要的影响。在实际训练过程中, 发现 PG-RNN 网络迭代到约 1.5 个 Epoch 之后, 误差就不再下降了, 网络性能也达到了相对稳定阶段。因此根据每个数据集的大小, 我们选择设置不同的迭代次数。

3.2　新生成密码长度分布和字符结构评估

密码长度和密码字符结构类型一直是衡量密码特性的重要指标。在该小节中, 我们参考了文献[5]的方法。对 Rockyou、CSDN、RenRen、Yahoo、Myspace 原始数据集以及各自新生成的密码数据集从密码长度和密码字符结构类型进行了统计分析。图 2(a) ~ (e) 分别表示 Rockyou、CSDN、RenRen、Yahoo、Myspace 的训练数据集和生成的不同量级的新密码集合在不同密码长度 (5~15) 上的数量分布情况。



(a) Rockyou



(b) CSDN

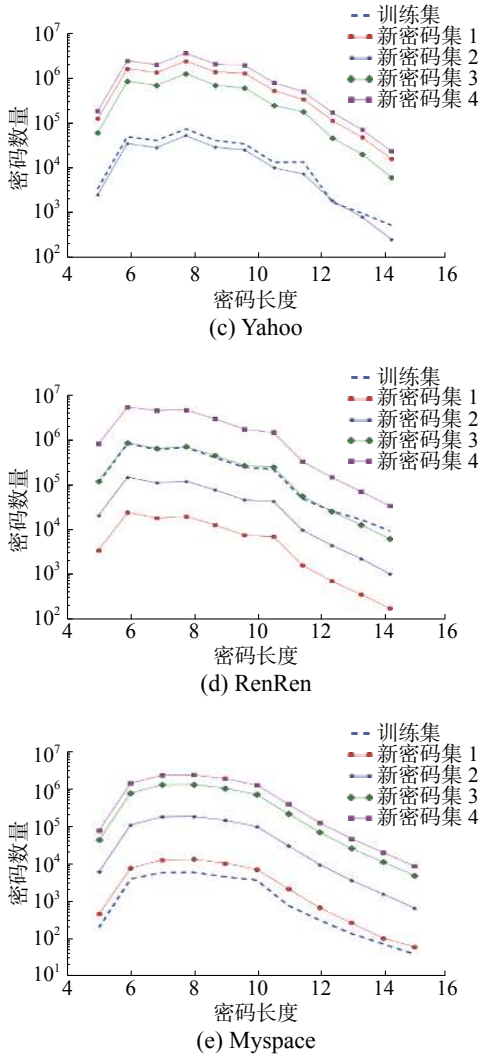


图 2　新生成的不同规模密码集的长度分布情况

Fig. 2　Length distribution of new password dataset with multiple scales

可以明显地看出, 通过我们的 PG-RNN 模型生成的新密码数据, 在长度分布上非常接近原始的训练数据, 当生成数量与原始训练集相当时, 二者几乎达到了重合的程度。对比 PG-RNN 与其他方法在 CSDN 密码集上的表现 (生成规模约为原始密码 1 倍), 原始数据集中数目最多的是长度为 8 的密码, 比例为 36.37%, PG_RNN 长度为 8 密码比例为 36.86%; 文献[5]中列出的方法在长度最多的密码数量上出现了不同程度的偏差, 其中 PCFG 和 4 阶 Markov, 分别达到了 6.2% 和 18.9%。

长度分布的衡量通常并不能很好地体现出密码之间的差异性。文献[2]中通过将密码切分为不同的模板, 反映出即使长度相同的密码, 也可能是由完全不同的字符类型组成。考虑此, 按照如下的几种字符结构类型对原始训练数据集和新生成的密码集 (~x1, ~x10), 进行了分类, 包括纯数字、纯字母 (大小写)、数字+字母 (大小写)、特殊字符共 4 类, 具体统计结果见表 1。对于 CSDN、RenRen 来说, 密码训练集都是以“纯数字”和“数字+字母”的形式为主, 比例分别占了各自对的 45.4% 和 38.8%、52.4% 和 25%; 而在 Rockyou 和 Yahoo 密码数据集中, “数字+字母”和“纯字母”占的比重最大, 这也反映了国内外用户在密码设置习惯上的一些差异。从表 1 中, 可以很容易看出, 无论是大约 1 倍的规模, 还是约 10 倍的规模, 我们的 PG-RNN 模型生成的新密码数据与原始的训练密码集的字符类型结构分布比例都非常地接近, 即便是对于占比重非常小的包含特殊字符的类型。

3.3　在训练集和测试集上的匹配度评估

参照文献[11]中的对比方法, 在这一小节中, 我们对 PG-RNN 模型生成的新密码数据进行了匹配度的评估, 也就是新生成密码与训练集和测试集的密码重合个数。重点对比了我们的方法与文献[11]中的 PassGAN 模型在 Rockyou 数据集上的效果; 同时针对其他几个数据集, 我们也给出了 PG-RNN 在测试集上的匹配度结果以及分析如表 2 所示。

表 2　CSDN 原始密码集和不同方法生成的新密码集 (x1 规模) 在密码数最多的长度 (L=8) 上的比较

Table 2　Comparison on CSDN primitive dataset and new datasets(x1 scale) generated by different methods on length (L=8) with the most passwords

模型	占密码总数百分比/%
原始密码集	36.37
PG-RNN	36.86
PCFG[5]	42.59
一阶 Markov[5]	11.65
三阶 Markov[5]	12.64
四阶 Markov[5]	17.46

密码生成工具都是通过学习现有数据集中的数据特征来产生新的密码数据集, 而新密码数据集与训练集的匹配度也能够反映出模型的学习能力。因此, 有必要将新生成的密码数据集与训练数据进行对比分析。文中重点对比了 PG-RNN 模型与文献[11]在 Rockyou 密码数据集上的表现, 具体结果如表 3 所示。从表格中可以直观地看出, 随着生成密码数量的增加, 新生成密码能够与训练集匹配上的密码个数也在增加, 这在 PG-RNN 和 PassGAN 两个模型上都能够很好地得到体现, 这也说明了 PG-RNN 模型和 PassGAN 都有着非常强的学习数据特征的能力。在匹配度上,

随着生成密码数量的增加，本文提出的 PG-RNN 方法与 PassGAN 相比，在匹配度的优势愈发明显；当生成密码数量在 10^8 时，PG-RNN 模型达到了 x2.24 倍以上的匹配度 (由于数据切分比例不同，我们的训练集包含的密码个数为 9 804 818 (无重复)，文献[11]为 9 926 278(无重复))。由于与训练集匹配上的密码是已知的，因此这部分密码完全可以通过基于训练集的字典攻击方式而得到。但是，与训练集的匹配度可以较好地表现出模型的学习能力的。值得一提的是，我们的 RNN 模型生成密码的重复率要远远小于 PassGAN，而在 PassGAN 模型生成的密码中，密码的重复率非常的高，达到 80% 以上 (随着生成密码数增加，甚至大于 90%)，事实上大量输出重复的密码并没有多大意义，反而会增加密码生成的时间。

表 3　不同训练密码集和规模约为训练集 1 倍、10 倍的新生成密码集在密码字符结构类型的统计情况

Table 3　Character structure categories on training dataset and new generated passwords at the scale of x1, x10 respectively on different datasets

网站名称	数据集	纯数字	纯字母	数字+字母	特殊字符
CSDN	训练集密码库	0.454	0.124	0.388	0.035
	PG-RNN-x1	0.445	0.117	0.396	0.041
	PG-RNN-x10	0.445	0.113	0.403	0.039
Rockyou	训练集密码库	0.166	0.288	0.484	0.062
	PG-RNN-x1	0.156	0.285	0.491	0.068
	PG-RNN-x10	0.168	0.285	0.473	0.074
RenRen	训练集密码库	0.524	0.206	0.25	0.02
	PG-RNN-x1	0.515	0.203	0.259	0.023
	PG-RNN-x10	0.516	0.209	0.25	0.025
Yahoo	训练集密码库	0.057	0.346	0.57	0.028
	PG-RNN-x1	0.052	0.341	0.572	0.036
	PG-RNN-x10	0.051	0.342	0.571	0.035
Myspace	训练集密码库	0.006	0.061	0.828	0.104
	PG-RNN-x1	0.013	0.061	0.817	0.11
	PG-RNN-x10	0.013	0.062	0.816	0.109

3.4　在测试集上的评估效果

此外，对 PG-RNN 和 PassGAN 两个模型生成的新密码，在测试集上进行了对比测试。详细的对比结果如表 4 所示。

表 4　PG-RNN 与 PassGAN 各自的生成密码在 Rockyou 训练集上的评估结果

Table 4　The evaluation results between PG-RNN and PassGAN on Rockyou training dataset

生成模型	总生成密码数	去重后密码数	重复率/%	在训练集中匹配上的密码数	匹配度/%
PG-RNN	1 000 000	996 808	0.319	31 530	0.327
	10 000 000	9 814 651	1.853	246 170	2.517
	100 000 000	92 363 373	7.637	1 081 612	11.027
Pass-GAN	1 000 000	182 036	81.796	27 320	0.28
	10 000 000	1 357 874	86.421	134 647	1.36
	100 000 000	10 969 748	89.030	487 878	4.92

其中，第 3 列和第 5 列分别表示的是在测试集中匹配上但是没有出现在训练集中的密码个数 (有重复)、在测试集中但不在训练集中的密码个数 (无重复)。对比两个模型的前两行数据，可以看出，我们的模型在生成的密码数多于 PassGAN 的情况下，能够在测试集上匹配上的比例大于后者，这是理所当然的 (由于切分比例不同，我们的测试集包含无重复密码个数是 4 201 550，

文献[11]中是 3 094 199, 计算比例时是相对于各自的测试集密码数而言, 这样比较起来相对公平)。对比两个模型的第 3、4 行数据结果, PG-RNN 在生成密码数量上与 PassGAN相同的情况下, 依然能够获得比 PassGAN 大的在测试集的覆盖率, 甚至超过了 1.2%, 这进一步说明本文提出的 PG-RNN 模型是非常具有竞争力的。需要指出的是, PassGAN 使用了复杂的多层残差卷积神经网络, 在网络模型复杂度和训练难度上都要远远高于 PG-RNN 模型。

除了 Rockyou 数据集, 我们也在表 5 ~ 6 中, 列出了我们的模型在其他数据集上的测试结果。从表格中看出, 我们的 PG-RNN 模型针对不同的数据集都有较好的效果。此外, 可以预见的是随着生成数量的进一步增大, 能够匹配上的数目会进一步增加。而针对 PG-RNN 模型在 Myspace 数据集上的表现相对于在其他数据集表现较差的原因, 我们分析主要在于神经网络的训练是高度依赖于数据的, 因而对于数据量较多的情况能够学习到更多的特征, 而我们收集到的 Myspace 数据集太小, 因此数据集体现的统计特征并不明显, 如表 6。但是, 总的来说, 基于递归神经网络的模型相对于人为设定规则和 Markov 等方法, 具备更强的发掘密码特征能力。

表 5　对比 PG-RNN 和 PassGAN 的生成密码在 Rockyou 测试集上的评估结果

Table 5　The evaluation results between PG-RNN and PassGAN on Rockyou test dataset

生成模型	生成的密码总数 (无重复)	在测试集中不在训练集中的密码数量 (有重复)	在测试集中的比例/%	在测试集中不在训练集中的密码数量 (无重复)	在测试集中的无重复比例/%
PG-RNN	515 431	7 065	0.168	6 943	0.165
	4 144 913	59 228	1.410	51 850	1.234
	10 969 748	140 835	3.352	118 551	2.822
	80 245 649	1 034 121	24.613	396 314	9.433
Pass-GAN	182 036	2 039	0.10	1 850	0.094
	1 357 874	12 489	0.60	11 398	0.576
	10 969 748	58 682	2.88	54 325	2.746
	80 245 649	172 997	8.51	162 652	8.221

表 6　PG-RNN 在其他密码数据集上的评估结果

Table 6　The evaluation results of PG-RNN on other datasets

数据集名称	生成的总密码数 (无重复)	在测试集中不在训练集中的密码 (有重复)	在测试集上匹配度/%	在测试集中不在训练集中的密码 (无重复)	在测试集上无重复匹配度/%
CSDN	36 913 159	201 304	15.17	48 705	3.67
RenRen	16 685 460	271 287	28.44	82 202	8.62
Yahoo	15 294 620	11 824	10.57	3 975	3.55
Myspace	762 163	321	2.93	189	1.72

4　结束语

本文提出了一种基于递归神经网络的密码猜测模型。在网上公开的泄露密码数据集 (包括 Rockyou、CSDN、RenRen、Yahoo、Myspace) 上对模型进行了一系列的训练、测试; 实验结果表明, 当在泄露密码数据上训练后, 针对字符级别建模的递归神经网络模型提供了一种端到端的密码生成解决方法, 能够很好地用来生成大量密码, 从而方便破译出更多潜在密码。

我们的模型针对不同的数据集, 以及生成不同密码规模情况下, 都能够较好地在密码结构字符类型, 密码长度分布等特征上接近原始训练数据的。此外, 在 Rockyou 数据集上, 我们的 PG-RNN 模型在生成数据规模相当的情况下, 在测试集中匹配超过 11% 的密码个数, 相对于 PassGAN 模型, 超过了 1.2%。我们的下一步工作主要分为以下两个方面: 1) 尝试其他的 RNN 网络结构, 并分析其在不同的结构在密码猜测上的效果; 2) 进一步观察 RNN 模型在生成密码时的内部数据表示状态。

参考文献：

[1] CASTELLUCCIA C, DÖRMUTH M, PERITO D, et al. Adaptive password-strength meters from markov models[C]//Proceedings of the 19th Network & Distributed System Security Symposium. San Diego, United States, 2012.

[2] HASHCAT[EB/OL]. [2017-10-12]. https://hashcat.net.

[3] John the Ripper password cracker[EB/OL]. [2017-10-15]. http://www.openwall.com/john/.

[4] WEIR M, AGGARWAL S, DE MEDEIROS B, et al. Password cracking using probabilistic context-free grammars [C]//Proceedings of the 30th IEEE Symposium on Security and Privacy. Berkeley, USA, 2009: 391–405.

[5] 韩伟力, 袁琅, 李思斯, 等. 一种基于样本的模拟口令集生成算法[J]. 计算机学报, 2017, 40(5): 1151–1167.
HAN Weili, YUAN Lang, LI Sisi, et al. An efficient algorithm to generate password sets based on samples[J]. Chinese journal of computers, 2017, 40(5): 1151–1167.

[6] MA J, YANG Weining, LUO Min, et al. A study of probabilistic password models[C]//Proceedings of 2014 IEEE Symposium on Security and Privacy. San Jose, USA, 2014: 689–704.

[7] AMICO M D, MICHIARDI P, ROUDIER Y, et al. Password strength: an empirical analysis[C]//Proceedings of 2010 IEEE INFOCOM. San Diego, USA, 2010: 1–9.

[8] GRAVES A. Generating sequences with recurrent neural networks[J]. Computer science, arXiv: 1308. 0850, 2013.

[9] SUTSKEVER I, MARTENS J, HINTON G E, et al. Generating text with recurrent neural networks[C]//Proceedings of the 28th International Conference on Machine Learning. Bellevue, USA, 2011: 1017–1024.

[10] Using neural networks for password cracking[OL/EB]. [2017-10-15]. https://0day.work/using-neural-networks-for-password-cracking/.

[11] HITAJ B, GASTI P, ATENIESE G, et al. PassGAN: a deep learning approach for password guessing[J]. arXiv: 1709.00440, 2017.

[12] GOODFELLOW I J, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets[C]//Proceedings of the 27th International Conference on Neural Information Processing Systems. Montreal, Canada, 2014: 2672–2680.

[13] MELICHER W, UR B, SEGRETI S M, et al. Fast, lean, and accurate: modeling password guessability using neural networks[C]//Proceedings of the 23rd USENIX Security Symposium. Austin, USA, 2016: 175–191.

[14] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735–1780.

[15] CHUNG J, GULCEHRE C, CHO K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv: 1412.3555, 2014.

[16] KOLEN J, KREMER S. Gradient flow in recurrent nets: the difficulty of learning LongTerm dependencies[M]. [S.l.]: Wiley-IEEE Press, 2001.

[17] BENGIO Y, SIMARD P, FRASCONI P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE transactions on neural networks, 1994, 5(2): 157–166.

[18] ROCKYOU[OL/EB]. [2017-10-13]. http://downloads.skullsecurity.org/passwords/rockyou.txt.bz2.

[19] YAHOO. Hackers expose 453, 000 credentials allegedly taken from Yahoo service (Updated)[EB/OL]. [2012-07-12]. http://arstechnica.com/security/2012/07/yahoo-service-hacked/.

[20] MYSPACE. Information of 427 million MySpace accounts leaked, selling as a package at the price of 2800 dollars in black market[EB/OL]. [2016-06-08]. https://www.wosign.com/english/News/myspace.html.

作者简介：

滕南君，男，1992 年生，硕士研究生，主要研究方向为数字信号处理、机器学习。

鲁华祥，男，1965 年生，研究员，博士生导师，主要研究方向为类神经计算芯片、类脑神经计算技术和应用系统、信息与信号处理。

金敏，女，1985 年生，助理研究员，主要研究方向为智能计算、模式识别与高性能计算。