基于深度学习的口令猜测方法的组合优化构造

郗志红[1,2], 周永彬[1,2], 李　勇[1], 樊一康[1], 谢子平[1], 石瑞鑫[1,2]

[1] 中国科学院信息工程研究所 北京 中国 100093
[2] 中国科学院大学网络空间安全学院 北京 中国 100049

摘要　目前的口令猜测方法主要分为两类, 分别基于统计方法和深度学习。与传统的基于统计的口令猜测方法相比, 基于深度学习的口令猜测方法在生成候选口令的数量及多样性方面均有显著技术优势。然而, 现有的基于深度学习的口令猜测方法通过逐字符或映射采样的方式生成候选口令, 未利用口令的内在结构特征, 通常需要生成大量的候选口令才能取得较理想的猜测效果, 在生成候选口令数较小的情况下, 猜测成功率较低。针对上述问题, 基于对口令结构与口令片段之间相互独立性的观察与认识, 对口令内在结构特征与猜测模型基础特性的具体分析, 本文提出一种以模块化方式对现有的基于深度学习的口令猜测方法进行组合优化的构造方法, 将合适的统计模型作为基础组件引入到口令猜测过程中, 弥补深度学习方法对口令结构特征学习方面的不足, 以期获得具有更高猜测成功率和更好猜测效率的新方法, 提高基于深度学习的口令猜测方法的实用性。实验结果表明, 与现有的基于深度学习的口令猜测方法相比, 经过组合优化构造出的新的口令猜测方法在同站和跨站口令猜测场景下的猜测成功率平均提高了 215.51%和 176.84%, 同时口令猜测模块之间的独立性有利于口令猜测过程的并行运行, 提高了口令猜测效率, 证明了本文组合优化设计方法的有效性。

关键词　口令猜测; 深度学习; 统计模型; 组合优化
中图法分类号　TP309.2　DOI 号 10.19363/J.cnki.cn10-1380/tn.2023.06.09

Combinatorial Optimization Construction of Password Guessing Method based on Deep Learning

XI Zhihong[1,2], ZHOU Yongbin[1,2], LI Yong[1], FAN Yikang[1], XIE Ziping[1], SHI Ruixin[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract　The current password guessing methods are mainly divided into two categories, which are based on statistical methods and deep learning. The existing password guessing methods based on deep learning have great advantages in the number and diversity of password guessing compared with the statistical password guessing methods. However, the existing password guessing methods based on deep learning generate candidate passwords in a character-by-character or map-sample manner. It is necessary to generate a large number of candidate passwords to get a better guess effect without using the internal structure characteristics of passwords. When the number of candidate passwords is small, the guessing success rate is low. Aiming at the above problems, based on the observation and understanding of the mutual independence between the password structure and the password fragments, this paper proposes a modular construction method to optimize the existing password guessing methods based on deep learning by analysising the characteristics of the password structure and the basic characteristics of the guessing model. In order to make up for the deficiency of the deep learning method in the learning and generation of password structure features, and to obtain a new method with higher guess success rate and better guess efficiency, some appropriate statistical models are introduced into the password guessing process as a basic component. Furthermore, it improves the practicability of password guessing method based on deep learning. The experimental results show that the password guessing success rate of the combined-optimized password guessing method is up to 215.51% and 176.84% higher than that of the existing password guessing methods based on deep learning in the same site and cross-site password guessing scenarios. At the same time, the independence between the password guess modules is conducive to the parallel operation of the password guess process and improves the efficiency of the password guess process. It shows the effectiveness of combinatorial optimization.

Key words　password guessing; deep learning; statistic model; combinatorial optimization

通信作者: 周永彬, 博士, 研究员, Email: zhouyongbin@iie.ac.cn。
本课题得到国家自然科学基金项目(No. 61632020, No. U1936209, No. 62002353)和北京市自然科学基金项目(No. 4192067)资助。
收稿日期: 2020-11-23; 修改日期: 2021-03-02; 定稿日期: 2023-02-16

1 引言

互联网的快速发展使得账户安全及隐私保护越来越需要身份认证技术保护, 虽然指纹认证、虹膜认证等生物识别技术以及智能卡、USB KEY 等基于硬件的身份认证技术不断地被研发和应用, 基于文本口令的身份认证技术由于具有易理解、易实现、易部署、使用成本低廉、可与其他认证技术混合使用等显著优点, 短时间内难以被完全替代, 仍然是互联网中应用最为广泛的认证技术[1–4]。然而, 在实际应用中, 用户通常选择便于记忆的口令, 导致大多数口令容易受到猜测攻击, 研究口令猜测方法, 并以此指导口令安全使用, 提高口令被猜测攻击的防护能力, 对账户安全及隐私保护具有重大意义。

在对口令进行初始设置或因故需要更换时, 为便于记忆与使用, 用户通常倾向于选择简短易记的字符串。例如, 使用个人相关信息(包括出生日期、出生地等)、常用词(如 password、com)、键盘邻近字符序列(如 qaz、asdfgh)等字符串来构造口令。通过对大规模口令集的统计分析, 用户口令和自然语言一样满足 Zipf 定律[5], 因此通过学习口令集中频繁出现的各种构造模式(如姓名+生日、出生地+生日)来生成符合真实用户口令创造习惯的候选口令是可行的[6–9], 合理利用上述用户构造口令的规律, 可以极大地降低口令猜测的搜索空间。这是导致多种口令猜测攻击得以奏效的主要技术根源之一。

随着自然语言处理领域深度学习技术的快速发展, 基于深度学习的口令猜测方法不断出现并成为口令猜测领域的研究热点。近年频繁发生的口令泄漏事件也为基于深度学习的口令猜测方法研究提供了客观条件, 亿级泄露口令数据可以帮助深度学习模型更加精确地刻画真实口令集的分布规律。按照使用的神经网络类型划分, 目前主要有基于循环神经网络[10]的口令猜测方法、基于生成对抗网络[11]的口令猜测方法和基于变分自编码器[12]的口令猜测方法。基于循环神经网络的口令猜测方法使用训练好的循环神经网络来逐字符生成候选口令。基于生成对抗网络的口令猜测方法利用生成网络和判别网络的对抗学习生成候选口令。基于变分自编码器的口令猜测方法使用编码器网络的编码映射和解码器网络的解码采样得到候选口令。此外, 作为对上述基于深度学习的口令猜测方法的改进, 研究人员近期提出利用测试集信息不断改进采样空间的动态口令猜测方法[13]。现有的基于深度学习的口令猜测方法较少地依赖专家知识, 可以生成数量多(10^{10} 以上)且多样性良好的候选口令。随着候选口令数量的逐渐增多, 基于深度学习的口令猜测方法可以逐渐逼近甚至超过基于统计的口令猜测方法, 成为目前口令猜测研究的热点方向[14]。

然而, 现有的基于深度学习的口令猜测方法在生成候选口令时采用逐字符生成或者映射采样的方式, 未能利用口令内在的结构特征, 在候选口令数量较少($\leq 10^8$)时猜测成功率不如基于统计的口令猜测方法[1], 一定程度上降低了它的应用效果。针对上述问题, 本文以口令片段的视角看待、分析口令结构特征和优化口令猜测过程, 在充分发挥基于深度学习的口令猜测模型在生成数量多、多样性好等优势基础上, 通过模块化设计方法, 引入适当的统计模型作为基础组件完成口令结构和部分类别的口令片段猜测, 以期获得更高的口令猜测成功率, 并给出了相应的实验验证结果。

本文组织结构如下: 第 1 节介绍研究背景和研究内容; 第 2 节介绍相关研究现状, 包括基于统计的口令猜测方法和基于深度学习的口令猜测方法; 第 3 节主要从口令结构方面进行分析, 详述深度学习口令猜测方法的组合优化的设计动机、模型结构和工作流程; 第 4 节与三种主要的基于深度学习模型的口令猜测方法在同站和跨站口令猜测两种场景下的口令猜测效果进行了对比; 最后, 第 5 节对本文进行总结和展望。

2 相关工作

2.1 基于统计的口令猜测方法

由于基于字典和规则的启发式方法[15-16]过于依赖人工经验, 研究人员以统计学作为理论基础, 提出基于统计的口令猜测方法, 包括基于 Markov [17-18]和基于概率上下文无关文法[19]两种猜测方法。

基于 Markov 的口令猜测方法利用口令字符之间的前后关联关系, 将口令视为一个由若干单字符构成的 Markov 链, 计算在给定输入时下一个字符的概率分布, 逐字符地生成候选口令。2005 年, Narayanan 等人首先利用口令和自然语言紧密相关的事实, 将统计自然语言处理中的 n-gram 模型应用到口令猜测中[17]。n-gram 模型基于 Markov 假设, 即口令中的字符只与它之前的 n-1 个字符相关(相当于 n-1 阶 Markov 模型), 通过将口令的字符序列建模成 Markov 链的形式来刻画口令的概率分布。例如, 4-gram 模型下的口令 pass123 的概率计算方式为

$$\Pr(\text{pass123}) = \Pr_{\text{l}}(\text{pas}) * \Pr_{\text{l}}(\text{s}\,|\,\text{pas}) * \Pr_{\text{l}}(1\,|\,\text{ass}) * \Pr_{\text{l}}(2\,|\,\text{ss1}) * \Pr_{\text{l}}(3\,|\,\text{s12})$$

其中, 概率 $\Pr_{\text{l}}$ 可通过对训练集的统计分析得到:

$$\Pr_{\text{l}}(a_4|a_1a_2a_3) = \frac{\text{Count}(a_1a_2a_3a_4)}{\text{Count}(a_1a_2a_3)}$$

概率 $\Pr_{\text{l}}(\text{pas})$ 可通过阶数更小的Markov模型来计算。

文献[17]中的口令猜测方法虽然可以生成候选口令, 但效率较低。2015 年, Dürmuth 等人改进了其中的口令枚举算法[18]。其核心思想是首先将概率值分为多个不相交的区间, 这些区间一般通过对概率取对数后进行适当的微调得到, 使其只能为非正整数。然后将每个区间表示的概率根据 Markov 模型的阶数分解成可能的区间向量, 最后通过遍历向量每个位置上可能的字符串来得到最终的候选口令。例如, 令口令字符空间为 $\Sigma = (a,b)$, Markov 模型阶数为 2, 要得到长度为 3 的候选口令, 假设经过对口令集的训练得到 $\Pr_{\text{l}}$ 对应的概率区间(用 L 表示)为

L(aa) = 0, L(ab) = -1
L(ba) = -1, L(bb) = 0
L(a|aa) = -1, L(b|aa) = -1
L(a|ab) = 0, L(b|ab) = -2
L(a|ba) = -1, L(b|ba) = -1
L(a|bb) = 0, L(b|bb) = -2

由 Markov 模型的概率计算公式可知, 通过 $\Pr_{\text{l}}$ 概率连续相乘得到口令概率, 由于区间由概率取对数得到, 口令概率区间可以拆分成 $\Pr_{\text{l}}$ 对应的概率区间的相加, 即一个口令概率区间对应一组区间向量。假设每个区间对应的向量长度为 2。首先考虑区间为 0 的情况, 0 只可能表示为两个 0 之和, 因此区间 0 只对应区间向量[0, 0],此时只有[L(bb), L(a|bb)]一种情况, 故候选口令为 bba。接着考虑区间-1, -1 可以表示为 0 和-1 的和, 因此区间-1 对应两个区间向量[0, -1]和[-1, 0], 前者有[L(aa), L(a|aa)]和[L(aa), L(b|aa)] 两种情况, 后者有[L(ab), L(a|ab)] 一种情况, 总共可得到三个候选口令 aaa、aab、aba。依此类推, 可以得到近似概率降序的候选口令集。

基于Markov的口令猜测方法关注字符之间的关联关系而非单纯的字符串概率, 具备一定程度的泛化性, 可以生成某些不在训练集中但符合训练集分布规律的内容。

上下文无关文法(Context-Free Grammar, CFG)是自然语言处理领域常用的工具之一, 概率上下文无关文法(Probabilistic Context-Free Grammar, PCFG)在 CFG 的基础上引入了概率, 目的是消除 CFG 在句法分析中产生的歧义[19]。2009 年, Weir 等人将概率上下文无关文法应用到了口令猜测中[20]。PCFG 包括四个部分: 起始符、终止符、非终止符、带有概率的产生式。为了构成完整的文法结构, Weir 将口令划分为字母段、数字段、特殊字符段, 这些不同长度的段作为非终止符, 段中的具体字符内容作为终止符, 起始符与口令结构、段与具体字符串之间的条件概率关系作为产生式, 构成一个 PCFG, 其对应的语言就是最终产生的候选口令集。例如, 假设训练集中的口令为 xzh123、xzh456、aaa123?、abcd12, 对应的概率上下文无关文法如表 1 所示。

表 1 概率上下文无关文法示例

Table 1 Probabilistic context-free grammar example

产生式	概率
△→L3D3	0.5
△→L3D3S1	0.25
△→L4D2	0.25
L3→xzh	0.67
L3→aaa	0.33
L4→abcd	1.0
D2→12	1.0
D3→123	0.67
D3→456	0.33
S1→?	1.0

其中△表示文法的起始符, L 表示字母段, D 表示数字段, S 表示特殊字符段, 段名后跟数字表示指明长度的段, 如 L3 表示长度为 3 的字母串。口令结构的概率与口令片段字符串概率的乘积即为口令概率, 如计算口令 aaa456 概率的方法为

$$\begin{aligned}&\Pr(\text{aaa456})\\&= \Pr(\Delta \to \text{L3D3})\Pr(\text{L3} \to \text{aaa})\Pr(\text{D3} \to 456)\\&= 0.5 \times 0.33 \times 0.33 = 0.05445\end{aligned}$$

为了生成候选口令, PCFG 首先统计分析训练集得到口令结构, 然后使用字典对口令结构进行填充得到以概率降序进行排列的候选口令。

与基于 Markov 的口令猜测方法相比, 基于 PCFG 的口令猜测方法关注口令结构与字符串之间的关联关系, 口令结构的数量一般远小于口令数量, 因此 PCFG 具备较高的训练速度和猜测效率。

基于统计的口令猜测方法的主要优势是在生成的猜测口令数量不大($\le 10^8$)时猜测成功率高于基于深度学习的口令猜测方法, 但模型参数的选择比较依赖于专家知识, 且由于统计方法的目标是准确地拟合训练数据, 其生成的猜测口令高度依赖于训练

集, 容易产生过拟合, 因此在生成更大规模候选口令的情况下猜测效果容易被基于深度学习的口令猜测方法超过。

2.2 基于深度学习的口令猜测方法

在自然语言处理领域, 已经产生了很多可以用来学习文本数据并生成新文本的深度学习模型[21–27]。口令也是一种文本, 且具有自然语言的某些特征[28–31], 近年来研究人员逐步深入研究深度学习模型特别是生成模型在口令猜测方面的应用, 基于深度学习的口令猜测方法逐渐崭露头角。

2016 年, Melicher 等人首次将长短期记忆网络(Long Short-Term Memory, LSTM)应用到口令猜测领域中[10]。循环神经网络(Recurrent Neural Network, RNN)是一种可以生成序列数据的神经网络, 其由隐藏层、输入层和输出层组成, 通过隐藏单元上的循环连接来保存过去的状态, 从而具备了一定长度的记忆。LSTM 是一种改进过的循环神经网络[32], 它在 RNN 的基础上增加了细胞状态来解决 RNN 中存在的梯度消失问题, 使得模型能生成更加符合训练集结构的序列数据。基于 LSTM 的口令猜测方法的核心思想与 Markov 模型类似, 区别在于 LSTM 使用神经网络计算下一位字符概率来逐字符生成候选口令, 即在输入空串时可以输出长度为 1 的候选口令及对应的概率, 再以长度为 1 的口令作为输入得到长度为 2 的候选口令及对应概率, 依次类推得到候选口令集。假设字符集合 $\Sigma=(a,b,c,d)$, 则模型原理如图 1 所示。

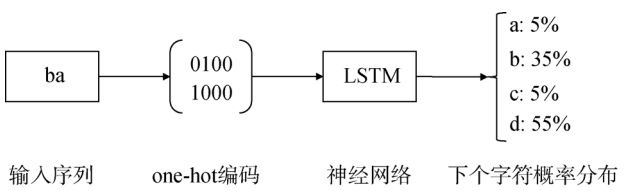

图 1 基于 LSTM 的口令猜测方法

Figure 1 Password guessing method based on LSTM

与基于 Markov 的口令猜测方法相比, 基于 LSTM 的口令猜测方法虽然核心原理类似, 但可以使用正则化或者裁剪网络连接的方式达到更强的泛化性, 因此在输出内容的丰富度方面具有明显优势, 这也是其被广泛应用于自然语言处理中文本生成的主要原因之一。

2017年, Hitaj等人利用生成对抗网络(Generative Adversarial Networks, GANs)的思想提出了一种新的口令猜测方法 PassGAN[11]。GANs 是一种以生成网络和判别网络之间的博弈作为训练过程的神经网路[33], 生成网络直接产生样本 $x=g(z;\theta^g)$, 而判别网络则通过 $d(x;\theta^d)$ 来指出 x 有多大概率是从真实的训练数据中抽取的而不是由生成网络生成的伪造数据。如果这个概率值为 0.5, 说明判别网络已经无法将生成网络的输出和真实数据区分开, 此时生成网络的输出即可作为真实数据的模拟。PassGAN 利用 GANs 来学习对真实口令集分布的近似, 通过生成候选口令的生成网络与鉴别候选口令的判别网络之间的对抗完成训练, 工作原理如图 2 所示。生成网络使用随机噪声生成伪造口令, 判别网络分析输入口令是来自生成网络生成的伪造口令还是取自真实口令集的口令,训练完成以后生成网络即可直接生成最终的候选口令。

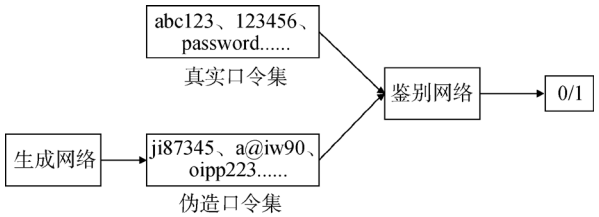

图 2 基于 GANs 的口令猜测方法

Figure 2 Password guessing method based on GANs

2020 年, Wang 等人通过使用变分自编码器(Variational Auto-Encoder, VAE)来实现候选口令的生成[12]。VAE 是一种将变分推断与深度学习进行结合的生成模型, 将输入通过编码器编码以后再通过解码器解码来生成序列, 通过减小预先选择的分布与真实后验概率分布之间的 KL 散度来完成训练, 从而可以生成符合输入数据真实分布的输出序列。基于 VAE 的口令猜测方法将真实口令集的分布作为变分自编码器的输入, 输入口令通过编码器映射到口令潜在空间, 训练完成以后, 解码器以高斯采样的方式对口令潜在空间进行解码, 输出最终的候选口令, 工作原理如图 3 所示。

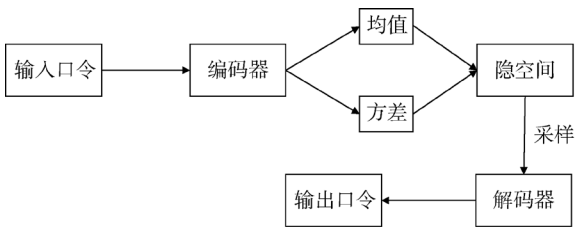

图 3 基于 VAE 的口令猜测方法

Figure 3 Password guessing method based on VAE

基于深度学习的口令猜测方法将真实口令作为输入, 通过 one-hot 等编码方式对输入口令进行处

理, 使用编码后的结果对神经网络进行训练以生成符合训练集分布特征的候选口令。如果训练集和测试集分布不同会导致协变量移位, 使得生成的候选口令无法达到理想的猜测成功率。针对上述问题, Pasquini 等人对基于 GAN 和 VAE 的口令猜测方法进行改进[13], 提出一种称为动态口令猜测(Dynamic Password Guessing, DPG)的方法, 该方法在口令猜测过程中利用被成功猜测的口令不断调整隐空间所在区域, 逐渐向测试集所在的隐空间偏移, 进而获得与测试集分布更相近的候选口令集合。

与基于统计的口令猜测方法相比, 基于深度学习的口令猜测方法在生成口令数量和多样性上有明显的优势, 随着候选口令数目的增多, 其猜测效果会逐渐逼近甚至超过基于统计的口令猜测方法。此外, 神经网络的训练过程是通过反向传播等算法自动进行的, 对专家知识的依赖相比基于统计的口令猜测方法也更少[10-12]。

然而, 由于输入数据实际上都被看作字符序列, LSTM、GANs、VAE 等深度学习模型难以充分利用口令的结构特征, 此外, 深度学习模型大都比较注重泛化性, 在训练时不会完全以拟合训练集数据作为优化目标。上述因素导致在候选口令数量较小($\leq 10^8$)时基于深度学习的口令猜测方法的猜测成功率与基于统计的口令猜测方法相比有较大差距[1,14]。

3 基于深度学习的口令猜测方法的组合优化构造

3.1 动机

与传统的基于统计的口令猜测方法[17-19]相比, 现有的基于深度学习的口令猜测方法[10-13]在生成候选口令的数量及多样性方面均有显著的技术优势, 但其未能充分挖掘和利用口令所具备的结构特征, 一般需要达到一定的猜测数量后(10^8 甚至更多)才能获得较为理想的猜测成功率, 从而降低了基于深度学习的口令猜测方法的实用性。本文以口令片段的视角看待和分析口令结构特征, 基于对口令内在结构特征与模型基础特性的具体分析, 提出以模块化方式对现有的基于深度学习的口令猜测方法进行组合优化, 通过将合适的统计模型作为基础组件引入到口令猜测过程中, 以期获得更高的猜测成功率和更好的猜测效率, 提高基于深度学习的口令猜测方法的实用性。

口令与普通单词主要区别在于口令具有结构特征。用户创造口令时往往使用易记易用的口令片段以一定的结构(比如“姓名+出生日期”“姓名首字母+地址名”等)组合生成口令。大部分用户口令可视为对组成某种口令结构的若干口令片段以相应的字符串填充后的结果,口令结构决定口令的基本框架, 其越接近用户创造口令的习惯或倾向, 猜测成功率就越高。现有的基于深度学习的口令猜测模型一般将训练集中一条口令作为一个基本的对象进行编码映射等处理操作, 在猜测时则以单个字符为粒度逐位生成候选口令(LSTM)或者直接以单个口令作为基本单元输出到候选口令集(PassGAN、VAE), 而不是以口令结构及口令片段的形式看待和分析口令数据, 未能对口令结构特征进行挖掘和利用。为了解决上述问题, 本文将口令结构从基于深度学习的口令猜测方法中剥离, 在单独生成的口令结构上进行候选口令的构建。生成候选口令的过程可以分解成两个方面, 一方面生成口令片段构成的候选口令结构, 另一方面生成不同类型、不同长度候选口令片段。

在候选口令结构生成方面, 一个口令结构代表一簇候选口令, 错误的口令结构容易显著降低口令猜测成功率。同时, 口令集中口令结构种类的数量远小于口令数量, 过强的泛化性会为口令结构的生成带来高代价(大量无效口令结构被生成)、低收益(较小的口令猜测成功率的提升)。因此, 口令结构的训练和生成以拟合训练集为主要目标, 不需要很强的泛化性。为了能尽可能生成更加符合用户口令创造习惯的口令结构和减少无效的口令结构数量, 本文选择使用强泛化性较弱的统计模型来实现口令结构的学习和生成。

在候选口令片段生成方面, 最终的候选口令数量及其内容在相当程度上取决于用于填充口令结构的各类口令片段的数量及内容。生成的口令片段越接近用户创造口令的习惯或倾向, 数量越多、多样性越好, 猜测成功率就越高。口令片段可以有多种划分方式, 既可以粗略地划分成字母段、数字段、特殊字符段, 也可以更加精细地将字母段划分成姓名段、地址段、常用字符段等。对于不同类别的口令片段, 可以灵活地根据自身性质采取不同的模型来训练和生成。以字母段为例, 由于口令中的字母串一般都是与自然语言紧密相关的, 因此自然语言处理中的一些模型如 n-gram、LSTM 等就可以很自然地应用到字母串的生成过程中。考虑到深度学习模型在自然语言处理领域的优势, 以及其在生成规模和多样性的长处, 可以用来生成数量足够且多样性良好的字母串集合, 进而最终获得大量高质量的候选口令。类似地, 口令中其他类型的口令片段也可以根据实际需

求和口令片段的性质来选取合适的生成模型。

综上所述, 为了克服或改善基于深度学习的口令猜测方法中存在的未利用口令结构特征导致达到理想的猜测成功率需要大量候选口令方面的问题, 本文提出一种模块化的方式对基于深度学习的口令猜测方法进行组合优化。例如, 为了利用口令的结构特征, 用模型A专门生成口令结构, 为了可以生成在数量和多样性均具备优势的候选口令集合, 生成口令片段1的内容用深度学习模型B, 为了在候选口令规模不大时可以达到更高的口令猜测成功率, 生成口令片段2的内容用基于统计的模型C等, 并使用口令空间的切分和各模型之间的可并行性来提高口令猜测效率, 将各个口令片段对应的模型生成的字符串按照口令结构进行填充, 以期获得具有更好猜测效果的候选口令集合, 如图 4 所示。

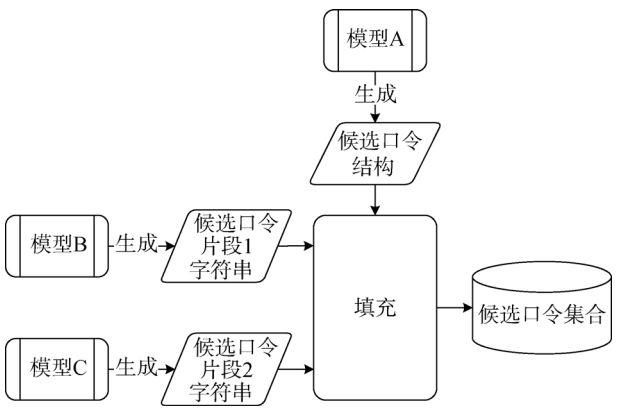

图 4 模块化口令猜测方法

Figure 4 Modular password guessing method

3.2 组合优化方法设计

口令结构和不同类型的口令片段可能具有不同的性质, 选择合适的模型作为基本构建组件时需以口令结构及口令片段的具体性质和猜测模型的技术特性为依据。具体地, 首先确定口令结构的划分标准, 在此基础上选择合适的用来生成候选口令结构和候选口令片段的模型, 最后确定生成候选口令的填充算法。

3.2.1 口令结构划分

为了能充分利用口令结构特征对基于深度学习的口令猜测方法进行优化, 需要结合口令的组成规律对口令结构进行划分。本文选择较为常用的口令表示方法, 直接将口令划分为字母段、数字段和特殊字符段。同时, 采用文献[19]中的记号标识方法, 用L表示字母段, D表示数字段, S表示特殊字符段, 后缀数字为片段长度。经过划分标记, 口令 xzh123 的结构可以表示为 L3D3, 口令 a12345?的结构可以表示为L1D5S1。

3.2.2 候选口令结构生成

对于候选口令结构的生成, 如 3.1 节所分析, 本文使用基于统计的模型。基于统计的模型包括PCFG、Markov 模型等, PCFG 生成的口令结构是直接通过对训练集中的口令结构进行统计得到的, 即其只能生成在训练集中出现过的口令结构, 有一定的局限性, 所以本文选择 Markov 模型作为口令结构猜测模型。具体地, 将划分后的口令片段(如 L1、L2、D1、S1 等)视为独立的单元, 口令结构视为若干口令片段单元组成的 Markov 链, 采用 Markov 模型对这些单元之间的概率关系进行建模。对于 Markov 模型的阶, 阶数为 1 可能会生成大量的无效结构, 如 L1、L2 等, 同时, 根据表 6 中对口令结构复杂程度的统计结果, 口令集中复杂口令结构所占比例较低, 过高的 Markov 阶数没有实际意义。因此本文采用二阶 Markov 模型。

3.2.3 候选口令片段生成

对于候选口令片段的生成, 需要结合不同类型口令片段的具体性质选择合适的模型。表 3～表 5 分别统计了三个口令集中最常出现的 10 个字母串、数字串和特殊字符串。

对于字母段, 根据表 3 的统计结果, love、password、princess、angel、ever、monkey、life、dragon、master、alex、shadow 为常见的英语单词, iloveyou、babygirl 为常见英文短句。除此之外, 其他字母串也有特殊的含义, 例如, qwe、qaz、asd、qwerty、wsx 表示键盘上的紧邻字符序列, abc 代表字母的自然顺序, www 是万维网(World Wide Web)的简称, com 是最常见的顶级域名。由此可见, 口令中的字母串通常与自然语言紧密相关, 适合使用自然语言处理模型来生成。为得到高质量的候选口令集合, 本文需要预设概率作为阈值, 以生成所有满足概率阈值的候选口令。GAN 和 VAE 通过从隐空间采样得到字符串, 难以生成满足某个概率阈值的所有字符串, 因此本文选择 LSTM 模型作为字母串的生成模型。与基于统计的自然语言处理模型相比, 优点在于可以提供在候选口令生成规模和多样性方面的优势。

此外, 针对 LSTM 模型生成速度慢的缺点, 本文进行了改进优化。原始的 LSTM 模型通过逐字符来生成字符串[10], 效率低时间长。本文借鉴 Markov 的思想通过限制字母之间的依赖长度来有效缩短生成时间。优化后的 LSTM 模型在生成某个位置的字母时, 只依赖于前面有限数量的字母, 依赖的字母数量即模型阶数。虽然阶数越高, LSTM 的学习效果

越好。但与高阶 Markov 一样, 更高的阶数同样会带来更高的时空复杂度[18]。大小写字母数量共有 52 个, 意味着模型每高一阶都会导致搜索空间扩大 52 倍, 过高的阶数会极大影响模型的运行速度和空间占用。为此, 本文将 LSTM 模型阶数设为 4, 可以在最大限度利用内存资源的情况下得到最佳的模型学习效果。除限制阶数外, 在对口令以概率降序搜索过程中, 为了进一步加快 LSTM 模型的生成速度, 本文利用 OMEN 中的有序枚举算法进行加速[18]。

对于数字段和特殊字符段, 常见的数字串和特殊字符串统计结果分别如表 4 和表 5 所示。可以看出, 口令集中的数字串和特殊字符串通常不具有语法或者语义意义, 但是具有比较明显的统计意义, 比如频繁出现的 123、12345、123456 等, 因此可以使用基于统计的模型来生成候选口令中的数字串和特殊字符串。考虑到 PCFG 的速度和实际效果都比较突出, 本文使用 PCFG 作为数字串和特殊字符串的生成模型。

3.2.4 候选口令生成

候选口令的生成使用遍历填充算法完成。一方面遍历生成的候选口令结构, 另一方面针对候选口令结构中不同类型以及长度的口令片段遍历相应的字符串集合, 并填充到候选口令结构中得到候选口令。

3.3 口令猜测过程

如 3.2 节所分析, 本文将口令猜测划分为四个模块, 分别生成候选口令结构和三种不同类型(字母、数字、特殊字符)的候选口令片段, 其中采用 Markov 模型来生成候选口令结构, 而字母段、数字段、特殊字符段的生成分别使用 LSTM、PCFG、PCFG。对生成的候选口令结构使用相应类型的候选口令片段字符串进行填充得到候选口令集合。本文将此 Markov+LSTM+PCFG+PCFG 的组合优化构造方法简称为 MLPP, 其包括以下四个阶段。

(1) 预处理阶段

对口令集中的原始口令进行预处理, 从每个口令中提取口令结构、字母串、数字串和特殊字符串四种类型的字符串, 例如, 对于口令 xzh123aaa, 可以得到 L3D3L3、xzh、aaa、123 四个字符串, 口令结构字符串单独进行存储, 口令片段字符串按照类型和长度分别进行存储。

(2) 训练阶段

对于存储的口令结构集合和不同类型的口令片段集合, 依照上节分析使用相应的模型进行训练。例如, 基于口令结构字符串 L3D3L3 等训练 2 阶 Markov 模型, 基于字母串 xzh、aaa 等训练 4 阶 LSTM 模型, 基于数字串 123 等训练 PCFG 模型。

(3) 生成阶段

模型训练完毕后, 生成相应类型的字符串(2 阶 Markov 模型生成口令结构字符串, 4 阶 LSTM 模型生成不同长度的字母串, PCFG 生成不同长度的数字串和特殊字符串)并分别存储。

(4) 填充阶段

遍历在生成阶段 Markov 模型产生的口令结构集合, 根据口令结构中的口令片段类型及长度遍历相应地在生成阶段产生的字符串集合并进行填充, 得到最终的候选口令集合。

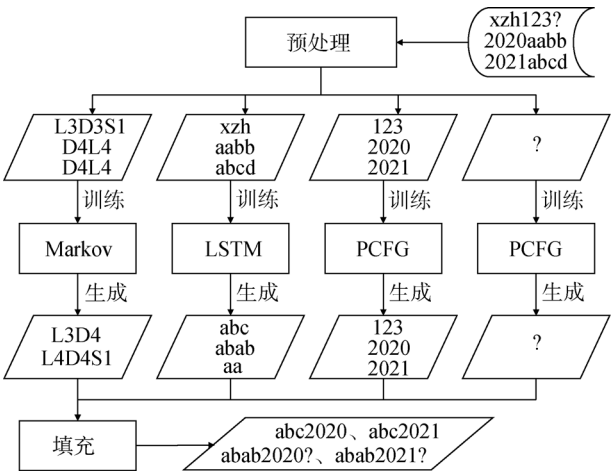

图 5 MLPP 示例

Figure 5 MLPP example

下面举例说明 MLPP 方法的猜测过程。如图 5 所示, 假设训练集中有 xzh123？、2020aabb、2020abcd 三个口令, 经过预处理、训练、生成阶段, 得到多种口令结构和口令片段, 然后根据每一个口令结构来遍历相应的口令片段对口令结构进行填充得到候选口令。理想的候选口令输出顺序应该是按照概率由高到低依次输出, 而候选口令的概率由口令片段中字符串的概率与口令结构的概率相乘得到, 例如口令 password123?的猜测概率的计算方式为

$$\begin{aligned}&\Pr(\text{abcde123?})\\&=\Pr_0(\text{L5D3})\Pr_0(\text{S1}|\text{L5D3})\Pr_0(\nabla|\text{D3S1})\\&*\Pr_1(\text{abcd})\Pr_1(\text{e}|\text{abcd})\Pr_2(123)\Pr_3(?)\end{aligned}$$

其中, $\Pr_0$ 表示通过 Markov 模型计算得到的口令结构的概率, $\Pr_1$、$\Pr_2$、$\Pr_3$ 分别表示字母串、数字串、特殊字符串在各自模型下计算得到的概率。为了形成一个概率空间, 本文使用终止符归一化方法[34], ∇ 代表终止符。同时, 归一化步骤已在计算口令结构概率时候完成, 因此在计算口令片段字符串的概率

时不再做归一化。

4 实验评估

4.1 口令集

本文实验基于从公开渠道获取到的 3 个口令集, 分别是 Gmail 口令集、Rockyou 口令集和 xato 口令集。Gmail 是谷歌公司的电子邮箱服务, 于 2014 年泄露将近 500 万条口令。Rockyou 是为用户提供照片等媒体服务的网站, 于 2009 年泄露超过 3200 万条数据。xato 是由安全研究人员 Mark Burnett 建立的发表信息安全相关文章的网站。2015 年, Mark Burnett 在 xato 上公布了在几年时间里采集到的 1000 万条用户口令数据, 形成了 xato 口令集。表 2 展示了上述口令集的具体信息。

表 2 口令集基本信息

Table 2 Password set basic information

站点	类型	泄露年份	口令集数量
Gmail	电子邮箱	2014	4 822 930
Rockyou	网络社交	2009	32 602 882
xato	信息安全	2015	9 555 952

下面分别对口令集中字母段、数字段、特殊字符段、口令结构进行统计分析, 如 3.2 节内容所示, 统计结果是确定在不同模块所使用模型的重要参考依据。

首先, 本文对不同口令片段进行分析, 表 3～表 5 为三个口令集中字母段、数字段和特殊字符段中出现频率最高的 10 个串。由于字母和数字在口令中出现较频繁, 太短的字符串无法反映有价值的信息, 因此本文对字母段和数字段进行统计时选择长度不小于 3 的串。

表 3 口令集中频率最高的 10 个字母串

Table 3 The 10 most frequent letter strings

Gmail	Rockyou	xato
password	password	password
qwerty	love	qwerty
abc	iloveyou	abc
love	princess	qwe
wsx	angel	qaz
monkey	ever	dragon
qwe	monkey	monkey
dragon	com	master
iloveyou	life	alex
qaz	babygirl	shadow

表 4 口令集中频率最高的 10 个数字串

Table 4 The 10 most frequent digit strings

Gmail	Rockyou	xato
123	123	123
123456	123456	123456
1234	123456789	1234
123456789	1234	123456789
12345	101	12345678
007	2007	12345
2010	143	777
12345678	2006	666
666	12345	2000
101	666	2010

表 5 口令集中频率最高的 10 个特殊字符串

Table 5 The 10 most frequent special strings

Gmail	Rockyou	xato
@	.	_
.	_	.
:	!	-
_	-	!
-	SPACE	@
*	@	*
$	*	$
#	#	#
+	/	?
SPACE	&	..

其次, 对口令集的口令结构进行分析。根据猜测的口令结构遍历相应字符串集合进行填充并生成这一口令结构代表的一簇候选口令是 MLPP 方法的核心思想, 口令结构越复杂, 程序中的循环层数越多, 候选口令生成的复杂度也就越高。为了减小程序复杂度, 需要对填充口令结构的复杂程度做一个合理的限制, 在不会显著影响猜测成功率的前提下, 摒弃过于复杂的口令结构来提高候选口令的生成速度。本文使用结构数来衡量口令结构的复杂程度, 结构数是指口令中顺序出现的口令片段数量, 比如 xzh123 的结构数为 2, xzh123xzh123 的结构数为 4。对于 Gmail、Rockyou、xato 口令集, 其口令结构复杂程度的统计结果如表 6 所示, 可以看出结构数大于 3 或 4 的口令占比很小, 因此, 将口令结构数的最大值限制为 3 或者 4, 可以满足在保证猜测成功率的前提下有效提高候选口令的生成速度的要求。

此外, 候选口令是在口令结构的基础上遍历相应的字符串集合填充而成的, 这就要求口令片段中字符串的猜测必须在生成最终候选口令开始之前完

表 6 口令结构复杂程度统计

Table 6 Password structure complexity statistics

口令集	结构数大于 2	结构数大于 3	结构数大于 4
Gmail	9.99%	4.51%	2.55%
Rockyou	6.46%	2.09%	1.06%
xato	11.05%	5.45%	3.05%

成, 即 MLPP 流程中填充阶段必须在生成阶段完成以后才能开始进行, 所以需要对要生成的口令片段字符串的长度做一个合理的限制, 否则, 生成并遍历类似于 D17 这种在真实口令集中出现概率极低的口令字符串会耗费很长时间。在本文中, 通过对真实口令集中字母段、数字段、特殊字符段这三类口令片段的长度占比进行统计分析, 分别确定了合适的长度阈值。统计结果如表 7～表 9 所示, 可以看出, 将字母串最大长度设置为 10、数字串最大长度设置为 10、特殊字符串最大长度设置为 3 时的候选口令空间即可以覆盖绝大多数的口令数据, 所以本文设置字母串长度阈值为 10、数字串长度阈值为 10、特殊字符串长度阈值为 3。

表 7 口令集中字母段长度占比

Table 7 Proportion of the length of letter strings

口令集	> 7	> 8	> 9	> 10
Gmail	30.62%	15.72%	8.69%	3.46%
Rockyou	29.65%	16.18%	8.93%	5.07%
xato	22.71%	9.09%	5.69%	3.60%

表 8 口令集中数字段长度占比

Table 8 Proportion of the length of digit strings

口令集	> 7	> 8	> 9	> 10
Gmail	12.45%	7.29%	4.54%	1.36%
Rockyou	17.22%	3.55%	2.22%	1.15%
xato	12.92%	6.54%	4.93%	0.43%

表 9 口令集中特殊字符段长度占比

Table 9 Proportion of the length of special strings

口令集	> 1	> 2	> 3
Gmail	8.91%	3.23%	1.36%
Rockyou	9.84%	3.54%	1.15%
xato	5.51%	1.93%	0.88%

4.2 实验参数

本文以 Gmail、Rockyou、xato 作为实验口令集, 将 MLPP 方法的实验结果与以 PassGAN、VAE、LSTM、DPG 为代表的现有的基于深度学习的口令猜测方法进行对比。为对比公平, 与文献[12]中的实验保持一致, 进行对比的 VAE 方法使用门控卷积神经网络(Gated Convolutional Neural Networks, GCNN)作为基本网络控制单元, 而另外三个对比模型 LSTM、PassGAN 和 DPG 则使用原始论文中的开源代码[35-37]。LSTM 代码中随机采样的生成方式一次只能生成10^6左右的候选口令, 因此将 LSTM 参数中的候选口令生成方式(guess_serialization_method)设置为顺序生成(参数值为 human)。其他 LSTM 参数和 PassGAN 参数、DPG 参数均为默认值。

在本文实验具体参数的设置上, 根据 4.1 节的分析结果, 将结构数的阈值定为 3, 即只对那些结构数不超过 3 的候选口令结构进行填充并生成最终的候选口令, 将在生成阶段产生的字母串长度阈值设为 10, 数字串长度阈值设为 10, 特殊字符串阈值设为 3, 这样可以在尽量不损失猜测成功率的前提下有效提高生成候选口令的速度。

本文实验分为两个实验场景, 场景一模拟同站口令猜测, 即分别以 Gmail、Rockyou、xato 的随机的 80%口令作为训练集, 剩余的 20%作为测试集; 场景二模拟跨站口令猜测, 此时训练集的选取与场景一保持一致, 但测试集为训练集之外的两个口令集。经过实验, 得到了 9 个实验结果, 分别如图 6～图 14 所示。这九个实验结果图的题注中标明了训练集和测试集, 例如 Gmail→Gmail 表示训练集是 80%的

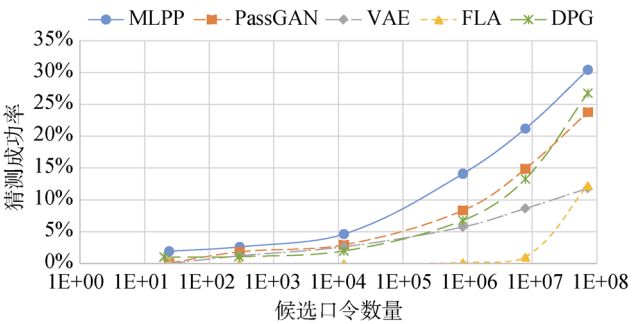

图 6 Gmail → Gmail

Figure 6 Gmail → Gmail

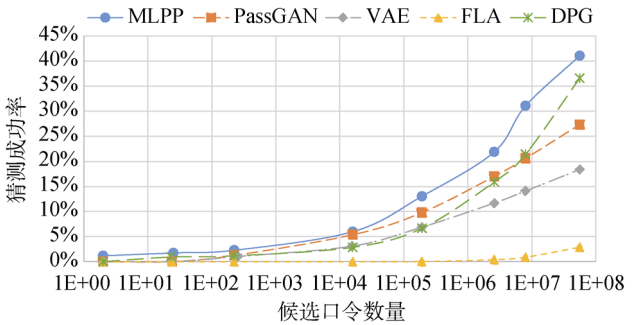

图 7 Rockyou → Rockyou

Figure 7 Rockyou → Rockyou

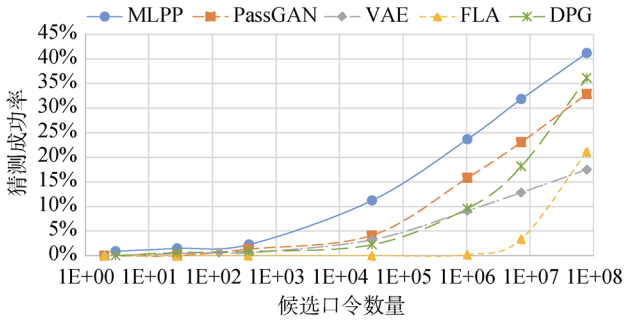

图 8　xato → xato

Figure 8　xato → xato

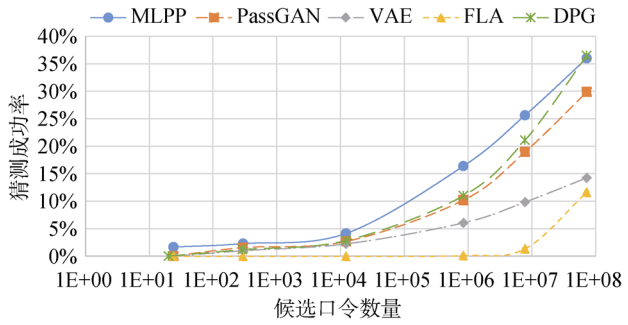

图 9　Gmail → Rockyou

Figure 9　Gmail → Rockyou

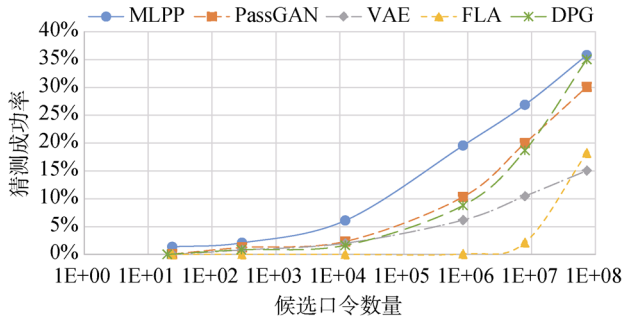

图 10　Gmail → xato

Figure 10　Gmail → xato

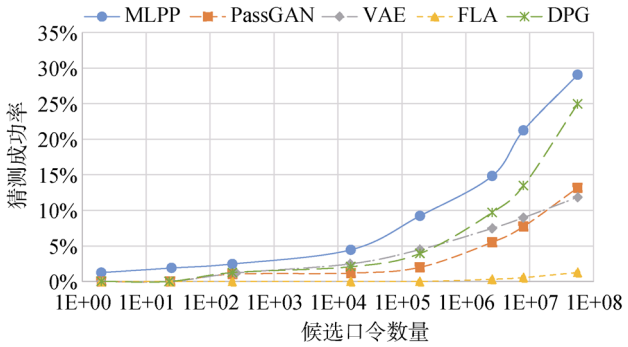

图 11　Rockyou → Gmail

Figure 11　Rockyou → Gmail

Gmail 口令，测试集为剩余的 20%的 Gmail 口令，Gmail→Rockyou 表示训练集是 80%的 Gmail 口令，测试集是整个 Rockyou 口令集。通过同站口令猜测成功率的对比，可以看出模型的学习能力，通过跨站口令猜测成功率的对比，可以看出模型的泛化能力。根据这九个实验的结果曲线，本文一方面考察 MLPP 猜测成功率随着口令猜测数增长的变化情况，另一方面通过与 PassGAN、VAE、LSTM、VAE 进行对比来验证针对深度学习口令猜测方法的组合优化是否有效。

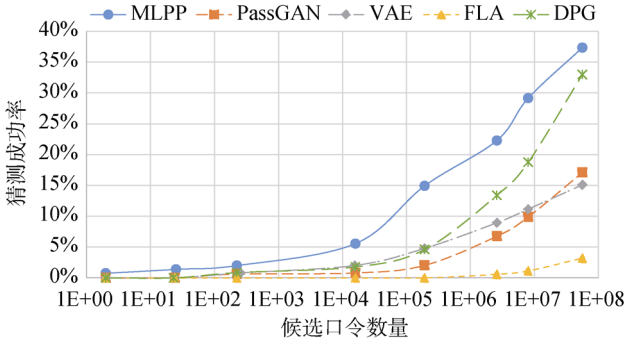

图 12　Rockyou → xato

Figure 12　Rockyou → xato

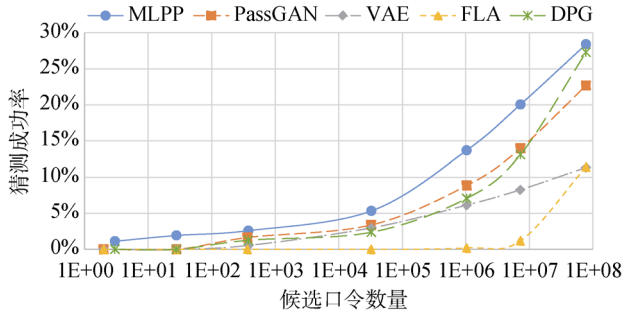

图 13　xato → Gmail

Figure 13　xato → Gmail

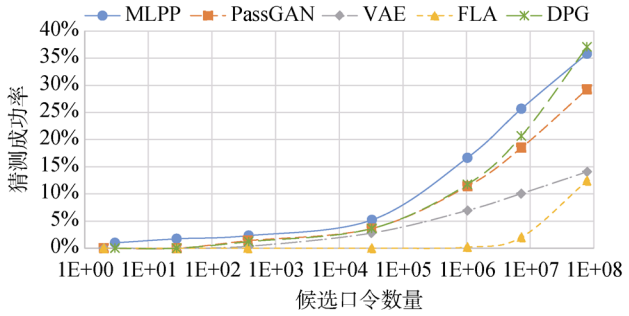

图 14　xato → Rockyou

Figure 14　xato → Rockyou

4.3　实验结果

4.3.1　同站口令猜测场景

同站口令猜测实验的结果如图 6～图 8 所示。可以看出，在口令猜测数不超过10^8时，MLPP 方法的

猜测成功率几乎始终在现有的基于深度学习模型的口令猜测方法之上, 平均提高 215.51%猜测成功率。具体地, 在三个不同口令集下, 与四种基于深度学习的口令猜测方法 PassGAN、VAE、LSTM、DPG 相比, MLPP 方法的猜测成功率分别相对提高了 25%～50%、123%～157%、95%～1321%、12%～14%, 而在 Rockyou 口令集下的提升最为明显, 与 PassGAN、VAE、LSTM、DPG 相比, MLPP 方法的猜测成功率分别提高了 50%、123%、1321%、12%, 体现了在同站口令猜测的情况下MLPP方法相比于基于深度学习模型的口令猜测方法在猜测成功率上的绝对优势。

4.3.2 跨站口令猜测场景

跨站口令猜测实验的结果如图 9～图 14 所示。从实验结果中可以看到, 模拟跨站口令猜测的实验场景二的结果与实验场景一基本类似, 在大多数实验中, MLPP 方法的表现最好。具体地, 与 PassGAN、VAE、LSTM 相比, MLPP 口令猜测方法的猜测成功率分别相对提高了 19%～121%、138%～155%、96%～2213%, 特别地, 当以 Rockyou 为训练集时, 与 PassGAN、VAE、LSTM 相比, MLPP 方法的猜测成功率分别至少提高了 119%、145%、1068%。与 DPG 方法相比, 在大部分的跨站口令猜测实验场景下, MLPP 方法的猜测成功率有 2%～16%等不同程度的提升。当以 Rockyou 作为测试集时, Rockyou 中足够多的口令使 DPG 方法可以充分地利用测试集信息校正隐空间, 其猜测成功率可以达到或者稍微超过本文提出的 MLPP 方法。具体地, 与 DPG 方法相比, 当以 Gmail 和 Rockyou 分别作为训练集和测试集时, MLPP 的猜测成功率降低了 1.32%, 当以 xato 和 Rockyou 分别作为训练集和测试集时, MLPP 的猜测成功率降低了 3.42%。

5 总结与展望

为了克服现有的基于深度学习的口令猜测方法中存在的未能利用口令的结构特征、训练计算量较大、在候选口令数目不大($\leq 10^8$)时口令猜测成功率较低的问题, 本文利用口令结构和口令片段之间的相互独立性, 对深度学习口令猜测方法以模块化形式进行组合优化。根据口令的结构特征将口令结构和不同口令片段的猜测拆解到不同的模块上进行, 进而引入基于统计的模型来对基于深度学习的口令猜测方法进行优化来获得更好的猜测效果。本文提出的经过组合优化的口令猜测方法同现有深度学习方法相比, 不仅同样可以生成大规模的候选口令, 而且可以达到更高的口令猜测成功率, 4.3 节的实验结果充分体现出了本文提出的优化方法对口令猜测成功率的巨大提升效果。除此之外, 口令猜测过程被拆解到不同模块以后, 各个模块之间是相互独立的, 模块的训练和生成可以并行运行, 进而模型可以同时对不同口令片段的猜测空间进行搜索, 提高了猜测效率。

未来仍存在以下方面需要继续研究: 一是模块组合形式的口令猜测方法相比单一的口令猜测模型参数数量更多, 任意模块参数的调整都可能带来整体口令猜测效果的改变, 需要研究系统地对这些参数进行调节和优化的方法, 使其能达到最佳的口令猜测效果; 二是口令中存在大量语义信息, 需要研究如何发掘口令中的语义信息并引入到本口令猜测方法中, 使其生成的候选口令更加符合用户设置习惯。

参考文献

[1] Zhou H, Wang J K, Wang B, et al. Comprehensive Overview of Plaintext Password Generation Models[J]. Computer Engineering and Applications, 2018, 54(4): 9-16.
(周浩, 王靖康, 王博, 等. 明文口令生成模型研究综述[J]. 计算机工程与应用, 2018, 54(4): 9-16.)

[2] Herley C, Van Oorschot P. A Research Agenda Acknowledging the Persistence of Passwords[J]. IEEE Security & Privacy, 2012, 10(1): 28-36.

[3] Bonneau J, Herley C, van Oorschot P C, et al. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes[C]. 2012 IEEE Symposium on Security and Privacy, 2012: 553-567.

[4] Freeman D, Jain S, Duermuth M, et al. Who are You? a Statistical Approach to Measuring User Authenticity[C]. Proceedings 2016 Network and Distributed System Security Symposium, 2016: 1-15.

[5] Wang D, Cheng H B, Wang P, et al. Zipf's Law in Passwords[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(11): 2776-2791.

[6] Bonneau J. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords[C]. 2012 IEEE Symposium on Security and Privacy, 2012: 538-552.

[7] Keith M, Shao B, Steinbart P J. The Usability of Passphrases for Authentication: An Empirical Field Study[J]. International Journal of Human-Computer Studies, 2007, 65(1): 17-28.

[8] Klein D V. Foiling the cracker: A survey of, and improvements to, password security[C]. Proceedings of the 2nd USENIX Security Workshop, 1990: 5-14.

[9] Morris R, Thompson K. Password Security[J]. Communications of the ACM, 1979, 22(11): 594-597.

[10] Melicher W, Ur B, Segreti S M, et al. Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks[C]. The 25th USENIX Conference on Security Symposium, 2016: 175-191.

[11] Hitaj B, Gasti P, Ateniese G, et al. PassGAN: A Deep Learning

Approach for Password Guessing[C]. International Conference on Applied Cryptography and Network Security. Cham: Springer, 2019: 217-237.

[12] Wang J W, Li Y, Chen X, et al. Modeling Password Guessability via Variational Auto-Encoder[C]. 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design, 2021: 348-353.

[13] Pasquini D, Gangwal A, Ateniese G, et al. Improving Password Guessing via Representation Learning[C]. 2021 IEEE Symposium on Security and Privacy, 2021: 1382-1399.

[14] Gao F. Research on password guessing method and evaluation technology based on deep learning[D]. Beijing: University of Chinese Academy of Sciences, 2019.
(高飞. 基于深度学习的口令猜测方法和评估技术研究[D]. 北京: 中国科学院大学, 2019.)

[15] Hashcat. https://hashcat.net/hashcat,Oct 2020.

[16] John the Ripper. URL https://www.openwall.com/john, Oct 2020.

[17] Narayanan A, Shmatikov V. Fast Dictionary Attacks on Passwords Using Time-Space Tradeoff[C]. The 12th ACM conference on Computer and communications security, 2005: 364-372.

[18] Dürmuth M, Angelstorf F, Castelluccia C, et al. OMEN: Faster Password Guessing Using an Ordered Markov Enumerator[C]. International Symposium on Engineering Secure Software and Systems. Cham: Springer, 2015: 119-132.

[19] Fujisaki T, Jelinek F, Cocke J, et al. A Probabilistic Parsing Method for Sentence Disambiguation[M]. Tomita M. Current Issues in Parsing Technology. Boston, MA: Springer, 1991: 139-152.

[20] Weir M, Aggarwal S, de Medeiros B, et al. Password Cracking Using Probabilistic Context-Free Grammars[C]. 2009 30th IEEE Symposium on Security and Privacy, 2009: 391-405.

[21] Graves A. Generating Sequences with Recurrent Neural Networks[EB/OL]. 2013: arXiv: 1308.0850. https://arxiv.org/abs/1308.0850

[22] Radford A, Metz L, Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks[EB/OL]. 2015: arXiv: 1511.06434. https://arxiv.org/abs/1511.06434

[23] Arjovsky M, Chintala S, Bottou L. Wasserstein GAN[EB/OL]. 2017: arXiv: 1701.07875. https://arxiv.org/abs/1701.07875

[24] Gulrajani I, Ahmed F, Arjovsky M, et al. Improved Training of Wasserstein GANs[C]. The 31st International Conference on Neural Information Processing Systems, 2017: 5769-5779.

[25] Yu L T, Zhang W N, Wang J, et al. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient[C]. The Thirty-First AAAI Conference on Artificial Intelligence, 2017: 2852-2858.

[26] Fedus W, Goodfellow I, Dai A M. MaskGAN: Better Text Generation via Filling in The_____[EB/OL]. 2018: arXiv: 1801.07736. https://arxiv.org/abs/1801.07736

[27] Guimaraes G L, Sanchez-Lengeling B, Outeiral C, et al. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models[EB/OL]. 2017: arXiv: 1705.10843. https://arxiv.org/abs/1705.10843

[28] Li Z G, Han W L, Xu W Y. A Large-Scale Empirical Analysis of Chinese Web Passwords[C]. The 23rd USENIX conference on Security Symposium, 2014: 559-574.

[29] Veras R, Collins C, Thorpe J. On Semantic Patterns of Passwords and their Security Impact[C]. NDSS, 2014: 1-16.

[30] Houshmand S, Aggarwal S, Flood R. Next Gen PCFG Password Cracking[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(8): 1776-1791.

[31] Li Y, Wang H N, Sun K. Personal Information in Passwords and Its Security Implications[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(10): 2320-2333.

[32] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[33] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative Adversarial Networks[J]. Communications of the ACM, 2020, 63(11): 139-144.

[34] Ma J, Yang W N, Luo M, et al. A Study of Probabilistic Password Models[C]. 2014 IEEE Symposium on Security and Privacy, 2014: 689-704.

[35] cupslab/neural_network_cracking. https://github.com/cupslab/neural_network_cracking. Oct 2020

[36] brannondorsey/PassGAN. https://github.com/brannondorsey/PassGAN.Oct 2020.

[37] pasquini-dario/PLR. https://github.com/pasquini-dario/plr.Oct 2020.

郁志红 于 2013 年在浙江海洋大学机械设计制造及其自动化专业获得学士学位。现在中国科学院信息工程研究所软件工程专业攻读硕士学位。研究领域为网络与系统安全。研究兴趣包括口令安全。Email: xizhihong@iie.ac.cn

周永彬 于 2004 年在中国科学院软件研究所计算机科学专业获得博士学位。现任中国科学院信息工程研究所研究员。研究领域为密码工程学。研究兴趣包括网络与信息安全理论及技术。Email: zhouyongbin@iie.ac.cn

李勇 于 2014 年在中国科学院大学计算机体系结构专业获得博士学位。现任中国科学院信息工程研究所高级工程师。研究领域为分布式系统、网络安全。研究兴趣包括口令安全、分布式系统。Email: liyong@iie.ac.cn

樊一康 于 2018 年在北京航空航天大学信息与通信工程专业获得工学硕士学位。现任中国科学院信息工程研究所助理研究员。研究领域为自然语言处理。研究兴趣包括人工智能安全、图神经网络。Email: fanyikang@iie.ac.cn

谢子平 于 2018 年在北京航空航天大学控制工程专业获得硕士学位。现任中国科学院信息工程研究所助理研究员。研究领域为 GPU 并行计算、口令安全。研究兴趣包括 GPU 并行计算、口令安全。Email: xieziping@iie.ac.cn

石瑞鑫 于 2018 年在北京理工大学物联网专业获得学士学位。现在中国科学院信息工程研究所网络空间安全专业攻读博士学位。研究领域为口令安全。研究兴趣包括口令与系统安全。Email: shiruixin@iie.ac.cn