快速、精简、准确:使用神经网络对密码猜测性建模

会议文件2016年8月

7位作者，包括:

卢乔·鲍尔
卡耐基梅隆大学
132份出版物8，188次引用
查看个人资料

威廉·梅利彻
卡耐基梅隆大学
14份出版物824次引用
查看个人资料

尼古拉斯·克里斯汀
卡耐基梅隆大学
138份出版物8，494次引用
查看个人资料

洛里·克兰诺
卡耐基梅隆大学
293份出版物24，938次引用
查看个人资料

《第23届USENIX安全研讨会论文集》(USENIX安全2016)，第175-191页。德克萨斯州奥斯汀。2016年8月。

快速、精简、准确：使用神经网络对密码可猜测性建模

威廉·梅利彻、布拉瑟·乌尔、肖恩·m·塞格雷提、萨兰加·科曼杜里、卢乔·鲍尔、尼古拉斯·克里斯汀、洛里·费思·克兰诺
卡耐基梅隆大学

摘要

人类选择的文本密码是当今占主导地位的身份验证形式，很容易受到猜测攻击。不幸的是，通过模拟对抗性密码猜测来评估密码强度的现有方法要么不准确，要么数量级太大，对于实时客户端密码检查来说太慢。我们建议使用人工神经网络来模拟文本密码抵抗猜测攻击的能力，并研究不同的结构和训练方法如何影响神经网络的猜测效果。我们表明，神经网络通常可以比最先进的方法(如概率上下文无关文法和马尔可夫模型)更有效地猜测密码。我们还展示了我们的神经网络可以被高度压缩——压缩到几百千字节——而不会显著降低猜测效率。基于这些结果，我们在JavaScript中实现了第一个原则性的客户端密码猜测模型，该模型分析了密码对亚秒延迟的任意持续时间的猜测攻击的抵抗力。总之，我们的贡献使更准确和实用的密码检查比以前可能的。

1 介绍

文本密码是目前最常见的身份验证形式，在可预见的未来，它将继续保持这种形式[53].不幸的是，用户通常选择可预测的密码，这使得密码猜测攻击成为可能。作为响应，主动密码检查用于评估密码强度[19].

评估密码强度的一种常见方法是运行或模拟密码猜测技术[35,59,92].一套配置良好的猜测技术，包括概率方法[37,65,93]和现成的密码恢复工具[74,83]，可以准确地对漏洞建模专家攻击者猜测的密码[89].不幸的是，这些技术通常是计算密集型的，需要几百兆字节到几十兆字节的磁盘空间，并且需要几天的时间来执行。因此，它们通常不适合实时评估密码强度，有时也不适合任何实际有用的密码强度评估。

为了更准确和更实用地测量人类选择的文本密码的强度，我们建议使用人工神经网络来猜测密码。人工神经网络(以下称为“神经网络”)是一种机器学习技术，旨在逼近高维函数。已经证明它们在产生新序列方面非常有效[49,84]，这表明它非常适合生成密码猜测。

在本文中，我们首先综合测试了改变神经网络模型大小、模型结构、训练数据和训练技术对网络猜测不同类型密码的能力的影响。我们将我们的神经网络实现与最先进的密码猜测模型进行比较，包括广泛研究的马尔可夫模型[65]和概率上下文无关文法[59,93]，以及使用损坏的字典条目的软件工具[74,83].在我们的测试中，我们使用最近提出的蒙特卡罗方法来评估概率模型对大量猜测的性能[34].我们发现神经网络通常比其他密码猜测方法更成功地猜测密码，尤其是在猜测超过1010次和非传统密码策略的情况下。这些案例非常有趣，因为密码猜测攻击通常会进行1010次以上的猜测[44,46]并且因为现有的密码猜测攻击在新的非传统密码策略上表现不佳[79,80].

虽然使用神经网络进行更有效的密码猜测本身就是一个重要的贡献，但我们也展示了我们使用的神经网络可以在猜测效率损失最小的情况下被高度压缩

完美。因此，我们的方法比现有的密码猜测方法更适合客户端密码检查。大多数现有的客户端密码检查器是不准确的[33]因为它们依赖于简单、容易压缩的试探法，例如计算密码中的字符数或字符类别。相比之下，我们表明高度压缩的神经网络比现有的客户端检查器更准确地测量密码强度。我们可以将这样的神经网络压缩到数百千字节，小到足以包含在移动设备的应用程序中，用加密软件捆绑，或用在网页密码计量器中。

为了证明神经网络对于客户端密码检查的实际适用性，我们用JavaScript实现并测试了一个神经网络密码检查器。这个实现，我们作为开源软件重新发布了，[1] 立即适用于移动应用程序、浏览器扩展和网页密码计量器。我们的实现在几分之一秒内给出了关于密码强度的实时反馈，并且它比现有的客户端方法更准确地测量了对猜测的抵抗力。

总之，本文做出了三个主要贡献，它们共同大大提高了我们检测和帮助消除弱密码的能力。首先，我们提出神经网络作为猜测人类选择的密码的模型，并全面评估如何改变它们的训练，参数和压缩影响猜测的有效性。在许多情况下，神经网络比最先进的技术猜测得更准确。第二，利用神经网络，我们创建了一个密码猜测模型，该模型对于客户端主动密码检查来说足够可压缩和有效。第三，我们构建并测试了这样一个检查器的JavaScript实现。在运行在商用硬件上的普通web浏览器中，这种实现模拟了任意高数量的具有亚秒延迟的敌对猜测，而只需要将数百千字节的数据传输到客户端。总的来说，我们的贡献能够在更广泛的常见场景中实现比以前更准确的主动密码检查。

2 背景和相关工作

为了强调密码强度的重要性，我们首先总结一下密码猜测攻击。然后，我们讨论评估密码强度的指标和模型，以及在密码创建过程中评估密码强度的轻量级方法。最后，我们总结了先前使用神经网络生成文本的工作。

[1] https://github.com/cupslab/neural_network_cracking

2.1 密码猜测攻击

密码易受猜测攻击的程度取决于具体情况。对于网络钓鱼攻击、键盘记录器或肩扛冲浪，密码强度无关紧要。一些系统实施了限速里坡，在少量错误尝试后锁定一个在线账户或设备。在这些情况下，除了最容易预测的百万个密码之外，其他密码都不太可能被猜到[39].

然而，在另外三种情况下，猜测攻击是一种威胁。首先，如果速率限制没有得到适当实施，据信2014年苹果iCloud上名人个人照片被盗就是这种情况50]，大规模猜测成为可能。第二，如果散列密码的数据库被盗，不幸的是这经常发生[20,23,27,45,46,67,73,75,87]，离线攻击是可能的。攻击者选择可能的候选密码，对其进行哈希处理，并在数据库中搜索匹配的哈希。当发现匹配时，攻击者可以利用跨帐户重复使用密码的高可能性，在其他系统上尝试相同的凭据[32].利用密码重用的攻击会带来现实世界的后果，包括最近Mozilla的Bugzilla数据库因管理员重用密码而遭到攻击[76]以及类似于易贝的中国在线购物网站淘宝上的2000万个账户因密码重复使用而受损[36].第三，加密密钥材料来源于密码或受密码保护的常见情况容易受到大规模猜测的攻击，就像网上账户的散列密码数据库一样。例如，对于跨设备同步的密码管理器[52]或隐私保护云备份工具(如SpiderOak [82])，存储在云端的文件安全性直接取决于密码强度。此外，用于非对称安全消息的加密密钥(例如，GPG私钥)、磁盘加密工具(例如，TrueCrypt)和Windows域Kerberos票据[31]受人工生成的密码保护。如果包含该密钥材料的文件被泄露，密码的强度对于安全性是至关重要的。这最后一个场景的重要性可能会随着采用而增加密码管理器和加密工具。

2.2 测量密码强度

密码强度模型通常采用两种概念形式中的一种。第一种依赖于纯粹的统计方法，如香农熵或其他先进的统计方法21,22].然而，由于需要不切实际的大样本量，我们认为这些类型的模型超出了范围。第二种概念性方法是模拟对抗性密码猜测

ing [34,65,89]. 我们的神经网络应用遵循这种方法。下面，我们描述了学术界广泛研究并用于对抗性密码破解的密码猜测方法，我们在分析中将其与神经网络进行了比较。密码猜测的学术研究集中在概率方法上，该方法将大量密码集作为输入，然后以递减的概率顺序输出猜测。密码破解工具依赖于高效的试探法来模拟常见的密码特征。

概率上下文无关文法一种可能的方法使用概率上下文无关文法(PCFGs)93]. PCFGs背后的直觉是密码是用模板结构(例如，6个字母后跟2个数字)和适合这些结构的终端构建的。密码的概率是其结构的概率乘以其终端的概率。

研究人员发现，对结构和终端使用不同的训练源可以提高猜测能力59]. 通过平滑将概率分配给看不见的终端，以及通过从训练数据中逐字提取密码而不将其抽象到语法中的语法生成的属性猜测也是有益的60]. 此外，使用自然语言字典来实例化终端改进了猜测，特别是对于长密码[91].

马尔可夫模型使用马尔可夫模型来猜测密码，于2005年首次提出[70]，最近得到了更全面的研究[37,65]. 从概念上讲，马尔可夫模型根据前面的字符或上下文字符来预测密码中下一个字符的概率。使用更多的上下文字符可以允许更好的猜测，但是有过度拟合的风险。平滑和补偿方法补偿过度拟合。研究人员发现，具有加法平滑的6克马尔可夫模型通常是对英语密码建模的最佳模型[65]. 我们在分析中使用这种配置。

在对抗性密码破解中，软件工具通常用于生成密码猜测[44]. 最流行的工具使用管理规则来转换单词列表(密码和字典条目)，或者用于模拟人类如何设计密码的常见行为的转换。例如，篡改规则可以附加一个数字，并将每个“a”改为“@”。这种类型的两个流行工具是Hash- cat [83]和开膛手约翰(JtR，[74]). 虽然这些方法不是直接基于统计建模，但是它们产生了相当准确的猜测89]这导致了它们的广泛使用[44].

2.3 主动密码检查

尽管前面讨论的密码猜测模型可以准确地模拟人类创建的密码[89]，它们需要数小时或数天以及数兆字节或数千兆字节的磁盘空间，这使得它们太耗费资源而无法向用户提供实时反馈。当前的实时密码检查器可以根据它们是否完全在客户端运行来分类。带有服务器端组件的Checkers可以更加精确，因为它们可以利用大量数据。例如，研究人员建议使用服务器端马尔可夫模型来衡量密码强度26]. 其他人已经研究了使用来自泄露的密码和自然语言语料库的训练数据来显示用户对他们下一步将键入什么的预测61].

不幸的是，服务器端组件给安全性带来了很大的缺点。在某些情况下，将密码发送到服务器进行密码检查会破坏所有的安全保证。例如，保护加密卷(如TrueCrypt)或加密密钥(如GPG)的密码，以及密码管理器的主密码，都不应离开用户的设备，即使是主动密码检查。因此，这些安全关键型应用程序经常无法进行准确的密码检查。在密码最终被发送到服务器的情况下(例如，对于一个在线帐户)，实时的服务器端组件既增加了等待时间，又打开了密码计量器，以应对基于键盘计时、消息大小和缓存的强大的旁路攻击[81].

先前的客户端密码检查器，例如那些完全在网络浏览器中运行的，依赖于易于编码的试探法。许多常见的血糖仪根据密码的长度或包含的不同字符类别对密码进行评级[33,88]. 不幸的是，在对客户端和服务器端密码计量器的全面测试中，除了一个计量器之外，其他计量器都非常不准确[33]. 仅zxcvbn[94,95]，它使用了几十种更高级的试探法，给出了相当准确的强度估计。然而，由于不能简洁地编码模型和计算实时结果，这种计量器不能直接模拟对抗性猜测。相比之下，我们的方法完全在客户端模拟对抗性猜测。

2.4 神经网络

我们用来模拟密码的神经网络是一种用于逼近高维函数的机器学习技术。它们被设计用来模拟人类神经，尤其擅长模糊分类问题和生成新序列。我们生成候选密码猜测值的方法很大程度上借鉴了先前的工作

基于前面元素的字符串中的下一个元素[49,84].例如，在生成字符串password时，可能会给神经网络passwor，并输出下一个出现d的概率很高。

虽然密码创建和文本生成在概念上是相似的，但是很少有研究尝试使用文本生成的见解来模拟密码。十年前，神经网络被提出作为一种方法

用于将密码分为两大类(弱或强)[30]，但这项工作并没有试图对密码被猜测的顺序或猜测攻击的其他方面进行建模。据我们所知，在密码猜测攻击中使用神经网络的唯一建议是最近的一篇博客文章[71].与我们为使神经网络在实践中有效而对不同参数进行的大量测试形成鲜明对比的是，这项工作几乎没有对神经网络的应用进行改进，这使得作者怀疑这种方法“是否有任何实际意义”。此外，这项工作只试图模拟一些可能的密码猜测，而不是我们使用蒙特卡罗方法来模拟任意数量的猜测。

从概念上讲，神经网络比其他方法有优势。与PCFGs和马尔可夫模型相比，由神经网络产生的序列可以是不精确的、新颖的序列49]，这使我们推断神经网络可能适用于密码猜测。概率密码猜测的现有方法(例如，马尔可夫模型[26])对内存的要求非常高，仅在客户端是不切实际的。然而，神经网络可以在比马尔可夫模型小得多的空间中模拟自然语言68].神经网络也被证明可以将一个任务的知识转移到相关的任务中97].这对于瞄准新颖的密码组合策略是至关重要的，对于这种策略，训练数据充其量是稀疏的。

3 系统设计

我们在一个大的设计空间中试验了广泛的选项，并最终达成了一个系统设计，该系统设计1)利用神经网络进行密码猜测，2)提供客户端猜测估计方法。

3.1 测量密码强度

与马尔可夫模型类似，我们系统中的神经网络被训练成在给定密码的前面字符的情况下生成密码的下一个字符。图1 说明了我们的建设。就像马尔可夫模型一样34,65]，我们依靠一个特殊的密码结束符号来模拟一个字符序列后密码结束的概率。例如，为了计算整个密码“坏”的概率，我们将

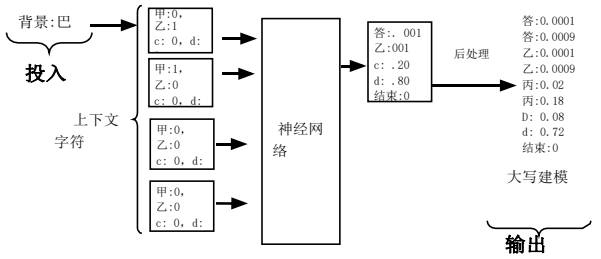


图1:使用神经网络预测密码片段的下一个字符的示例。给定上下文“ba ”,网络被用于预测“d”。这个网络使用四个字符的上下文。每个下一个字符的概率是网络的输出。网络上的后处理可以推断大写字符的概率。

从一个空密码开始，查询网络，看能不能看到一个“b”，然后在“b”后面看到一个“a”，然后在“ba”后面看到一个“d”，然后在“bad”后面看到一个完整的密码。为了从神经网络模型生成密码，我们使用改进的波束搜索[64]，深度优先和广度优先搜索的混合。如有必要，我们可以通过过滤这些密码来抑制不需要的密码(例如，违反目标密码策略的密码)的生成。然后，我们按概率对密码进行排序。我们使用波束搜索，因为广度优先的内存需求不可扩展，还因为它允许我们比深度优先搜索更好地利用GPU并行处理能力。从根本上说，这种猜测枚举的方法类似于马尔可夫模型中使用的方法，它可以从相同的优化中受益，例如近似排序[37].与马尔可夫模型相比，神经网络模型的一个主要优势是可以在GPU上高效地实现。

计算猜测数在通过模拟猜测攻击来评估密码强度的过程中，我们计算密码的猜测数，或者说，如果按照可能性的降序来猜测密码，攻击者需要猜测多少次才能得到该密码。通过枚举计算猜测数的传统方法计算量很大。例如，在我们对NVidia GeForce GTX 980 Ti的未优化实施中，枚举超过1010个密码大约需要16天。然而，除了猜测数计数之外，我们还可以使用蒙特卡罗模拟准确有效地估计猜测数，如Dell'Amico和Filippone所建议的[34].

3.2 我们的方法

训练神经网络需要许多设计决策。设计空间迫使我们决定

建模字母表、上下文大小、神经网络结构的类型、训练数据和训练方法。我们沿着这些维度进行实验。

模型架构在这项工作中，我们使用递归神经网络，因为它们已被证明对在字符级自然语言的上下文中生成文本是有用的49,84].递归神经网络是一种特定类型的神经网络，其中网络中的连接可以处理序列中的元素，并使用内部存储器来记住关于序列中先前元素的信息。我们在第一节中实验了两种不同的递归架构5.1。

字母表大小我们专注于字符级模型，而不是更常见的单词级模型，因为没有建立用于pass- word生成的单词字典。我们还补充了我们的分析与探索性的实验使用音节水平的模型5.1。我们决定探索基于机器学习先前工作的混合模型[68].在混合结构中，除了字符之外，神经网络还可以模拟子词单元，如音节或记号。我们选择基于之前的工作对2，000个不同的令牌进行建模[68]并以我们处理字符的方式来表示这些记号。对标记化模型更彻底的研究将探索更多和更少的标记。使用标记化的结构，模型然后可以输出下一个字符是‘a’或标记‘pass’的概率。我们通过沿着字符类别边界对训练集中的单词进行kenizing并选择2000个最常用的单词来生成标记列表。

像以前的工作[26]，我们根据经验观察到，对所有字符进行建模会给模型带来不必要的负担，并且一些字符，如大写字母和罕见符号，在神经网络之外进行建模会更好。我们仍然可以通过将模型的输出解释为模板来创建带有这些字符的密码。例如，当神经网络预测“A”字符时，我们对预测进行后处理，根据“A”和“A”在训练数据中出现的次数分配它们各自的概率，从而预测“A”和“A ”,如图所示1。这里的直觉是，当替代的启发式方法可以有效地模拟某些现象(例如，小写字母和大写字母之间的转换)时，我们可以减少神经网络消耗的资源量。

密码上下文预测依赖于上下文字符。例如，在图中1上下文字符是“ba ”,目标预测是“d”。增加上下文字符的数量会增加训练时间，而减少上下文字符的数量可能会降低猜测的成功率。

我们尝试使用密码中所有前面的字符作为上下文，并且只使用前面的十个字符。我们在初步测试中发现，使用10个字符在猜测和训练上同样成功，而且速度快了一个数量级，因此决定采用这一选择。当上下文字符少于十个时，我们用零填充输入。相比之下，表现最好的马尔可夫模型通常使用上下文的五个特征[34,65].如果给定太多的上下文，马尔可夫模型可能会过度拟合，而神经网络通常会在参数太多时过度拟合。

以相反的顺序提供上下文字符(例如，从“rowssap”而不是“passwor”预测“d ”)有时会提高性能[48].我们在第一节中根据经验评估这种技术5.1。

模型大小我们还必须决定模型中包含多少参数。为了衡量改变模型大小对猜测成功的影响，我们测试了一个具有15，700，675个参数的大型神经网络和一个具有682，851个参数的较小网络。选择较大的尺寸是为了限制模型使用的时间和GPU内存，这需要一个半星期来在我们较大的训练集上进行完全训练。较小的尺寸被选择用于我们的浏览器实现，因为它实际上可以通过互联网发送；压缩后，这个网络有几百千字节。我们使用各种密码策略评估了两种规模的模型，因为每种策略对规模限制的反应可能不同，并在第节中描述了结果5.1。

迁移学习我们试验了一种利用迁移学习优势训练神经网络的特殊方法，在这种方法中，神经网络的不同部分在训练过程中学习识别不同的现象97].针对非传统密码策略的一个关键问题是几乎没有训练数据。例如，在我们更大的训练集中，有1.05亿个密码，但是只有260万个密码满足要求最少16个字符的密码策略。训练样本的稀疏性限制了猜测方法对这种非传统策略的有效性。然而，如果对所有密码进行训练，所学习的模型不是最佳的，因为它生成的密码对于我们的目标策略来说是不准确的，即使忽略不满足策略的密码。转移学习让我们训练一个关于所有密码的模型，但只对更长的密码进行猜测。

当使用迁移学习时，首先对训练集中的所有密码训练模型。然后，模型的下层被冻结。最后，只对训练集中合适的密码重新训练该模型

政策。直觉是，模型中的较低层学习关于数据的低级特征(例如,“a”是元音)，而较高层学习关于数据的高级特征(例如，元音通常跟随辅音)。类似地，模型中的较低层可以开发计算密码中字符数的能力，而较高层可以识别密码通常为八个字符长。通过微调更高级别的参数，我们可以利用模型对所有密码的了解，并将其重定向到训练数据稀疏的策略。

训练数据我们用不同的训练数据集进行实验；我们分几节描述两组密码的实验4.1 和5.2，并且在部分中的训练数据中包括自然语言5.1。对于一般的机器学习算法来说，训练数据越多越好，但前提是训练数据与我们测试的密码非常匹配。

3.3 客户端模型

部署客户端(例如，基于浏览器的)密码强度测量工具提出了严峻的挑战。为了使用户体验到的延迟最小化，这些工具应该快速执行，并且通过网络传输尽可能少的数据。高级猜测工具(例如，PCFG、马尔可夫模型以及类似JtR和Hash- cat的工具)在大规模并行服务器上运行，并且需要大约数百兆字节或数千兆字节的磁盘空间。通常，这些模型也需要几个小时或几天来返回强度度量测试的结果，即使在高效计算方面有了新的进展34]，这不适合实时反馈。相比之下，通过结合使用神经网络的大量优化，我们可以构建精确的密码强度测量工具，这些工具足够快以进行实时反馈，并且足够小以包含在网页中。

3.3.1 优化模型尺寸

为了在浏览器中部署我们的原型实现，我们开发了简洁编码的方法。我们利用图形技术为基于浏览器的游戏和可视化编码3D模型[29].我们的编码管道包含四个不同的步骤:权重量化、定点编码、之字形编码和无损压缩。我们的总体策略是发送更少的比特，并利用浏览器实现本身支持的现有无损压缩方法，如gzip压缩[41].我们在一节中描述管道中每一步对压缩的影响5.3。我们还描述了在Bloom filters中对一个简短的密码列表进行编码。

权重量化首先，我们对神经网络的权重进行量化，用较少的位数来表示。我们只发送最重要的数字，而不是发送描述权重的32位浮点数的所有数字。权重量化通常用于减小模型大小，但会增加误差[68].我们在第二节中说明了量化对误码率的影响5.3。我们通过实验发现，将权重量化到三个十进制数字会导致最小的误差。

定点编码其次，我们没有使用浮点编码来表示权重，而是使用了定点编码。由于权重量化步骤，许多权重值被量化为相同的值。定点编码允许我们使用无符号整数而不是网络上的浮点数来更准确地描述量化值:我们可以在内部表示
−5.0和5.0，最小精度为0.005，介于精度为1的1000和1000之间。避免浮点值将节省四个字节。虽然像gzip这样的无损压缩部分减少了对定点编码的需求，但我们发现这种缩放在实践中仍然提供了改进。

之字形编码第三，负值通常在网络上发送更昂贵。为了避免发送负值，我们使用锯齿形编码[8].在ZigZag编码中，通过使用最后一位作为符号位来编码有符号值。因此，0的值被编码为0，但-1的值被编码为1，1被编码为2，-2被编码为3，依此类推。

无损压缩我们使用常规的gzip或deflate编码作为压缩管道的最后阶段。gzip和deflate在模型大小方面产生相似的结果，并且都被浏览器和服务器广泛支持。我们没有考虑其他的压缩工具，比如LZMA，因为浏览器对它们的本地支持并不广泛，尽管它们通常会产生稍微小一点的模型。

Bloom Filter单词表为了增加客户端猜测的成功，我们还存储了一个经常被猜测的密码的单词表。如同先前的工作[89]，我们发现对于某些类型的密码破解方法，预先设置训练密码可以提高猜测的有效性。我们将训练集中最常出现的前两百万个密码存储在一系列压缩的布隆过滤器中[69].

因为布隆过滤器不能将密码映射到破解密码所需的猜测次数，而只能计算

在一个集合中，我们在不同的组中使用多个布隆过滤器:在一个布隆过滤器中，我们包括需要少于10次猜测的密码；在另一个例子中，所有密码都需要少于100次的猜测；诸如此类。在客户端，在每个过滤器中查找一个密码，并分配一个与具有最小密码集的过滤器相对应的猜测号码。这允许我们在不增加布隆过滤器的误差界限的情况下粗略地估计密码的猜测数。为了大幅减少对这些布隆过滤器进行编码所需的位数，我们只发送符合策略要求的密码，并且这些密码的神经网络计算的猜测数与实际猜测数相差三个数量级以上。为了限制整个模型的大小，我们在压缩后将这个单词列表限制在150KB左右。我们发现，要显著提高猜测的成功率，需要更大的空间。

3.3.2 优化延迟

我们依靠预计算和缓存来使我们的原型足够快以进行实时反馈。我们的目标延迟接近100毫秒，因为这是阈值，低于该阈值更新会立即出现[72].

预计算我们预计算猜测数，而不是按需计算猜测数，因为所有按需计算猜测数的方法都太慢，无法提供实时反馈。例如，即使最近计算效率有所提高[34]，我们执行速度最快的模型——马尔可夫模型——需要一个多小时来估计我们的测试集密码的猜测数，而其他方法需要几天。预计算减少了将密码概率转换为猜测数字的延迟:它变成了在客户端的表中的快速查找。

这种类型的预计算的缺点是，由于概率到猜测数映射的量化，猜测数变得不精确。我们通过实验测量(参见第5.3)我们估计的准确性，发现对准确性的影响很低。出于密码强度评估的目的，我们认为这个缺点可以忽略不计，部分原因是结果通常以更大量量化的形式呈现给用户。例如，用户可能被告知他们的密码是“弱”或“强”此外，预计算引入的不准确性可以被调整为导致安全错误，因为任何个人密码的猜测数都可能被低估，但不会被高估。

缓存中间结果我们也缓存中间计算的结果。计算10个字符密码的概率需要11个完整的计算机

神经网络的站，每个字符一个，结束符号一个。通过缓存每个子串的概率，我们显著地加快了候选密码由于在末尾添加或删除字符而改变的速度。我们在第节中通过实验展示了缓存的好处5.3。

多线程在客户端，我们在一个独立于用户界面的线程中运行神经网络计算，以提高用户界面的响应能力。

3.4 履行

我们在Keras库[28]和neo- cortex浏览器实现上的客户端实现[5]的神经网络。我们使用Keras的Theano后端库，它通过使用GPU而不是CPU来更快地训练神经网络[17,18].我们的实现用Python编程语言训练网络和猜测密码。浏览器中的猜测数计算是用JavaScript执行的。我们的模型通常使用三个长短期记忆(LSTM)重现层和两个紧密连接的层，总共五层。在客户端，我们使用WebWorker浏览器API在它们自己的线程中运行神经网络计算[10].

对于某些应用，如密码仪表，保守地估计密码强度是可取的。虽然我们也想尽量减少总体错误，但在客户端，我们更愿意低估密码的抗猜测性，而不是高估它。为了在我们的客户端实现中更严格地低估猜测值，我们计算猜测值时不考虑大小写。我们在实践中发现，我们的模型能够以这种方式计算更严格的低估，而不会高估许多密码的强度。我们不为服务器端模型这样做，因为这些模型用于生成候选密码猜测，而不是估计猜测数。在计算猜测数之后，我们对它们应用一个常数比例因子，作为一个安全参数，以产生更多错误为代价使模型更加保守。我们将在第5节详细讨论这种权衡5.3。

4 测试方法

为了评估我们的神经网络实现，我们将其与多种其他密码破解方法进行了比较，包括PCFGs、马尔可夫模型、JtR和Hash- cat。我们猜测准确性的主要衡量标准是我们的人工密码测试集的可猜测性。个人密码的可猜测性是通过猜测者猜测破解密码的次数来衡量的

密码。我们用两组训练数据和五组测试数据进行实验。对于每组测试数据，我们计算在特定次数的猜测后被破解的密码的百分比。在我们的测试集中，更准确的猜测方法正确猜测更高百分比的密码。

对于概率方法——PCFG、马尔可夫模型和神经网络——我们使用最近的工作，通过蒙特卡罗方法有效地计算猜测数[34].对于蒙特卡罗模拟，我们生成并计算至少一百万个随机密码的概率，以提供准确的估计。虽然这种技术的准确误差很大程度上取决于每种方法、猜测数和个人密码，但通常我们观察到猜测数估计值的10%以下的95%置信区间；误差超过10%的密码只有在超过1018次猜测后才能被猜出。对于所有的蒙特卡罗模拟，为了完整性，我们模拟了多达1025个猜测。这可能是对猜测次数的过高估计，即使是资源充足的攻击者也能够或愿意对一个密码进行猜测。

为了使用基于混乱规则的方法——JtR和Hashcat——计算密码的可猜测性，我们列举了这些方法做出的所有猜测。这提供了准确的猜测数字，但比我们用其他方法模拟的猜测要少。在我们不同的测试集上，基于混淆规则的方法进行了大约1013到1015次猜测。

4.1 培训用数据

为了训练我们的算法，我们混合使用了泄露和破解的密码集。我们认为这是合乎道德的，因为这些密码集已经公开，我们使用它们不会造成额外的伤害。

我们探索两组不同的训练数据。我们称第一组为先前工作使用的密码可猜测性服务(PGS)训练组[89].它包含了你[90]和雅虎！[43]泄露的密码集。对于使用自然语言的猜测方法，它还包括web2列表[11]、谷歌网络语料库[47]，和一个变形字典[78].这组密码共有3300万个密码和590万个自然语言单词。

第二组(PGS++训练集)用额外泄露和破解的密码集扩充了PGS训练集[1,2,3,6,7,9,12,13,14,15,16,20,23,25,42,43,55,56,57,62,63,67,75,77,85,90].对于使用自然语言的方法，我们包括与PGS集相同的自然语言源。这个集合总共有1.05亿个通行词和590万个自然语言词。

4.2 测试数据

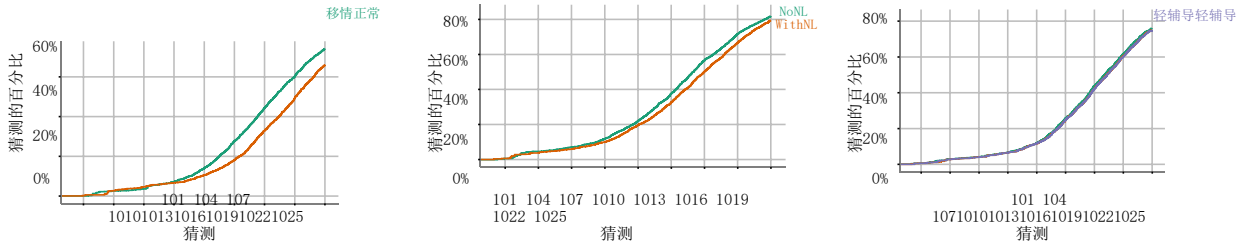
对于我们的测试数据，我们使用了从Mechanical Turk (MTurk)收集的密码，以及从000webhost [40].除了只需要八个字符的常见策略之外，我们还研究了三种不太常见的密码策略，这三种策略显示出更强的抗猜测性[66,80]: 4class8、3class12和1class16，所有这些都在下面描述。我们选择MTurk集合来获取在比泄露数据中所表示的更多的密码里坡下创建的密码。使用MTurk生成的密码被发现与真实世界中的高价值密码相似[38,66].尽管如此，我们还是选择了000webhost漏洞，将我们的结果与最近泄露的密码集中的真实密码进行额外的比较。总之，我们使用了五个测试数据集:

- 1class8:为调查研究收集了3，062个超过8个字符的密码[59]
- 1class16:为一项研究收集了2054个超过16个字符的密码[59]
- 3class12: 990密码必须包含至少三个字符类别(大写字母、小写字母、符号、数字),并且为研究收集的长度至少为12个字符[80]
- 4class8: 2，997个密码，必须包含所有四个字符类别，并且为研究收集的长度至少为八个字符[66]
- webhost:从000webhost漏洞中包含至少8个字符的密码中随机抽取30，000个密码[40]

4.3 猜测配置

PCFGWe使用了带有终端平滑和混合结构的PCFG版本[60]，并在训练数据中包括自然语言词典，对每个单词进行加权以计为通行单词的十分之一。我们还分离了对结构和终端的训练，并且只对符合目标策略的密码训练结构。此方法不会生成与目标策略不匹配的密码。

对于PCFG来说，蒙特卡罗方法不能为具有相同概率的密码估计唯一的猜测数字。这种现象在具有锯齿状边缘的蒙特卡洛图中表现出来，其中许多不同的密码被赋予相同的猜测数字(例如，在图5c 1023年前)。我们假设最佳攻击者可以以任何顺序排列这些猜测，因为根据模型，它们都具有相同的可能性。因此，我们给所有这些猜测分配最低的猜测数。这是对PCFG猜测有效性的严格高估，但实际上并没有改变结果。



(a) 移情学习 (b) 自然语言词典 (c) 当家庭教师

图2:神经网络的替代训练方法。x轴代表对数标度的猜测次数。y轴显示猜测的1类16密码的相应百分比。在(b)中，WithNL是用自然语言字典训练的神经网络，NoNL是不用自然语言字典训练的神经网络。

我们训练了4、5和6克的马尔可夫模型。先前的工作发现6-gram模型和0.01的加法平滑是大多数密码集的有效配置[65].我们的结果是一致的，我们在测试中使用了6克模型和附加平滑。我们丢弃与目标策略不匹配的猜测。

我们使用流行的破解工具Hashcat和John the Ripper (JtR)来计算猜测数字。对于Hashcat，我们使用软件中包含的best64和gen2规则集[83].对于JtR，我们使用SpiderLabs的篡改规则[86].我们选择这些规则集是因为先前的工作发现它们在猜测通用密码时是有效的[89].为了创建每个工具的输入，我们按照频率降序排列了各自的训练集。对于JtR，我们删除与目标策略不匹配的猜测。然而，对于Hashcat，我们没有这样做，因为Hash- cat的GPU实现可能会遭受重大的性能损失。我们认为这模拟了一个真实世界的场景，在这个场景中也会产生这种惩罚。

5 估价

我们进行了一系列实验来调整我们的神经网络的训练，并将它们与现有的猜测方法进行比较。在截面中5.1，我们描述了通过使用不同的训练方法来优化神经网络的猜测有效性的实验。选择这些实验主要是为了指导我们关于模型参数的决策，以及沿着我们在第节中描述的设计空间进行训练3.2，包括训练方法、模型大小、训练数据和网络架构。在截面中5.2，我们比较了神经网络的猜测与其他猜测算法的有效性。最后，在小节中5.3中，我们描述了我们的浏览器实现的有效性、速度和大小，并将它与其他浏览器密码测量工具进行了比较。

5.1 训练神经网络

我们进行了实验，探索如何调整神经网络训练，包括修改网络大小，使用子词模型，包括训练中的自然语言词典，以及探索替代架构。我们并不认为这些实验是对太空的彻底探索。事实上，改善神经网络是一个活跃的研究领域。

移情学习我们发现移情学习训练，在第一节中描述3.2，提高猜测效率。数字2a以对数标度显示迁移学习的效果。例如，在1015次猜测中，22%的测试集通过迁移学习被猜到，而没有迁移学习的只有15%。我们使用一个16 MB的网络，对我们的1class16密码执行了这个实验，因为它们与我们的大多数训练集特别不同。在这里，迁移学习主要在较高的猜测数上改进密码猜测。

包括自然语言词典我们实验了在神经网络训练数据中包括自然语言词典，假设这样做会提高猜测的有效性。我们用1类16个密码进行了这个实验，因为它们特别可能从自然语言词典的训练中受益91].使用迁移学习方法对长密码训练有自然语言数据和无自然语言数据的网络。第一批训练数据中包含了自然语言。数字2b 表明，与我们的假设相反，对自然语言的训练降低了神经网络的猜测效率。我们认为，与PCFG等其他方法相比，神经网络并不受益于自然语言，因为这种训练方法并不区分自然语言字典和密码训练。然而，可以用自然语言以其他方式增强训练数据，也许会产生更好的结果。

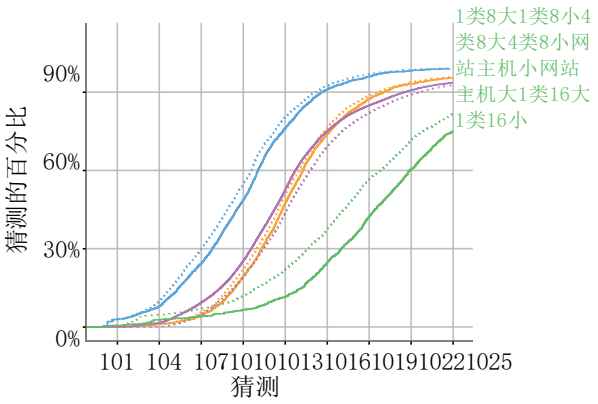


图3:神经网络大小和密码猜测性。
虚线是大网络；实线是小网络。

密码标记化我们发现，使用混合的子单词级密码模型不会显著提高低猜测数的猜测性能。混合模型可能以多种不同的方式表示同一个单词。例如，该模型可以将一个单词捕获为一个标记“pass”，或者捕获为字母“p”、“a”、“s”、“s”。因为蒙特卡罗模拟假设密码是唯一表示的，而不是使用蒙特卡罗方法来估计猜测数，所以我们通过枚举最可能的107次猜测来计算猜测数。然而，在如此低的猜测次数下，我们显示这种标记化只有很小的影响，如图所示4b。我们对长密码进行了这个实验，因为我们相信它们会从标记化中受益最多。这个实验表明，可能有早期的好处，但在其他方面，模型的学习是相似的。我们认为这个结果是探索性的，因为我们的猜测截止值较低，也因为调整标记化的其他选项可以产生更好的结果。

模型大小我们发现，至少对于一些密码集，神经网络模型可以比其他模型小几个数量级，而对猜测有效性几乎没有影响。我们测试了以下两个模型大小如何影响猜测的有效性:一个具有1，000个LSTM单元或15，700，675个参数的大模型使用60 MB，一个具有200个LSTM单元或682，851个参数的小模型使用2.7 MB。

这些实验的结果如图所示3。对于1class8和4class8策略，减小模型大小的影响很小，但很明显。但是，对于1class16密码来说，这种影响更加明显。我们将较长和较短里坡系统在模型大小方面的差异归因于这些策略之间密码组成的根本差异。长密码更类似于英语短语，与短密码相比，长密码建模可能需要更多的参数，因此需要更大的网络

密码。webhost测试集是唯一一个较大模型表现较差的测试集。我们认为这是由于我们用于该模型的特定训练数据缺乏适用性。我们将在第节中详细讨论训练数据的差异5.2。

辅导网络为了提高我们的小模型猜测长密码的效率，我们试图用从更大的网络中随机生成的密码来辅导我们的小神经网络。虽然这对于轻度辅导有轻微的积极影响，但在随机数据与真实数据的比例大约为1比2时，这种影响似乎并不明显

扩大到更重的辅导。数字2c使用辅导时，无论猜测的准确性是轻还是重，都显示出最小的差异。

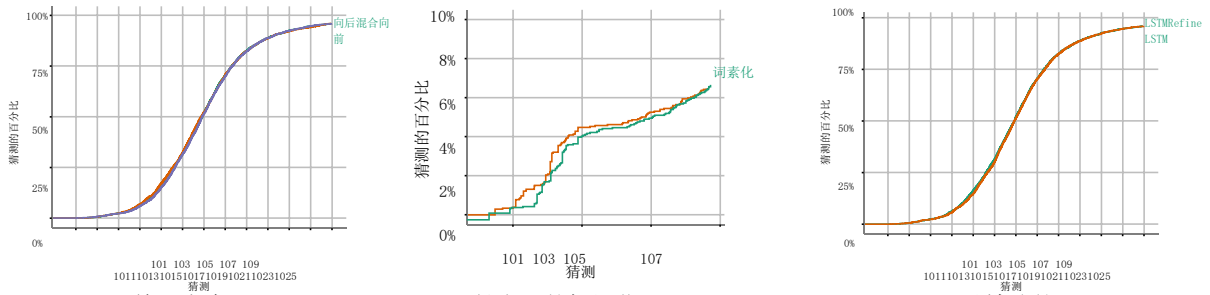
向后与向前训练，如第节所述3.2在神经网络的某些应用中，向后处理输入比向前处理更有效48].我们进行了向前、向后猜测密码的实验，并使用了一种混合方法，其中一半网络向前检查密码，另一半向后检查密码。我们观察到总体上只有微小的差异。在最大差异点，接近109次猜测，混合方法猜测了测试集的17.2%，向后猜测了测试集的16.4%，向前猜测了测试集的15.1%。数字4a 展示了这个实验的结果。由于混合方法增加了训练所需的时间量，而准确性只有很小的提高，因此对于其他实验，我们使用反向训练。

递归结构我们实验了两种不同类型的递归神经网络结构:长短期记忆(LSTM)模型[54]以及对LSTM车型的改进[58].我们发现，这种选择对网络的总产出几乎没有影响，改进的LSTM模型稍微更准确，如图所示4c。

5.2 猜测有效性

与其他个人密码猜测方法相比，我们发现神经网络更擅长在猜测次数更多时猜测密码，以及在获取更复杂或更长的密码策略时，如我们的4class8、1class16和3class12数据集。例如，如图所示5b，神经网络通过1015次猜测猜出了4class8密码的70%，而第二好的猜测方法猜出了57%。

模型在猜测特定密码的有效程度上有所不同。明格斯，如图所示5代表一种理想化的猜测方法，在这种方法中，只要密码被我们的任何



(a) 培训方向。 (b) 长密码的标记化。 (c) LSTM对精致的LSTM。

图4:附加的调优实验。我们的LSTM实验测试了16M参数的复杂密码。我们发现性能差异很小。我们的记号化实验检查了长密码。我们关于训练方向的实验包括在复杂密码上使用16M参数进行向后、向前以及向前和向后的训练。

猜测方法，包括神经网络、马尔可夫模型、PCFG、JtR和Hashcat。MinGuess优于神经网络表明，尽管事实上神经网络通常单独优于其他模型，但使用多种猜测方法仍应优于使用任何单一猜测方法进行精确的强度估计。

对于我们测试的所有密码集，神经网络从大约1010次猜测开始就优于其他模型，并在此之前匹配或击败了其他最有效的方法。数字5-6显示用PGS数据集训练的不同猜测方法的性能，并给出图表7-8展示了用PGS++数据集训练的相同猜测方法。这两个数据集在第节中有更详细的描述4.1。在这一部分，我们使用了我们的1570万参数的大型神经网络，在两个训练集上进行迁移学习训练。虽然性能因猜测方法和训练集而异，但总的来说，我们发现神经网络在高猜测数和跨策略时的性能适用于两组训练数据，只有一个例外，如下所述。因为这些结果适用于多个训练和测试集，所以我们假设神经网络在猜测我们没有测试的许多策略下创建的密码时也表现良好。

在使用PGS++训练数据的webhost测试集中，神经网络的表现比其他方法差。对于webhost来说，所有使用PGS++数据集的猜测方法都不如PGS数据集有效，尽管有些方法，比如PCFG，只是稍微有点影响。因为所有方法的性能都更差，而且当使用PGS训练数据时，神经网络比其他方法做得更好-类似于其他测试集-我们认为PGS++训练数据对这个测试集特别无效。如图所示3 显示，这是唯一一个较小的神经网络表现明显优于较大的神经网络的数据集，这表明较大的神经网络模型更严格地适合低质量的数据，这限制了较大网络的概括能力。

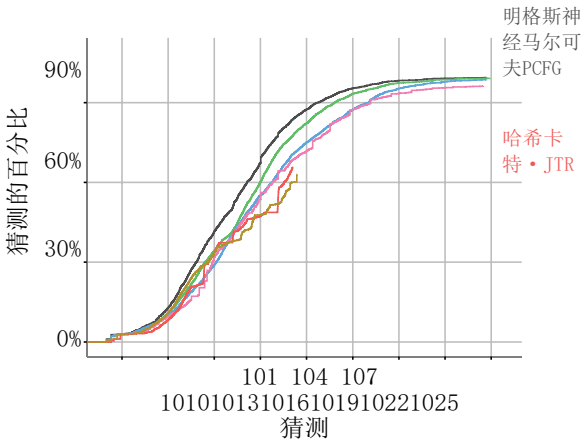
定性地说，我们的神经网络实现在其他方法之前猜测的密码类型是与训练集中的密码不同的新密码。我们的神经网络实现很晚才猜到的密码类型，但通过其他方法很容易猜到的密码类型通常与自然语言词典中的单词相似，或者在训练数据中出现频率很低。

资源需求通常，PCFGs需要最多的磁盘、内存和计算资源。我们的PCFG实现将其语法存储在4.7GB的磁盘空间中。马尔可夫模型是我们的第二大实现，需要1.1GB的磁盘空间。Hashcat和JtR不需要大量的空间来存储它们的规则，但是需要存储整个训练集，756MB。相比之下，我们的服务器端神经网络只需要60MB的磁盘空间。虽然60MB仍然比不压缩的情况下可以有效传输到客户机的容量大，但与其他型号相比，这是一个实质性的改进。

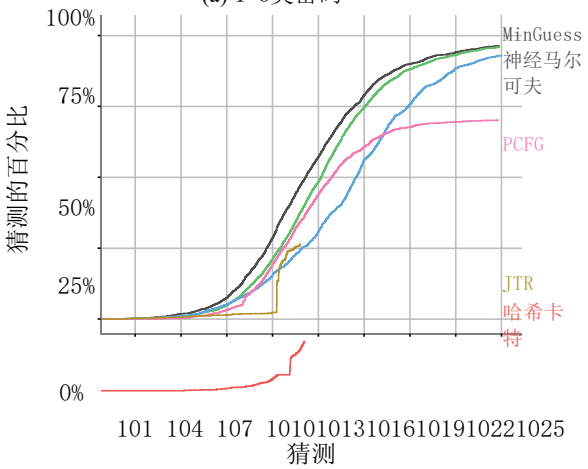
5.3 浏览器实现

虽然有效的模型可以适应60MB，但对于浏览器中的实时密码反馈来说，这仍然太大了。在这一节中，我们将评估第节中讨论的压缩神经网络模型的技术3.3通过比较压缩模型与所有服务器端模型的猜测效率，我们的大型神经网络、PCFG、马尔可夫模型、JtR和Hashcat。

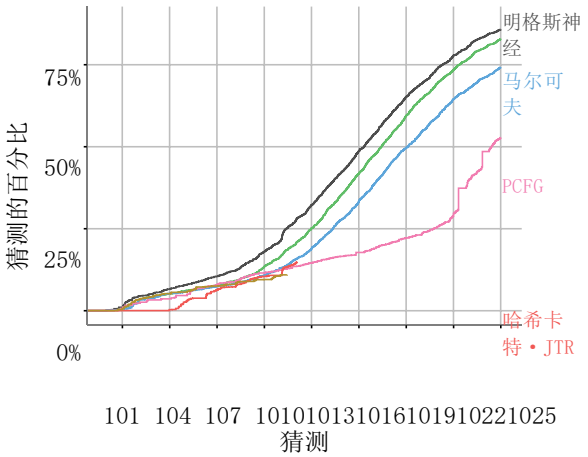
模型编码我们的主要尺寸度量是gzip- ed模型尺寸。我们的压缩阶段使用JSON for- mat，因为它支持JavaScript平台。我们探索了使用MsgPack二进制格式[4]，但发现在gzip压缩后，编码大小没有好处，解码速度也有小的缺点。不同流水线阶段对压缩的影响如表所示1。



(a) 1 8类密码

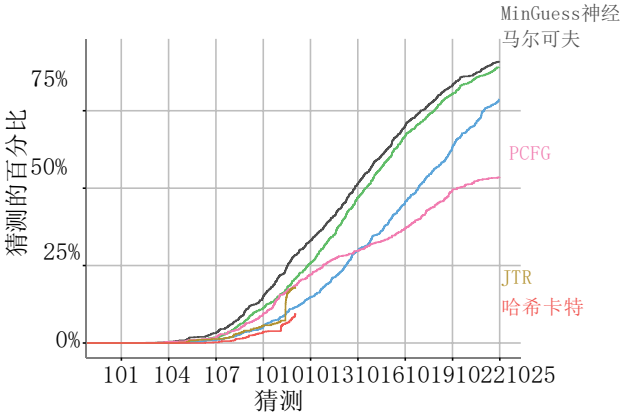


(b) 4类8密码

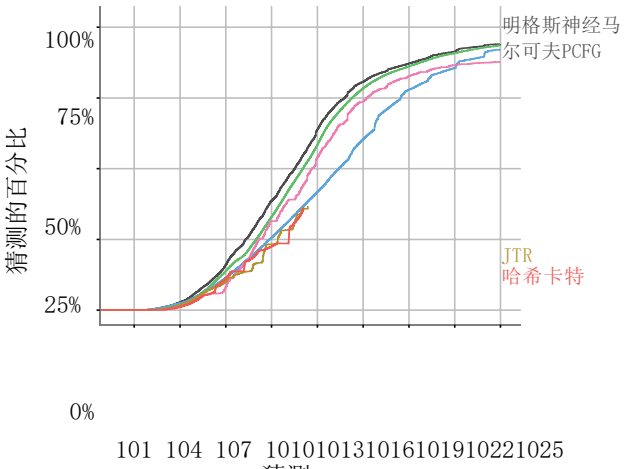


(c) 1 16类密码

图5:使用PGS数据集，不同猜测方法的密码集的可猜测性。MinGuess代表任何方法的最小猜测次数。y轴以不同的方式缩放，以最好地显示比较性能。



(a) 3类12密码



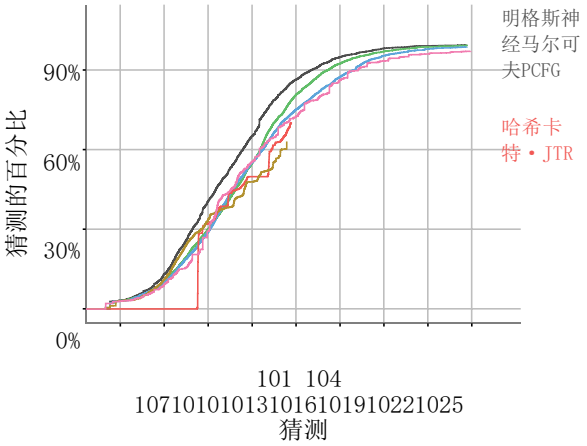
(b) 网站主机密码

图6: **不同密码设置的可猜测性使用PGS数据集的猜测方法(续)。**

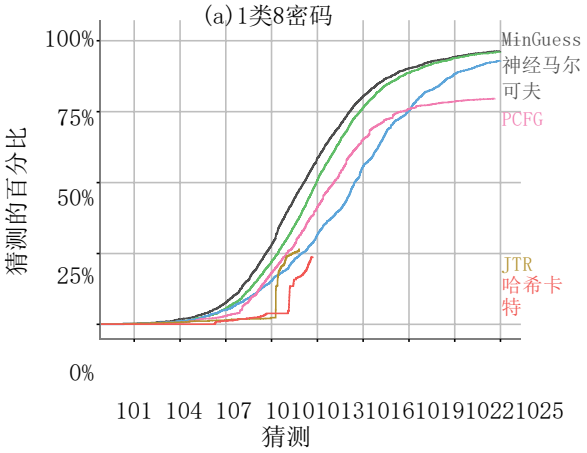
我们还在浏览器实现中量化了模型的参数，以进一步减小模型的大小。权重和曲线量化都是有损操作，其对猜测的影响如图所示9。曲线量化表现为猜测曲线的锯齿形状，但是猜测曲线的整体形状基本不变。

权重和概率曲线量化由于当前从概率计算猜测数的方法太慢，需要几个小时或几天才能返回结果，我们预先计算从密码概率到猜测数的映射，并将该映射发送给客户端，如第节所述3.3.2。这种映射可以通过量化猜测概率曲线来有效地编码。量化曲线会导致安全错误，也就是说，我们低估了密码的强度。

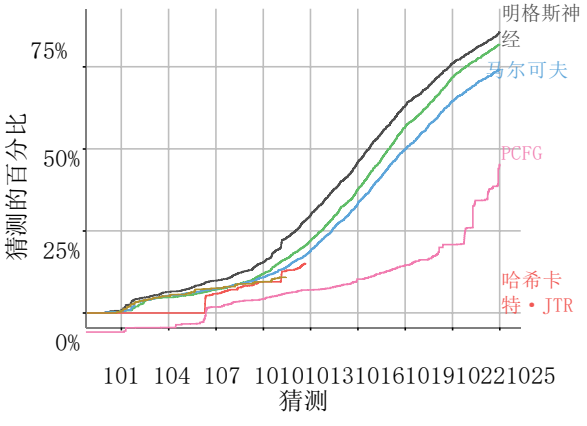
评估反馈速度尽管计算密码的可猜测性需要大量的计算，但我们的原型实现足够高效，可以提供实时的用户反馈。一般来说，快于100毫秒的反馈被认为是即时的[72];因此，这是我们的基准。我们执行了两个测试来测量计算猜测数字的速度:第一个测试使用半缓存的密码来测量产生猜测数字的时间；第二个计算每个密码的总时间。半缓存测试测量在密码末尾添加字符时计算猜测数字的时间。我们认为这代表了用户在实践中的体验，因为用户通常通过一个字符一个字符地键入来创建密码。



(a) 1类8密码



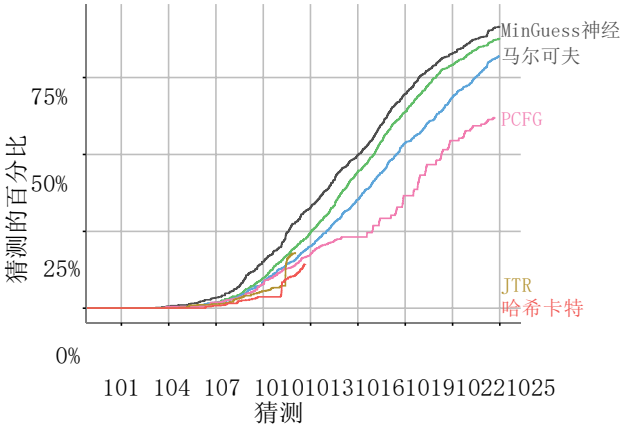
(b) 4类8密码



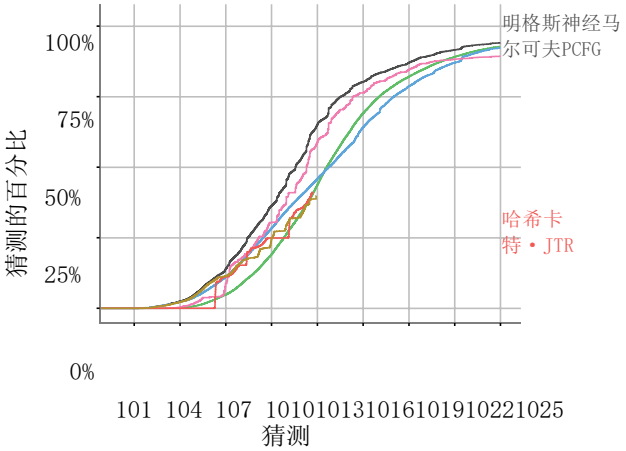
(c) 1 16类密码

图7:不同密码集的可猜测性
使用PGS++数据集的猜测方法。明格斯代表任何方法的最小猜测次数。

我们在运行OSX的笔记本电脑上用2.7 GHz i7处理器，使用Chrome网络浏览器(版本48)。在这些测试中，我们从1class8训练集中随机选择了500个密码。在半缓存测试中，计算一个猜测数的平均时间是17ms(stdev:4 ms)；在全密码测试



(a) 3类12密码



(b) 网站主机密码

图8: **不同密码设置的可猜测性使用PGS++数据集的猜测方法(续)。**

与其他密码计量器相比，我们将我们的客户端神经网络实现的准确性与其他客户端密码强度估计器进行了比较。密码强度的近似值可能被低估或高估。我们称过高估计密码强度为不安全错误，因为它们表示密码比实际更难猜测。我们表明，我们的计量器可以更精确地测量密码的抗猜测性，不安全密码的抗猜测性减少了一半

中，平均时间是124毫秒(stdev: 48毫秒)。然而，半缓存测试和非缓存测试的执行速度都足够快，可以向用户提供快速的反馈。

ror是现有的客户端模型，基于启发式算法。在这一节中，我们的基本事实是理想化的MinGuess方法，在第2节中有所描述5.2。

先前的工作发现几乎所有的主动密码强度估计器都是不一致的，并且对密码的抗猜测性估计很差[33].最有希望的评估者是Dropbox的zxcvbn测量仪94,95]，它依赖手工制作的试探法、统计方法和明文字典作为训练数据来估计猜测数字。值得注意的是，这些纯文本字典与用于我们训练数据的字典不同，这限制了我们从这些比较中全面归纳的能力。探索配置zxcvbn的其他方法超出了本评估的范围。我们

管道阶段	尺寸gzip-ed	尺寸原始JSON格式
	6.9M	2.4M
量化	4.1M	716K
定点	3.1M	668K
之字形编码	3.0M	664K
移除空间	2.4M	640K

表1:不同管道阶段对模型大小的影响。此表显示了以1class8密码策略为目标的小型模型，具有682，851个参数。每一级包括前一级，例如定点级包括量化级。我们在最高压缩级别使用gzip。

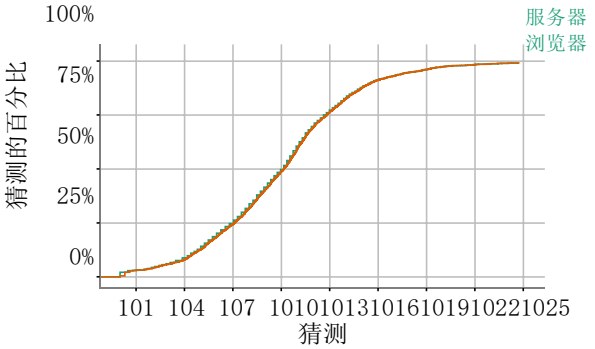


图9:具有权重和曲线量化的压缩浏览器神经网络与未量化网络的比较。Browser是我们的带有权重和曲线量化的浏览器网络。服务器是同样的没有权重和曲线量化的小神经网络。

将我们的结果与zxcvbn和Yahoo！me- ter，这是一个使用非常简单的试探法来估计密码强度的例子。

雅虎！米不产生猜测数字，但箱密码为最弱，较弱，弱，强，强。我们忽略二进制名称的语义值，并检查计量员将具有不同猜测数字(由所有猜测方法的猜测值计算)的密码分类到五个二进制中的准确性。比较雅虎！米到我们的最小猜测数(表2)，我们取每个箱(例如“较弱的”箱)的中间实际猜测数

		完全不安全	
1类8	神经网络	1311	164
	zxcvbn	1331	270
	雅虎！	1900	984
4类8	zxcvbn	1853	231
	神经网络	1826	
	雅虎！	1328	647

表2:不同客户端仪表的总错误分类数和不安全错误分类数。因为雅虎！仪表提供不同的宁滨，我们对其输出进行预处理，以便进行更公平的比较，如第节所述5.3。

然后将每个密码的最小猜测数字映射到最接近对数刻度的箱子。例如，在雅虎！米，猜测数5.4 104是“较弱”bin的中值；任何接近于5.4 104而不是对数标度上其他箱的中间值的密码，我们都认为属于“较弱”的箱。我们希望这是对雅虎准确性的高估。米。尽管如此，我们的工作和以前的工作[33]找到雅虎！米没有其他方法精确，包括zxcvbn米。

我们发现我们的客户端神经网络方法比我们测试的其他方法更准确

不安全错误和可比的安全错误最多减少两倍，如图所示10 和桌子2。这里，我们使用我们的神经网络仪表实现，并进行第节所述的调谐3.4。我们使用客户端布隆过滤器执行了1class8测试，如第节所述3.3.1，而4class8测试没有使用布隆过滤器，因为它不会显著影响准确性。两个测试都将网络输出缩小300倍，并忽略大小写以给出更保守的猜测数字。我们选择比例因子来调整网络-

努力使尽可能多的安全错误zxcvbn。此外，我们发现，与我们的神经网络实现相比，zxcvbn仪表的误差通常处于非常低的猜测值，这可能特别不安全。例如，对于10，000个最可能的密码，zxcvbn产生84个不安全的错误，而我们的神经网络只产生11个不安全的错误。

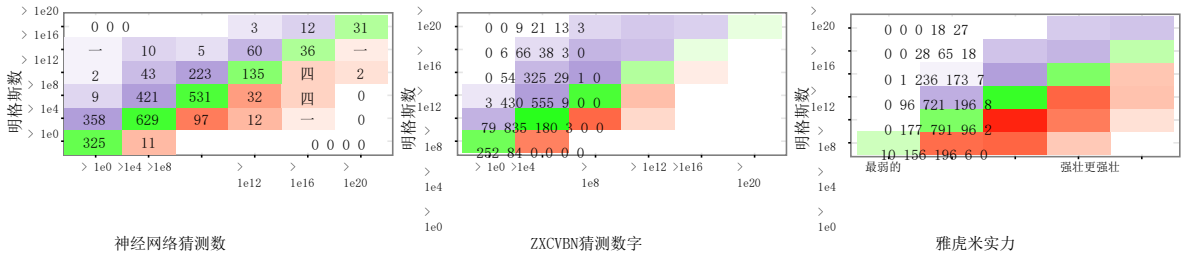
除了更准确之外，我们认为神经网络方法更容易应用于其他密码策略。现有的最好的计量器zxcvbn是手工制作的，针对一个特定的密码策略。另一方面，神经网络只需通过再训练就可以轻松地重定向到其他策略。

6 结论

本文描述了如何使用神经网络来模拟人类选择的密码和测量密码强度。我们展示了如何建立和训练在效率和效果方面优于最先进的密码猜测方法的神经网络

可以构建提供良好的密码强度测量的客户端密码计量器。

针对密码猜测调整神经网络，以及开发精确的客户端密码强度指标，这两者都是肥沃的研究土壤。先前的工作已经使用神经网络来学习更大的模型集合的输出[24]并获得了比我们的网更好的结果



(a) 客户端神经网络 (b) zxcvbn (c) 雅虎!

图10:客户端猜测值与所有服务器端方法的最小猜测值的比较。箱中的数字代表该箱中密码的数量。例如，神经网络将358个密码评定为猜测100到104次，而服务器端方法将它们评定为猜测104到108次。测试密码是我们的1class8集合。雅虎！米不提供猜测数字，因此，有一个不同的x轴。红色阴影表示力量的高估，紫色阴影表示力量的低估，绿色阴影表示力量的准确估计。颜色强度随着箱子中密码的数量而增加。

工作辅导(部分5.1).其他工作通过使用矩阵分解或专门的训练方法实现了比我们更高的神经网络压缩比[51,96].利用自然语言、符号化模型或其他神经网络架构的进一步实验可能会让密码更有效地被猜到。虽然我们根据猜测的有效性来衡量客户端的强度指标，但剩下的一个挑战是提供用户可理解的建议来改进密码创建过程中的密码。

7 承认

我们要感谢Mahmood Sharif参与关于神经网络的讨论，并感谢Dan Wheeler的反馈。这项工作得到了PNC金融服务创新中心、微软研究院和John & Claire Bertucci的部分资助。

参考

[1] CSDN密码泄露。http://thepasswordproject.com/泄露的密码列表和字典。

[2] 信仰作家泄露。https://wiki.skullsecurity.org/密码#泄露的密码。

[3] Hak5泄露。https://wiki.skullsecurity.org/密码#泄露的密码。

[4] Msgpack:它像JSON，但是又快又小。http://msgpack.org/index.html。

[5] 大脑皮层Github知识库。https://github.com/scienceai/neocortex。

[6] Perl monks密码泄露。http://news.softpedia.com/news/PerlMonks-ZF0-Hack-Has-Wider-Implications-118225.shtml。

[7] Phpbb密码泄露。https://wiki.skullsecurity.org/Passwords。

[8] 协议缓冲区编码。https://developers.google.com/protocol-buffers/docs/encoding。

[9] 斯特拉特福 泄漏。http://thepasswordproject.com/泄露的密码列表和字典。

[10] 使用网络工作者。https://developer.mozilla.org/en-US/docs/Web/API/Web Workers API/使用Web Workers。访问时间:2016年2月。

[11] 英语单词的“web2”文件。http://www.bee-man.us/computer/grep/grep.htm#web2, 2004.

[12] 密码泄露:Elitehacker。https://wiki.skullsecurity.org/Passwords, 2009.

[13] 密码泄露:启示a。https://wiki.skullsecurity.org/Passwords, 2010.

[14] Specialforces.com密码泄露。http://www.databreaches.net/update-specialforces-com-hackers-acquired-8000-credit-card-numbers/, 2011.

[15] YouPorn密码泄露,2012年。http://thepasswordproject.com/泄露的密码列表和字典。

[16] 2013年WOM维加斯密码泄露。https://www.hackread.com/wom-vegas-breached-10000-user-accounts-leaked-by-darkweb-goons/。

[17] BASTIEN, f ., LAMBLIN, p ., PASCANU, r ., BERGSTRA, j ., GOODFELLOW, I. J ., BERGERON, a ., BOUCHARD, n ., BENGIO, Y. Theano:新特性和速度改进。进行中。NIPS 2012深度学习研讨会(2012)。

[18] 伯格斯特拉,j .,布鲁勒,o .,巴斯蒂安,f .,兰布林,页(page的缩写)、PASCANU,r .、DESJARDINS,g .、TURIAN,j .、WARDE- FARLEY,d .和BENGIO,Y. Theano:一个CPU和GPU数学表达式编译器。进行中。SciPy (2010年)。

[19] 通过主动密码检查提高系统安全性。计算机与安全14,3 (1995),233–249。

[20] 高客黑客:一百万个密码是如何丢失的。浅蓝色Touchpaper博客,2010年12月。http://www.lightbluetouchpaper.org/2010/12/15/the-gawker-hack-how-a-million-passwords-were-lost/。

[21] 猜测的科学:分析7000万密码的匿名语料库。进行中。IEEE Symp。安全与隐私(2012年)。

[22] 个人密码强度的统计度量。进行中。WPS (2012)。

[23] BRODKIN, LinkedIn黑客攻击的10个左右最糟糕的密码。Ars技术公司,2012年6月6日。http://arstechnica.com/security/2012/06/10-or-so-of-the-worst-passwords-exposed-by-the-linkedin-hack/。

[24] c .bucilua˘、r .卡鲁阿纳和a .尼古列斯库-米齐尔的模型压缩。进行中。KDD (2006年)。

[25] 伯内特，m。Xato密码设置。https://xato.net/。

[26] 来自马尔可夫模型的自适应密码强度计。进行中。NDSS (2012年)。

[27] Kickstarter黑客攻击中泄露密码和电子邮件地址。美国广播公司新闻，2014年2月17日。http://abcnews.go.com/Technology/passwords-email-addresses-leaked-kickstarter-hack/story?id=22553952。

[28] CHOLLET，F. Keras Github知识库。https://github.com/fchollet/keras。

[29] 网络GL模型:端到端。在OpenGL Insights中。2012.

[30] CIARAMELLA，a .，D'ARCO，p .，DE SANTIS，a .，GALDI，c .，TAGLIAFERRI，r.《用于主动密码检查的神经网络技术》.电气和电子工程师协会TDSC 3，4 (2006)，327 - 339。

[31] CLERCQ，J. D .重置KRBTGT活动目录帐户的密码，2014年。http://windowsitpro.com/security/resetting-password-krbtgt-active-directory-account。

[32] DAS，a .，BONNEAU，j .，CAESAR，m .，BORISOV，n .，和WANG，x .进行中。NDSS (2014年)。

[33] 从非常弱到非常强:分析密码强度表。进行中。NDSS (2014年)。

[34] 蒙特卡洛强度评估:快速可靠的密码检查。进行中。CCS (2015年)。

[35] DELL 'AMICO，p . michiar di，和y . ROUDIER，Password strength:一项实证分析。进行中。INFOCOM (2010年)。

[36] 重复登录使得2000万阿里巴巴账户受到攻击。ZDNet，2016年2月5日。http://www.zdnet.com/article/login-duplication-allows-20m-alibaba-accounts-to-be-attacked/。

[37] DU RMUTH，m .，ANGELSTORF，f .，CASTELLUCCIA，c .，PER- ITO，d .，和CHAABANE，A. OMEN:使用有序马尔可夫枚举器进行更快的密码猜测。进行中。埃索斯(2015年)。

[38] FAHL，s，哈巴赫，m，ACAR，y，和史密斯，m。进行中。汤(2013)。

[39] 弗洛里·NCIO博士、赫利博士和范·奥尔肖特教授:《互联网密码研究管理指南》。进行中。USENIX丽莎(2014)。

[40] 1300万个密码似乎已经从这个免费的网络主机中泄露。福布斯，2015年10月28日。http://www.forbes.com/sites/thomasbrewster/2015/10/28/000webhost-database-leak/。

[41] 杰.盖利-吉兹普。http://www.gzip.org/。

[42] 10，000个Hotmail密码被秘密泄露到网上。登记册，2009年10月5日。http://www.theregister.co.uk/2009/10/05/hotmail密码泄露/。

[43] 黑客暴露了453，000个合法从雅虎服务获取的凭证。Ars技术公司，2012年7月12日。http://arstechnica.com/security/2012/07/yahoo-service-hacked/。

[44] 《黑客解剖:黑客如何像“qeadzcwrsfxv1331”一样洗劫密码》。Ars技术公司，2013年5月27日。http://arstechnica.com/security/2013/05/how-crackers-make-minced-meat-out-of-your-passwords/。

[45] 为什么LivingSocial的5000万次密码泄露比你想象的要严重。Ars技术公司，2013年4月27日。http://arstechnica.com/security/2013/04/why-livingsocials-50-million-password-breach-is-graver-than-you-may-think/。

[46] 古丁，d。曾经被视为防弹，1100万+阿什利麦迪逊密码已经破解。Ars Technica，2015年9月10日。http://arstechnica.com/security/2015/09/once-seen-as-bulletproof-11-million-ashley-madison-passwords-already-cracked/。

[47] 谷歌。Web 1T 5克版本1，2006年。http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13。

[48] 使用递归神经网络的监督序列标记。斯普林格，2012。

[49] 用循环神经网络产生序列。arXiv预印本arXiv:1308.0850，2013。

[50] 变态用来从苹果iCloud上窃取裸照的警察工具。连线，2014年9月2日。https://www.wired.com/2014/09/eppb-icloud/。

[51] 深度神经网络压缩流水线:剪枝，量化，霍夫曼编码。arXiv预印本arXiv:1510.00149，2015。

[52] 五个最好的密码管理器。LifeHacker，2015年1月11日。http://lifehacker.com/5529133/。

[53] 一项关于密码持久性的研究议程。IEEE安全与隐私杂志10，1(2012年1月)，28－36。

[54] HOCHREITER，s．和SCHMIDHUBER，j．长短期记忆。神经计算9，8（1997），1735－1780。

[55] 索尼密码的简要分析。http://www.troyhunt.com/2011/06/brief-sony-password-analysis.html，2011.

[56] 澳大利亚广播公司黑掉了网上发布的近5万份用户证书，其中一半在45秒内被破解。Techgeek，2013年2月27日。http://techgeek.com.au/2013/02/27/abc-australia-hacked-nearly-50000-user-credentials-posted-online/。

[57] 约翰斯通，L. 9,885个用户账户被匿名从美国的调解人那里泄露。http://www.cyberwarnews.info/2013/07/24/9885-user-accounts-leaked-from-intercessors-for-america-by-anonymous/，2013.

[58] 对循环网络架构的经验探索。进行中。ICML（2015年）。

[59] KELLEY，P. G．，KOMANDURI，s．，MAZUREK，M. L．，SHAY，r．，VIDAS，t．，BAUER，l．，CHRISTIN，n．，CRANOR，L. F．，LOPEZ，J. Guess again(一次又一次):通过模拟密码破解算法测量密码强度。进行中。IEEE Symp。安全与隐私(2012年)。

[60] 用有限样本模拟对手评估密码强度。2016年卡内基梅隆大学博士论文。

[61] KOMANDURI，s．、SHAY，r．、CRANOR，L. F．、HERLEY，c．和SCHECHTER，s.《心灵感应词:通过读取用户的思想来防止弱密码》。进行中。USENIX安全(2014年)。

[62] KREBS，b．欺诈bazaar carders.cc被黑。http://krebsonsecurity.com/2010/05/fraud-bazaar-carders-cc-hacked/。

[63] 黑客发布了他们声称的超过21，000个属于分流客户的用户账户的详细信息。ZDNet，2012年7月13日。http://www.zdnet.com/article/over-21000-plain-text-passwords-stolen-from-billabong/。

[64] 哈比语音识别系统。卡内基梅隆大学博士论文，1976年。

[65] 马军，杨文伟，罗，李，n。概率密码模型的研究。进行中。IEEE Symp。安全与隐私(2014年)。

[66] MAZUREK，M. L .、KOMANDURI，s .、VIDAS，t .、BAUER，l .、CHRISTIN，n .、CRANOR，L. F .、KELLEY，P. G .、SHAY，r .和UR，b .测量整个大学的密码可猜测性。进行中。CCS (2013年)。

[67] Twitter漏洞泄露了25万用户的电子邮件和密码。登记册，2013年2月2日。

[68] MIKOLOV，t .，SUTSKEVER，I .，DEORAS，a .，LE，H.-S .，KOM- BRINK，s .，和CERNOCKY，j .，神经网络的子词语言建模。预印本(http://www.fit.vutbr.cz/~imikolov/rnnlm/char.pdf)，2012.

[69] 压缩布鲁姆过滤器。IEEE/ACM网络汇刊(TON) 10，5 (2002)，604－612。

[70] NARAYANAN，a .和SHMATIKOV，v .使用时空折衷对密码的快速字典攻击。进行中。CCS (2005年)。

[71] 使用神经网络破解密码。博客帖子。https://0day.work/using-neural-networks-for-password-cracking/，2016.

[72] 可用性工程，第125184069卷。波士顿学术出版社，1993年。

[73] Adobe黑客攻击比以前想象的要大。纽约时报Bits博客，2013年10月29日。http://bits.blogs.nytimes.com/2013/10/29/adobe-online-attack-was-bigger-than-previously-thought/。

[74] 《开膛手约翰》。http://www.openwall.com/john/，1996-.

[75] PROTALINSKI，hack后824万Gamigo密码泄露。ZDNet，2012年7月23日。http://www.zdnet.com/article/8-24-million-gamigo-passwords-leaked-after-hack/。

[76] RAGAN，S. Mozilla的bug跟踪门户遭到破坏，重复使用密码是罪魁祸首。CSO，2015年9月4日。http://www.csoonline.com/article/2980758/。

[77] 我的空间密码没那么蠢。http://www.wired.com/politics/security/commentary/securitymatters/2006/12/72300，2006.

[78] 皱眉。面向拼写检查的单词表。http://wordlist.sourceforge.net，2015.

[79] SHAY，r .，BAUER，l .，CHRISTIN，n .，CRANOR，L. F .，FOR- GET，A .，KOMANDURI，s .，MAZUREK，M. L .，MELICHER，w .，SEGRETI，S. M .，UR，b .一勺糖？指导和反馈对密码创建行为的影响。进行中。迟(2015)。

[80] SHAY，r .，KOMANDURI，s .，DURITY，A. L .，HUH，P. S .，MAZUREK，M. L .，SEGRETI，S. M .，UR，b .，BAUER，l .，CHRISTIN，n .，CRANOR，L. F .长密码能安全可用吗？进行中。迟(2014)。

[81] 宋，D. X，瓦格纳，d，田，x。对SSH的击键和定时攻击的定时分析。进行中。USENIX安全研讨会(2001年)。

[82] 蜘蛛橡树。零知识云解决方案。https://spideroak.com/，2016.

[83] 哈什卡特•施陶贝。https://hashcat.net/oclhashcat/，2009-.

[84] 用递归神经网络生成文本。进行中。ICML (2011年)。

[85] TRUSTWAVE。eHarmony密码转储分析，2012年6月。http://blog.spiderlabs.com/2012/06/eharmony-password-dump-analysis.html。

[86] TRUSTWAVE蜘蛛实验室。蜘蛛实验室/Kore logic-规则。https://github.com/SpiderLabs/KoreLogic-Rules，2012.

[87] TSUKAYAMA，H. Evernote被黑；数百万人必须更改密码。华盛顿邮报，2013年3月4日。https://www.washingtonpost.com/8279306c-84c7-story.html 11e 2-98 a3-B3 db 6b 9 AC 586。

[88] UR，b，KELLEY，P. G .，KOMANDURI，s .，LEE，j .，MAASS，m .，MAZUREK，m .，PASSARO，t .，SHAY，r .，VIDAS，t .，BAUER，l .，CHRISTIN，n .，和CRANOR，L. F .你的密码怎么样？强度计对密码创建的影响。进行中。USENIX安全(2012年)。

[89] UR，b .，SEGRETI，S. M .，BAUER，l .，CHRISTIN，n .，CRANOR，长度f .，KOMANDURI，s .，KURILOVA，d .，MAZUREK，M. L .，MELICHER，w .和SHAY，r.《在密码猜测性建模中测量真实世界的精确度和偏差》。进行中。USENIX安全(2015年)。

[90] 万斯答:如果你的密码是123456，就让它hackme。纽约时报，2010年1月20日。http://www.nytimes.com/2010/01/21/technology/21password.html。

[91] VERAS，r .，COLLINS，c .和THORPE，j.《密码的语义模式及其安全影响》。进行中。NDSS (2014年)。

[92] WEIR，m .、AGGARWAL，s .、COLLINS，m .和STERN，h .通过攻击大量暴露的密码来测试密码创建策略的指标。进行中。综合传播战略(2010年)。

[93] WEIR，m .，AGGARWAL，s .，MEDEIROS，B. D .和GLODEK，B.使用概率上下文无关文法的密码破解。进行中。IEEE Symp。安全与隐私(2009年)。

[94] 现实的密码强度估计。https://blogs.dropbox.com/tech/2012/04/zxcvbn-realistic-password-strength-estimation/，2012.

[95] WHEELER，D. L. zxcvbn:低预算密码强度估计。进行中。USENIX安全(2016年)。

[96] 基于奇异值分解的深度神经网络低足迹说话人自适应和个性化。进行中。ICASSP (2014年)。

[97] 深度神经网络的特征有多可转移？进行中。NIPS (2014年)。

查看出版物统计信息