计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210300008

MLSTM:一种基于多序列长度 LSTM 的口令猜测方法

常 庚[1] 赵 岚[2] 陈 文[1]
1 四川大学网络空间安全学院 成都 610065
2 西南电子设备研究所 成都 610036
(037173001@163.com)

摘 要 当前,口令仍然是重要的用户身份认证方式,使用有效的口令猜测方法来提高口令攻击的命中率是研究口令安全的主要方法之一。近年来,研究人员提出使用神经网络 LSTM 来实现口令猜测,并证实其命中率优于传统的 PCFG 口令猜测模型等。然而,传统 LSTM 模型存在序列长度选择困难的问题,无法学习到不同长度序列之间的关系。文中收集了大规模口令集合,通过对用户口令构造行为以及用户设置口令的偏好进行分析发现,用户个人信息对口令设置有重要影响。接着提出了多序列长度 LSTM 的口令猜测方法 MLSTM(Multi-LSTM),同时将个人信息应用到漫步口令猜测,以进一步提高猜测命中率。实验结果表明,与 PCFG 相比,MLSTM 的命中率最多提升了 68.2%,与传统 LSTM 和三阶马尔可夫相比,MLSTM 命中率的增加范围分别是 7.6%~42.1%和 23.6%~65.2%。

关键词:口令猜测;神经网络;口令分析;用户信息;口令安全
中图法分类号 TP309

MLSTM:A Password Guessing Method Based on Multiple Sequence Length LSTM

CHANG Geng[1],ZHAO Lan[2] and CHEN Wen[1]
1 School of Cyber Science and Engineering,Sichuan University,Chengdu 610065,China
2 Southwest China Research Institute of Electronic Equipment,Chengdu 610036,China

Abstract Password is one of the most important methods of user authentication. Using effective password guessing methods to improve the hit rate of password attacks is the main approach to study password security. In recent years,researchers have proposed to use long short-term memory (LSTM) neural network to guess password and have demonstrated it is superior to traditional password guessing models,such as Markov model and PCFG(probabilistic context free text) model. However,the traditional LSTM model has the problem that it is hard to select the length of the sequence and cannot learn the relationship between different length sequences. This paper collects large-scale password sets and analyzes the user's password construction behaviors and the preference for passwords setting,and finds that the user's personal information has important influences on the password settings. Then a multiple sequence lengths of LSTM password guessing model MLSTM(Multi-LSTM) is proposed and the personal information is applied to trawling guessing. Experimental results demonstrate that compared with PCFG,the cracking rate is increased by 68.2% at most. While compared with traditional LSTM and 3th-order Markov,the hit rates are increased by 7.6%~42.1% and 23.6%~65.2% respectively.

Keywords Password guessing,Neural network,Password analysis,User information,Password security

1 引言

近年来,图形认证、指纹认证和人脸识别等许多新的身份认证方法被相继提出,但是上述方法存在特殊硬件要求[1]、认证环境受限等[2]问题,无法从根本上代替口令认证[3]。目前,口令仍然是身份认证最广泛使用的手段[4-5],用户需要强制设置口令作为首要身份认证步骤。然而随着不断出现的口令泄露事件,口令认证的安全性受到了严重的威胁[6],因此研究口令分布规律、探索口令猜测方法及口令安全具有重要意义。

口令猜测攻击包括漫步猜测攻击和定向猜测攻击。在漫步猜测攻击中,攻击者没有特定的目标,其主要目的是破解

到稿日期:2021-03-01 返修日期:2021-07-19
基金项目:国家重点研发计划(2019QY0800);国家自然科学基金(61872255)
This work was supported by the National Key R & D Program of China(2019QY0800) and National Natural Science Foundation of China (61872255).
通信作者:陈文(wenchen@scu.edu.cn)

尽可能多的口令；在定向猜测攻击中，攻击者通过收集目标特定的个人信息以提高猜测效率，在有限数量的猜测内破解目标口令。口令猜测的代表性工具包括 HashCat[7] 和 JtR (John the Ripper)[8] 等，需要根据规则生成口令，命中率较低。而传统的漫步猜测方法主要包括马尔可夫模型(Markov Models)[9] 和概率上下文无关文法(Probabilistic Context-Free Grammars，PCFG)[10]，这两种模型均基于统计概率。2016 年，Melicher 等[11] 提出使用神经网络中的长短期记忆网络(LSTM)来实现口令猜测。它需要预先设置一个固定的序列长度，在生成口令时，根据输入序列预测口令的下一个字符。然而，如果设置的序列长度过短，就会导致预测的结果发散性太强，准确度低；反之，如果设置的序列长度过长，则会由于训练集的数据稀疏问题导致模型效果差，生成很多重复的口令。Hitaj 等[12] 提出了 passGAN，使用 GAN (Generative Adversarial Network)通过模型的对抗学习来提高口令猜测的准确性。

当前，个人信息在定向口令猜测中已经被大量应用，然而在漫步口令猜测中，由于用户个人特征信息的稀疏性，传统的马尔可夫模型和概率上下文无关文法难以直接将个人信息应用到口令猜测以提升大量口令的破解命中率。本文收集了近年来被泄露的 4 个真实口令数据集，通过对用户的口令构造习惯以及特征进行分析，系统地探索口令中的常见语义(如日期、姓名)，我们发现用户在构造口令时含有大量的个人信息，接着提出多序列长度口令生成模型，将个人信息引入训练过程。本文将所提模型与主流的口令猜测模型进行了比较，验证了该模型的有效性。

本文的贡献主要有以下两个方面：

(1)提出了一种口令生成模型，能够有效利用不同长度时间序列之间的关系，保证生成口令的多样性，提升口令猜测的命中率；

(2)将个人信息融入模型的训练过程，以提升口令猜测的针对性和目标命中率，避免了传统漫步口令攻击中无法有效利用个人信息的难题。

本文第 2 节介绍了主流口令猜测方法，包括 Markov、PCFG 以及神经网络；第 3 节主要通过对真实口令进行数据统计，分析用户的口令特征；第 4 节介绍了模型设计；第 5 节在数据集上进行实验，将所提模型与主流的口令猜测模型进行对比，得出实验结果；最后总结全文。

2　相关工作

2.1　Markov

2005 年 Narayanan 等[9] 首次提出使用 Markov 模型进行口令生成，其实质是一个统计模型，通过统计前后字符之间的频率来计算下一个字符的概率值。Markov 模型首先要定义阶(gram)，在 N 阶 Markov 模型中，需要统计出长度为 N 的子串之后临近的字符频数。该模型在进行模拟口令猜测攻击时可以分为两个阶段：训练集训练阶段和生成口令阶段。首先，在训练集训练阶段需要统计口令中每个子串之后邻近的字符频数，比如在 4 阶 Markov 模型中，口令“abc123”需要先统计首字符是“a”的频数，接着统计出“a”后是字符“b”的频数、“ab”后是字符“c”的频数、“abc”后是数字“1”的频数、“abc1”后是数字“1”的频数以及“bc12”后是数字“3”的频数。每个口令经过模型训练之后，所得的概率值相乘便可得到目标口令的概率值。因此得出口令“abc123”的概率的计算式为：

$$P(abc123)=P(a)\times P(b|a)\times P(c|ab)\times P(1|abc)\times P(2|abc1)\times P(3|bc12) \tag{1}$$

在生成口令阶段，基于得到每个口令的概率，依次输出猜测集。Ma 等[13] 对马尔可夫模型进行了改进，通过引入平滑方法帮助解决高阶马尔可夫模型中的稀疏性和过拟合问题。Wang 等[14] 将个人信息替换成不同标签，从而将马尔可夫模型应用于定向口令猜测。马尔可夫模型也可以通过计算破解目标口令的所需猜测数进行口令强度评估[15]。Markov 使用大量的统计信息，依据前文推测下一个字符出现的概率。由于 Markov 只是简单的概率统计，学习不到口令内隐含的特征，而且需要大规模的数据集进行训练才会变得更加有效。

2.2　PCFG

2009 年 Weir 等提出了一种基于概率上下文无关文法(Probabilistic Context Free Grammars，PCFG)的漫步口令猜测算法[10]。PCFG 的核心思想是将整个口令按照数字(D)、字母(L)和特殊字符(S)结构进行分割，该模型包括训练和生成口令两个阶段。在训练阶段，通过统计计算得到各种结构的频率以及各结构中各种字符组件的频率；比如口令“chang123 ”，PCFG 将其表示为 $L_5D_3S_1$ 结构，其中 L_5 代表“chang”，D_3 代表“123”，S_1 代表“？”。首先需计算出口令集中以 $L_5D_3S_1$ 作为模式的频率；接着计算出“chang”在长度为 5 的字母串(L_5)中的频率，“123”在长度为 3 的数字串(D_3)中的频率，以及“？”在长度为 1 的特殊符号(S_1)中的频率。因此口令“chang123”的概率为：

$$p(\text{`chang123?'})=p(L_5D_3S_1)\times p(L_5=\text{`chang'})\times p(D_3=\text{`123'})\times p(S_1=\text{`?'}) \tag{2}$$

在生成口令阶段，计算出每个口令的概率值，按照从高到低的顺序依次生成。因为口令都具有一定的规律性[16]，所以 PCFG 在生成口令中具有很好的效果。2014 年，Veras 等[17] 改进了 PCFG，使用 Natural Language Processing(NLP)来探索 L 字段的信息。Houshmand 等[18] 在 2015 年通过对口令进行分析，得出在口令生成中键盘规则和单词组合具有重要影响的结论，通过在 PCFG 添加键盘模式和多单词模式后，提升了口令破解的效果。2016 年，Li 等[19] 将 PCFG 应用于定向攻击猜测，该方法称为 Personal-PCFG，其使用个人信息将攻击对象的手机号、生日等信息分别用不同结构的标签替换。Hranicky 等[20] 通过对语法的修改，加快了密码猜测速度。然而 PCFG 依赖于统计训练集中的已有口令结构，无法产生新的口令结构。

2.3　神经网络 LSTM

递归神经网络(RNN)在处理时间序列数据上有很大

优势[21]，它基于上下文元素预测生成下一个元素的概率[22]，在自然语言处理领域已有广泛的使用，如用于文本预测、语义分析等。在 RNN 中，LSTM[23]引入多种类型的门操作来保证神经网络能够具有长期记忆的能力，并减少梯度爆炸问题。2016 年，Melicher 等[11]首次提出将 LSTM 用于口令生成，其根据输入的序列预测下一个字符。LSTM 模型又可以分为单词级和字符级，其生成口令的过程与 Markov 方法类似。两种方法均是计算字符产生概率来生成猜测口令集。神经网络经过学习训练来得到生成的各字符的概率并非基于简单的已有样本统计。其以字符序列为输入，以下一个字符作为标签，监督地学习序列的高维特征。然而，LSTM 需要提前设置序列长度，对于不同序列之间的关系，无法进行学习。

随着 GAN(Generative Adversarial Network)的兴起[24]，2017 年 Hitaj 等提出使用 GAN 来生成口令[12]。Nam 等[25]优化了 GAN 结构，采用了基于 RNN 以及使用双重鉴别器的 GAN。Xia 等[26]使用 PCFG 和 GAN 进行口令猜测，在训练阶段用标签代替字母、数字和特殊字符，将字符级训练扩展到单词级训练。使用 GAN 进行口令猜测会生成大量重复的密码，不能很好地学习到用户创建口令的习惯，这些口令在准确率上低于传统方式。

3 用户口令行为分析

由于互联网应用越来越多，为了方便记忆，用户往往不是随机地设置口令，因此了解用户的口令设置行为习惯是非常必要的。Wang[27]等也证明了口令分布服从 Zipf 定律。本节基于互联网上泄露的大量真实口令数据，通过统计学方法，发掘用户口令集组成的规律、用户设置口令的偏好，以及用户信息对其设置口令的影响。

3.1 数据集介绍

本文在互联网上收集了被泄露的 4 个中文网站口令数据，它们来自不同类型和规模的网站，总计 2078 万个口令，可以基本反映出用户创建口令时的特点，以及口令中字符的分布规律。其中一些数据集包含了邮箱、姓名和用户名等个人信息，本文仅仅使用其中的口令部分。

2014 年 12 月，火车购票系统出现了口令泄露事件，十万余条 12306 用户数据遭到泄露[28]，导致数据在网上疯狂传播。7k7k 口令集[29]是 7k7k 网站泄露的口令集合，该网站是一个游戏网站，用户群体比较年轻。178 口令集[30]是 178 网站泄露的口令集合，该网站是中国国内的一个游戏网站，用户多为年轻人。csdn[31]是目前中国开发者的交流平台，规定密码 length 8+，其用户大多为计算机行业人员，对口令安全度比较重视。

本文首先对数据集进行了清洗，主要工作包括：1)去除注册邮箱等无关字段，仅仅使用这些数据集中的口令部分；2)由于这些数据集中的口令可能会包含一些非 ASCII 码字符，会对实验结果产生干扰，因此对数据集进行了预处理，删除包含非 ASCII 字符的口令；3)由于基本上所有网站或应用对口令创建时的长度有要求，一般长度都是不小于或等于 6，另一方面，大多数用户在人为设定口令时，为了方便自己记忆和输入，不会选择长的口令，因此排除掉长度小于 6 以及长度大于 20 的口令。最终的数据集如表 1 所列。

表 1 数据集中的口令信息

Table 1 Password information in dataset

Dataset	Web service	Language	Leaked Time	Original	Length<6	Length>20	Not ASCII	After cleaning
12306	Train Ticketing	Chinese	Dec. 2014	131646	651	2	0	130993
7k7k	Gaming	Chinese	Dec. 2011	5365338	166056	48	62	5199172
178	Gaming	Chinese	Dec. 2011	9072892	268	819	0	9071805
csdn	Programmer Forum	Chinese	Dec. 2011	6428631	45483	46	301	6382801

3.2 包含个人信息的口令

在定向猜测攻击中，需要收集特定目标的个人信息。比如对“zhangsan”的账户发起攻击，则需要收集“zhangsan”的姓名、出生日期、手机号等。由于本文研究的是漫步口令攻击，想要在大规模口令中了解口令中的个人信息特点，必须重新构建语义词典。

为了构建中文姓名词典，使用了互联上收集到的姓名数据[32]。比如对于“zhangsan”，分别将“zhangsan”加入“拼音姓名全名”词典，“zhang”加入“拼音姓”词典，“zs”加入“拼音姓名缩写”词典。在日期字典中，生成了 1970-01-01 到 2015-01-01 的所有日期，并将它们加入词典，例如将“19900101”加入“YYYYMMDD”词典，“900101”加入“YYMMDD”词典。本文构造了 5 个不同类别的词典，使用最左最长匹配算法，将口令与字典中的每个字符匹配。

表 2 列出了 4 个口令数据集中用户口令与用户个人信息之间存在大量关联。其中，12306 口令集中个人信息占比最高，这可能是由于 12306 是中国铁路购票网站，属于日常使用软件，与个人生活息息相关。在创建口令中，用户喜欢加入日期，认为这会增加口令长度，增强安全性，同时也方便记忆。姓名拼音在口令中的占比更高，“拼音姓名缩写”甚至占比高达 33.274%。因此，姓名和日期在口令中具有重要的作用。

表 2 口令中的个人信息

Table 2 Personal information in passwords

(单位：%)

Semantic dictionary	12306	7k7k	178	csdn
Date_YYYYMMDD	6.465	5.542	3.819	8.282
Date_YYMMDD	7.430	8.027	7.259	7.944
Pinyin_fullname	13.410	9.221	11.511	12.396
Pinyin_familyname	16.562	5.401	10.183	8.224
Pinyin_name_abbr	33.274	14.204	23.716	26.341

3.3　口令中的姓氏分布

全国第六次人口普查资料显示[33]，以河北省为例，全省共有 1100 多个姓氏，其中 100 个姓氏的人口在 10 万人以上，超百万人的姓氏有 11 个。前十大姓氏约占常住人口的 49.1%，十大姓氏的拼音分别是："wang""zhang""li""liu""zhao""yang""chen""sun""ma""guo"。

姓氏在创建口令中频频使用，比姓名更具有普适性，然而现有工作还没有对此现象进行研究。本节将探究姓氏在口令数据集中的分布特点。表 3 列出了来自不同数据集的出现次数最高的前十大姓氏的拼音。从表中可以看出，虽然口令集来自不同网站，但在频率最高的一些姓氏上具有相似性。4 个数据集中均出现了 5 个姓氏的拼音"li""wang""xiao""zhang""yu"，其中前 3 名均出现了"li""wang"（二者均为十大姓氏的拼音）。3 个数据集中甚至均出现了 9 个姓氏的拼音"li""wang""liu""xiao""zhang""yu""ma"。178 数据集中前十大姓氏占比达到了 23.602%，最低比例在 csdn 中也达到了 19.165%。一些攻击在生成口令时，利用个人信息特点以及姓氏分布特性可以大大提升成功率。

表 3　数据集中占比最高的十大姓氏

Table 3　Top-10 most popular Pinyin_family name of each dataset

（单位：%）

Rank	12306	7k7k	178	csdn
1	li(4.501)	wang(3.964)	bai(3.933)	li(3.832)
2	wang(3.478)	ni(2.657)	li(3.129)	wang(2.529)
3	liu(2.380)	li(2.309)	wang(2.584)	liu(2.029)
4	xiao(2.275)	ma(2.000)	xiao(2.332)	xiao(1.663)
5	zhang(2.079)	wei(1.693)	ai(2.059)	zhang(1.653)
6	chen(1.817)	yu(1.454)	zhang(2.038)	ma(1.646)
7	yang(1.811)	xiao(1.445)	yu(2.014)	chen(1.560)
8	yu(1.723)	yang(1.302)	sun(1.880)	yang(1.525)
9	ma(1.664)	zhang(1.290)	liu(1.843)	yu(1.448)
10	wu(1.605)	chen(1.198)	he(1.792)	wei(1.280)
top-10	23.333	19.312	23.602	19.165

4　MLSTM 模型

4.1　传统 LSTM 模型

传统 LSTM 神经网络在口令猜测中需要提前设置固定的序列长度 L。在训练阶段根据长度 L 截取训练集，如果训练样本数据长度小于 L，则训练样本将无法将加入训练集。在生成时期，输入 L 的字符序列以预测生成口令的下一个字符。研究者在确定 LSTM 神经网络的序列长度 L 时，需要对某个训练集进行多次优化才能找到适合的固定序列长度。

假设模型设置的序列长度 L 为 4，在训练过程中，LSTM 采取滑动窗口模式截取训练集，具体如图 1 所示，个人信息如姓名拼音"lisi"的缩写"ls"长度为 2，"lisi"长度为 4，由于其后都不再有输入字符，因此"lisi"和"ls"等无法截取加入训练集。在生成过程中采用滑动窗口：第一次输入的数据需要在数据集中随机采样，例如取到"123a"时，预测出下一个字符"b"，在下一轮预测中，由于 LSTM 序列长度固定为 4，输入将会变为"23ab"，预测的下一个字符为"c"，最终生成了口令"123abc"。当模型设置的序列长度 L 较大时，使用较长的序列来预测下一个字符出现的概率，得到的结果更加准确。然而实际中的口令样本不足，导致训练不足，模型出现欠拟合的现象。

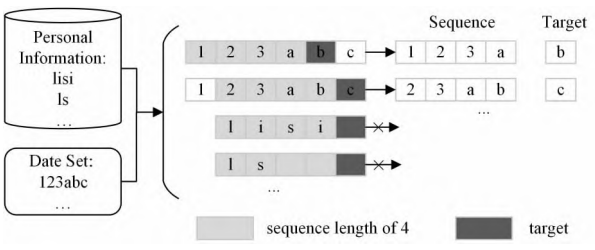

图 1　LSTM 训练集

Fig. 1　Training set of LSTM

如果设定模型的时间序列长度 L 为 2，在生成过程中假设想要生成口令"123abc"，假设第一次输入的数据取到"12"，预测出字符"3"；但在下一轮预测中，输入变为"23"，预测结果具有较大的随机性（在 12306 数据集上"23x"的熵值是"123x"的 1.21 倍），可能出现错误的预测结果。当模型设置的序列长度较小时，口令和个人信息都可以加入训练集，使用较短的序列来预测下一个字符出现的概率，生成的口令具有多样性，但是由于使用的个人信息序列太少，具有较大的随机性和不确定性，导致对当前位置的字符概率计算不够准确，生成更多无规则的字符串（见表 4），口令猜测结果的准确率低。

表 4　短序列 LSTM 生成的部分口令

Table 4　Sample of passwords generated by traditional LSTM with short sequence length

zxzxxxzxz0007	abcai6t	peng8211111
tlzli719	zxc1983561	z89773a
shunaingen	Gh19890	wx6561167qw
3833318164hh	2j7xsd	fhq7531312
13960z3	woder11	12314320wl

4.2　MLSTM

为了解决序列长度选择困难的问题，在模型中引入个人信息特征，本文提出了 MLSTM（Multi-LSTM）模型。MLSTM 模型由多个 LSTM 模型串联组成，在生成阶段，当输入的样本序列长度较短时，使用序列长度较短的 LSTM 模型，保证生成结果的多样性。短序列 LSTM 输出结果作为较长序列 LSTM 模型的输入，最大程度上使用学习到的信息，保证生成样本的准确性。

如图 2 所示，MLSTM 由两个阶段组成（即训练和生成）。在训练阶段分别训练出序列长度为 1 到 n 的神经网络模型，学习不同序列之间的关系。

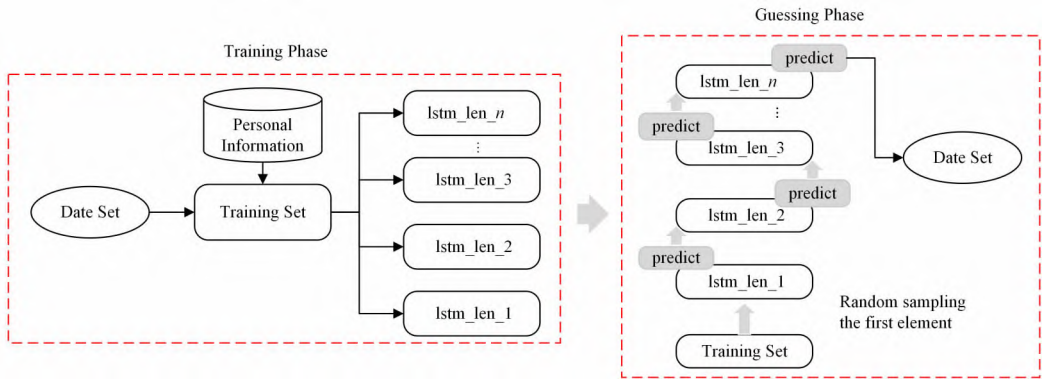

图 2 模型的两个阶段

Fig. 2 Two phases of model

同样以口令“123abc”为例，假设序列长度最大为 5，在训练阶段的训练集如图 3 所示，相比传统 LSTM(见图 1)，MLSTM 既能学习到“123”之间的关系和“abc”之间的关系，同时还能学习到“123”与“abc”之间隐含的关系。同时，对于 MLSTM 来说，拼音姓名缩写“ls(lisi)”可以由序列长度为 1 的模型截取加入训练集学习，姓名全称“lisi”可以由序列长度为 1，2 和 3 的模型截取加入训练集学习，因此能够有效地将个人信息加入训练集，利用不同序列长度的 LSTM 对个人信息进行学习。

在生成阶段：如果设定模型的初始序列长度为 1，第一次输入的数据从数据集进行随机取样，取到“1”，预测出字符“2”；其次使用序列长度为 2 的 LSTM，输入序列为“12”，预测字符标签为“3”；接着使用序列长度为 3 的 LSTM，输入序列为“123”，预测字符标签为“a”；然后使用序列长度为 4 的 LSTM，输入为“123a”，预测字符标签为“b”，以此类推，直到预测口令结束或者达到最大口令长度为止。

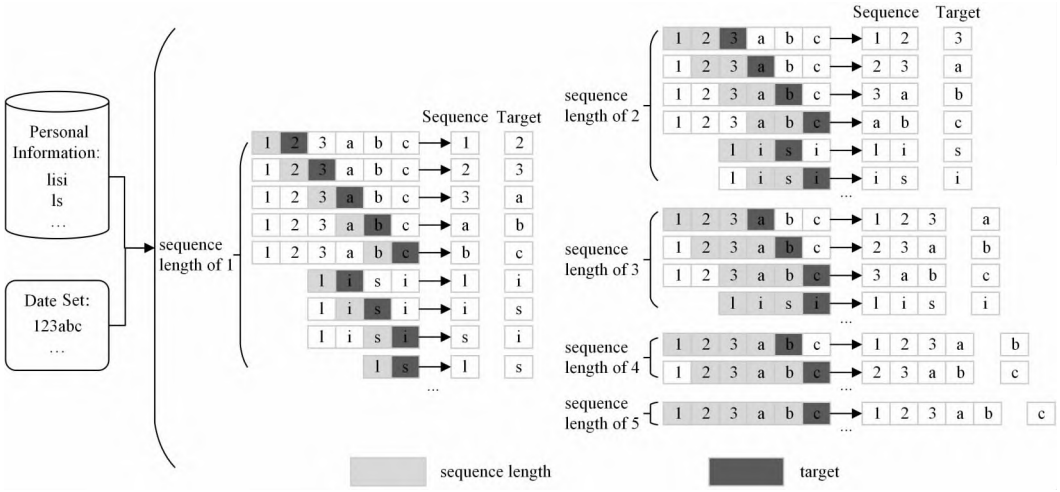

图 3 MLSTM 训练集

Fig. 3 Training set of MLSTM

经过多次实验论证，使用序列长度分别是 1～5 的 LSTM 神经网络来组成 MLSTM 效果最优(在实验测试中，当采用序列长度为 6 的模型时，会出现欠拟合现象。每个 LSTM 网络具有 2 个 LSTM 层，每个 LSTM 层具有 256 个神经元)。当序列长度为 5 时，采用滑动窗口方式预测后面的字符，直至口令结束或者达到最大长度。

5 实验结果与分析

本文将 MLSTM 与 Markov(其中 Markov 包括 3th-order Markov，4th-order Markov)、PCFG、LSTM 进行对比，3 种模型都需要对数据集进行训练。由于不同网站面对的用户群体不同，口令数据集呈现的特征也不同，因此模拟真实场景，使用 A 网站泄露的口令作为训练集进行训练，猜测阶段生成的口令攻击 B 网站，将命中率作为评价指标。

为了利用第 3 节中得出的中文口令的特性，在文献[32]中提取了高频的“拼音姓名全名”1 000 个(在拼音姓名全名也会学习到姓氏特征)和“拼音姓名缩写”1 000 个。在训练集太大的情况下，个人信息特征会被湮没，为了使姓名特征在训练中得到有效的权重，将从数据集中随机选择 10000 个口令，并将其加入前面提取的 2000 条个人信息作为训练集。在其他的全部数据集上验证有效性。例如从 12306 口令集随机选取 10000 个口令并加入 2000 条个人信息作为训练集，测试集是 178 数据集的所有口令。

实验结果图 4 所示，在 10^3 次猜测范围内，Markov 的性能最差，剩余各个模型的性能差别很小，在各个数据集上各有优势。由于本文研究的是漫步口令猜测，而不是定向口令猜测，不应将在小规模测试集上获得的最佳性能作为评估标准。

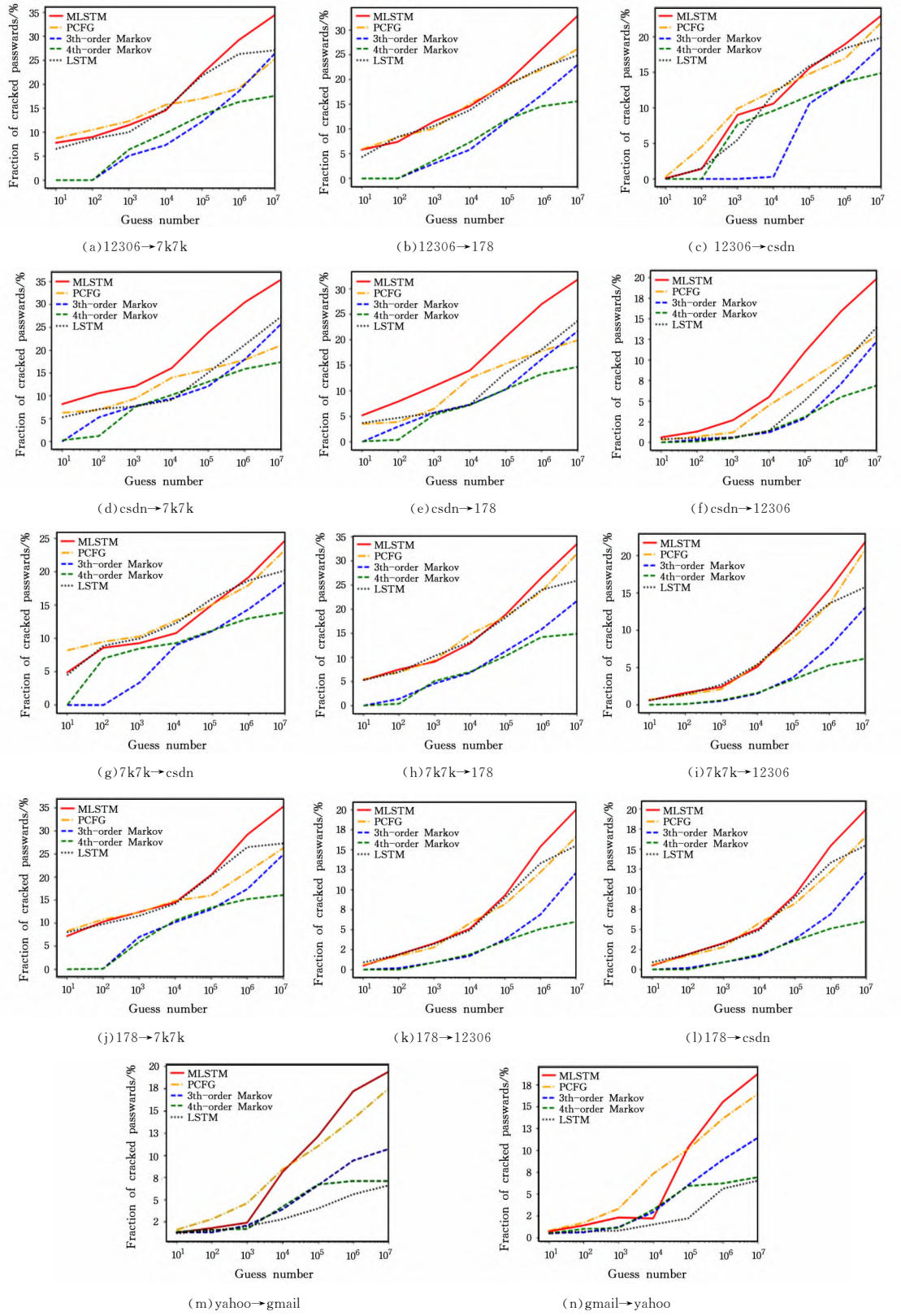

图 4　不同模型的实验结果

Fig. 4　Experiment results of different password guessing methods

在 10^5 次猜测前，Markov 的性能一直很差。与 Markov 相比，MLSTM 的命中率最多提升了 144.7%。随着猜测次数的增加，在 10^6 和 10^7 次猜测中 Markov 的效果逐渐提升，但 MLSTM 模型的优势更加明显，因此 Markov 和 MLSTM 差距仍然很大。尽管模型的匹配率随着更多的猜测次数而达到更高的水平，但是在猜测 10^7 次时，在不同测试集上 MLSTM 的命中率是最高的，比 PCFG 模型提升了 4.5%～68.2%，比 LSTM 模型提升了 7.6%～42.1%，比 Markov 模型中性能较好的 3th-order 提升了 23.6%～65.2%。

在训练 csdn 数据集中，如图 4(d)－图 4(f)所示，ML-

STM 攻击效果大大优于 LSTM,PCFG 以及 Markov。在猜测次数为 10^7 时,MLSTM 比其他模型中性能最好的 LSTM 提升了 42.1%。

可以明显看到,不同的训练集对实验结果有一定影响,但 MLSTM 的综合性能最好的,尤其是在 csdn 和 12306 训练集上其性能远超其他模型,这是因为口令中包含了大量的个人信息,而 csdn 和 12306 数据集的个人信息占比是最高的。同时 MLSTM 平衡了生成口令多样性和准确性。LSTM 的性能最稳定,受训练集的影响较小,性能始终处在中等。性能最差的是 4th-order Markov。PCFG 模型在 csdn 训练集上的性能最差,在其余的训练集上性能都较好,但是 MLSTM 在 csdn 训练集上的性能却是最好的。在 10^7 次猜测中(验证集为 178 数据集),MLSTM,PCFG,LSTM 和 Markov 模型(3th-order Markov)的命中率分别为 31.8%,19.9%,23.7% 和 21.7%。在攻击 12306 和 csdn 时,在猜测 10^7 次时所有模型的破解率始终没超过 25%,这是因为 csdn 规定密码长度大于 8,作为计算机行业论坛,其对口令安全度比较重视。对于 12306,由于口令数据有十万余条,样本分布具有不均匀性。

为了验证模型在多种语言情况下的有效性,本文在英文数据集 gmail[34] 和 yahoo[35] 上进行了测试,如图 4(m)—图 4(n)所示,可以看到 MLSTM 的效果均优于他模型,其中在 10^7 次猜测中 MLSTM 的破解率最高分别达到 19.4% 和 18.7%,在 yahoo 数据集上比 PCFG 提升了 14.2%。相较于传统模型,MLSTM 在序列长度较短时,保证了所生成候选口令的多样性;在序列长度较长时,通过长、短 LSTM 的连续迭代输入,最大程度地使用学习到的信息,提升了生成口令的准确性。

结束语 本文对中文口令集进行了统计分析,探索了几种口令属性(如个人信息占比和姓氏分布等),提出了多序列长度的神经网络口令猜测方法 MLSTM。在生成阶段,当输入的样本序列长度较短,使用序列长度较短的 LSTM 模型,可保证生成结果的多样性。短序列 LSTM 的输出结果作为较长序列 LSTM 模型的输入,进行口令生成。模型在保证生成结果的多样性的同时提升了准确性,解决了序列长度选择困难的问题。同时本文首次将个人信息引入基于神经网络的漫步口令猜测,利用模型特点将个人信息有效地加入训练过程。实验结果表明,MLSTM 的性能最好,与 PCFG 相比,MLSTM 在不同测试集上的命中率最多可提升 68.2%;与三阶马尔可夫链模型相比,MLSTM 的命中率增加范围为 23.6%~65.2%,比传统 LSTM 模型最多可提高 42.1%。MLSTM 方法在口令猜测方面相比 PCFG、传统 LSTM 和 Markov 模型具有明显的优势。同时实验结果也证明,用户在设置口令时,为了保证账户的安全性,应尽量少使用个人信息。

口令认证在很长一段时间内仍然是一种重要的认证方式,研究口令生成对用户设置更加安全的口令具有重要意义。在以后的工作中,将进一步研究口令更深层次的特征,探究口令生成的效率[36-37],以及优化模型结构使模型具有更好的适应性。

参 考 文 献

[1] BIDDLE R, CHIASSON S, VAN OORSCHOT P C. Graphical passwords: Learning from the first twelve years[J]. ACM Computing Surveys (CSUR), 2012, 44(4): 19.

[2] VAN DER PUTTE T, KEUNING J. Biometrical fingerprint recognition: don't get your fingers burned[C]// Smart Card Research and Advanced Applications. Boston: Springer, 2000: 289-303.

[3] ZHAO W, CHELLAPPA R, PHILLIPS P J, et al. Face recognition: A literature survey[J]. ACM Computing Surveys, 2003, 35(4): 399-458.

[4] BONNEAU J, HERLEY C, VAN OORSCHOT P C, et al. Passwords and the Evolution of Imperfect Authentication[J]. Communications of the ACM, 2015, 58(7): 78-87.

[5] WANG P, WANG D, HUANG X. Advances in password security[J]. Computer Research and Development, 2016, 53(10): 2173-2188.

[6] 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.

[7] Hashcat[OL]. https://hashcat.net/oclhashcat/.

[8] PESLYAK A. John the Ripper[OL]. http://www.openwall.com/john/.

[9] NARAYANAN A, SHMATIKOV V. Fast Dictionary Attacks on Passwords Using Time-Space Tradeoff[C]// Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS2005). Alexandria, VA, USA: ACM, 2005: 7-11.

[10] 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. IEEE, 2009: 391-405.

[11] MELICHER W, UR B, SEGRETI S M, et al. Fast, lean, and accurate: Modeling password guessability using neural networks [C]// Proceedings of USENIX Security. 2016.

[12] 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.

[13] MA J, YANG W, LUO M, et al. A study of probabilistic password models[C]// 2014 IEEE Symposium on Security and Privacy. IEEE, 2014: 689-704.

[14] WANG D, ZHANG Z, WANG P, et al. Targeted Online Password Guessing: An Underestimated Threat[C]// ACM CCS. 2016.

[15] DELL'AMICO M, MICHIARDI P, ROUDIER Y. Measuring Password Strength: An Empirical Analysis[J]. arXiv: 0907.3402, 2009.

[16] LI Z, HAN W, XU W. A Large-Scale Empirical Analysis of Chinese Web Passwords[C]// Usenix Conference on Security Symposium. USENIX Association, 2014.

[17] VERAS R, COLLINS C, THORPE J. On the Semantic Patterns of Passwords and their Security Impact[C]// Network & Dis-

tributed System Security Symposium. 2014.

[18] HOUSHMAND S,AGGARWAL S,FLOOD R. Next Gen PCFG Password Cracking[J]. IEEE Transactions on Information Forensics & Security,2017,10(8):1776-1791.

[19] LI Y,WANG H,SUN K. A study of personal information in human-chosen passwords and its security implications. [C]// IEEE Conference on Computer Communications(INFOCOM 2016). IEEE,2016.

[20] HRANICKÝ R, LIŠTIAK F, MIKUŠ D,et al. On practical aspects of PCFG password cracking[C]// IFIP Annual Conference on Data and Applications Security and Privacy. Cham:Springer, 2019:43-60.

[21] SUTSKEVER I,MARTENS J,HINTON G E. Generating Text with Recurrent Neural Networks[C]// International Conference on Machine Learning. DBLP,2016.

[22] GRAVE A. Generating sequences with recurrent neural networks[J]. arXiv:1308.0850,2013.

[23] SUNDERMEYER M,SCHLÜTER R,NEY H. LSTM Neural Networks for Language Modeling[C]// Interspeech. 2012.

[24] MIRZA M, OSINDERO S. Conditional generative adversarial nets[J]. arXiv:1411.1784,2014.

[25] NAM S,JEON S,KIM H,et al. Recurrent GANs Password Cracker For IoT Password Security Enhancement[J]. Sensors, 2020,20(11):3106.

[26] XIA Z Y,YI P,LIU Y Y,et al. GENPass:A Multi-Source Deep Learning Model for Password Guessing[J]. IEEE Transactions on Multimedia,2019,22(5):1323-1332.

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

[28] 12306[OL]. http://www.12306.com/.

[29] 7k7k[OL]. http://www.7k7k.com/.

[30] 178[OL]. http://www.178.com/.

[31] csdn[OL]. http://www.csdn.net/.

[32] https://github.com/wainshine/Chinese-Names-Corpus.

[33] The Sixth National Census [EB/OL]. (2012-02-28). http://www.stats.gov.cn/ztjc/zdtjgz/zgrkpc/dlcrkpc/.

[34] gmail[OL]. http://gmail.google.com.

[35] yahoo[OL]. http://www.yahoo.com.

[36] XIE Z J,ZHANG M,LI Z H, et al. Analysis of Large-scale Real User Password Data Based on Cracking Algorithms[J]. Computer Science,2020,47(11):48-54.

[37] LI B,ZHOU Q L,SI X M,et al. Optimized Implementation of Office Password Recovery Based on FPGA Cluster[J]. Computer Science,2020,47(11):32-41.

CHANG Geng, born in 1998, postgraduate. His main research interests include password security and deep learning.

CHEN Wen, born in 1983, Ph.D, associate professor, master supervisor, is a member of China Computer Federation. His main research interests include network security and data mining.

(责任编辑:喻藜)