网络信息安全智能技术与应用的研究进展与趋势

CCF 计算机安全专业委员会

杨　珉[1]　张　磊[1]　荆继武[2]　刘欣然[3]　胡传平[4]

[1]复旦大学，上海

[2]中国科学院大学，北京

[3]中国科学院软件所，北京

[4]公安部第三研究所，北京

摘　要

本文从网络信息安全智能技术与应用的研究进展与趋势这一角度出发，回顾和阐述近几年国际和国内相关的网络安全智能技术的研究工作，详细分析信息安全新技术和新应用的发展，并结合我国实际情况针对网络安全的未来研究提出几点启示和展望，以帮助国内科研工作者迎头赶上国外相关研究。具体来说，本文分别从系统安全漏洞智能挖掘技术、安全防护与补丁检测技术、网络攻击及检测技术、人工智能及其安全等 5 个角度对相关工作进行整理和比较。经过详细对比分析，本文发现，国内研究目前已在部分关键技术领域取得突破，在国际顶级会议上发表了一批高质量论文。但是，与国外研究相比，国内研究起步较晚，其研究成果也一般较为分散，在某一领域内的数个方向上均有少数研究。值得一提的是，国内有关人工智能及其安全的研究虽然数量不多，但是已经基本可以涵盖该领域的关键方向，且已经有部分高质量论文发表。这表明，我国人工智能及其安全已经表现出克服起步晚这一困难的趋势，相信在不久的未来，会涌现更多高质量研究成果。

关键词：网络安全，人工智能，安全研究

Abstract

In this paper, we summarize the recent studies on the intelligent technology used in cyber security, and reveal its trend by thoroughly comparing the research difference between Chinese researchers and foreign researchers. Then, we propose several suggestions to help the Chinese researchers to conduct their future studies. More specifically, we collect and summarize recent research papers from five perspectives, including vulnerability detection in system security, security protection and patch existing detection, cyber-attack and detection, artificial intelligence and its security, etc. After thorough comparison, we find that Chinese researchers have made breakthroughs in some key study areas, and published a number of high-quality papers in top international conferences. However, compared with foreign researches, domestic research started late and widely scattered in a number of study areas. That is, most areas only contain a few high-quality researches and are lack of systematical studies. It is worth mentioning that, although the number of domestic research on artificial intelligence and its security is small, it can basically cover the key research questions in this area, and a series of

high-quality papers have been published. This indicates that Chinese researchers have overcome the difficulty of starting late, and more and more high-quality research papers will be published in the near future.

Keywords: Cyber security, Artificial Intelligence, Security research

1 引言

随着互联网、信息化技术和人工智能的快速发展，人们的日常生活越来越多地与网络空间高度耦合。从衣食住行到金融支付、亲子教育、社会交往，网络空间已成为社会生活的重点支撑。网络空间安全也成为国家战略安全中不可分割的一块重要内容。与此同时，各类安全问题和攻击也层出不穷，手段多样化，技术智能化，对个人隐私和国家安全都造成极大威胁。

近几年，网络信息安全逐渐在全球范围内都成为热度极高的话题。信息泄露事件频发、安全漏洞数量剧增、恶意软件肆虐、网络攻击活动猖獗，网络威胁的数量、影响和恶性程度都与日俱增。例如，在主流移动操作系统上，每年新增的恶意软件数量已经超过 400 多万个，且部分样本开始利用智能技术隐藏其恶意意图。此外，网络安全及其智能技术也逐渐表现出政治化的特点。在新冠病毒疫情期间也成为美国用于攻击我国的借口。因此，亟须对网络信息安全智能科技与应用的发展进行充分研究。

一是攻击手段日益复杂、攻击危害愈加严重。与传统的网络攻击事件相比，现在的网络攻击手段吸收借鉴了其他前沿科技的成熟经验，其工作模式已经从传统黑客的小作坊模式，发展成为利用各种智能科技的有规划有预谋的智能作业模式。在网络链接的任何一个环节，都有可能发起攻击，直接威胁网络用户。例如，最近一起针对区块链应用的 DAO 攻击使受害者受到了约 6000 万美元的损失。

二是用户数据重要性日益提升、规模也迅速变大。随着产业信息化、数字化、网络化进程的加速，互联网与各行业不断融合，致使企业和个人的数据大量被保存在网络空间内。受利益驱动，这些数据也成为各类网络攻击事件的主要目标。此外，随着区块链等一批新型网络技术逐渐被政府和科技部门采用，针对他们的敏感数据的攻击也会快速增加。

三是国内研究起步较晚，成果较为分散。网络安全已成为国内和国际上的行业热点，其本身也与国家利益息息相关，也逐渐成为国际舆论的新武器。但是，国内近几年才开始大幅推进相关科学研究与学科建设，起步相对较晚。因此，亟须对相关科研成果进行整理，以厘清网络信息安全智能技术与应用的研究进展与趋势，以辅助国内研究。

鉴于此，本文从网络信息安全智能技术与应用的研究进展与趋势这一角度出发，回顾和阐述近几年国际和国内相关的网络安全智能技术的研究工作，详细分析信息安全新技术和新应用的发展，并结合我国实际情况针对网络安全的未来研究提出几点启示和展望，以帮助国内科研工作者迎头赶上国外相关研究。具体来说，本文分别从系统安全漏

洞智能挖掘技术、安全防护与补丁检测技术、网络攻击及检测技术、人工智能及其安全等5个角度对相关工作进行整理和比较。经过详细对比分析，本文发现，国内研究目前已在部分关键技术领域取得突破，在国际顶级会议上发表了一批高质量论文，例如对网络安全中网络DNS的研究，对系统安全中移动安全的研究以及对人工智能及其安全的研究等等。但是，与国外研究相比，国内研究起步较晚，其研究成果也很难表现出较高的系统性，而且一般较为分散，在某一领域内的数个方向上均有少数研究。此外，随着今年网络安全研究的火热，越来越多的研究人员开始涉足这个领域，但是在国内，目前还是少数科研人员贡献了大量的高水平论文。为此，为了帮助国内研究者快速追赶甚至超过国外研究，本文归纳总结了几点意见：

首先，应注重网络安全研究的特殊性。网络安全研究作为一个新兴领域，有其本身固有的特点。但是，部分研究者在涉足这个领域时，仍保持着其在原有领域内的研究风格甚至内容，只是将相关研究包装在网络安全这个大话题之下，即换汤不换药。但是，只有充分分析了网络安全本身特殊性的科研工作才会容易让学界认可。例如，作为近几年的后起之秀，移动安全的很多科研工作都聚焦在其独有的生态环境之下。虽然，从方法论上依然采用了传统的程序分析等技术，但是他们指出传统分析技术并不能解决移动安全的特殊问题，例如异步调用等。此外，移动操作系统也在传统Linux内核上叠加了很多其他系统组件，而这些新组件的安全问题也自然成为学界所关心的移动安全中的重要问题。

其次，应注重网络安全研究的系统性。目前，受限于起步较晚，国内研究呈现出较为分散的特点。除了少数科研人员在某些关键技术领域取得一系列突破外，大部分的研究其关联性都比较弱。这也给网络安全领域的学科建设和科研教学提供了相应依据，即对学生和科研人员的培养应注重体系，而不是以多个方向多个领域的入门课程为主。在相关课程和实验的设计上，应注重连贯性，由浅入深，加深学生和科研人员对该领域的认知，而不是浮于表面。在相关研究上，也应当由浅入深，逐步分析问题的根本原因。同样仍以移动安全为例，在深入分析了系统应用层和框架层安全问题后，也可再进一步分析系统内核层安全，也即是反思以Linux内核为主体的系统架构是否能满足当前移动操作系统的安全需求。

最后，应注重网络安全研究的科学性。相对于其他科研领域，网络安全的一大特点就是学界和工业界关联较深，很多网络安全科研问题也即是当下工业界十分关心的工程实践问题。但是，网络安全研究应该跳出具体工程实践问题，充分分析思考其背后的科学原理。例如，漏洞挖掘及其技术是工业界和学术界都非常关心的热点问题，但是相对于对具体漏洞的攻击分析，科学研究应当更加注重漏洞的成因分析、漏洞检测的新技术理念、漏洞危害的生态分析、借助漏洞对软件系统设计的反思等等。

未来几年仍是网络安全及其智能技术的高速发展时期。对此，我们认为，智能技术将是网络安全技术的一个重要突破点。但是，应该注意的是，这里的关键部分依然是安全，即利用智能技术解决安全问题和人工智能系统本身的安全问题两个维度。科研工作应着重分析其问题的安全属性，而不是仅仅将人工智能的相关工作包装在网络安全的外

壳之下。此外，随着互联网技术的进一步发展，用户隐私将会进一步暴露在网络上。用户隐私数据和数字资产的保护也会一直成为一个话题的研究重点。工控系统、智能电网、智能家居等物联网技术也是十分重要的研究方向。但是对普通科研人员来说，其门槛较高，需要相关专业设备。最后，漏洞挖掘相关技术仍旧会是研究热点。例如，近几年非常火热的模糊测试技术（Fuzzing）同时受到学界和工业界的追捧。值得一提的是，符号化执行技术作为漏洞挖掘的关键技术之一，国内鲜有研究，也希望国内能尽快在这一领域取得关键性突破。

本文的后续章节安排如下：第 2 节介绍网络安全及其智能技术的相关国际研究现状，第 3 节介绍国内研究进展，第 4 节对国内外研究进展进行比较，第 5 节对相关发展趋势进行展望，第 6 节对本文进行总结。

2 国际研究现状

本章节从系统安全漏洞智能挖掘技术、安全防护与补丁检测技术、网络攻击及检测技术、人工智能及其安全等 5 个方面对相关国际研究进行整理与分析。

2.1 系统安全漏洞智能挖掘技术

软件系统的安全机制是整个网络空间抵御攻击者的第一道防线，也是安全研究和网络攻击者的首要关注目标。而且近年来随着移动互联网的快速发展，以安卓系统为代表的移动操作系统成为当下系统安全漏洞研究的主要目标。从研究对象的角度，这些研究工作可分为以下几种：

（1）研究移动系统生态中的特有漏洞

这类漏洞产生的原因一般为移动系统的固有设计缺陷，或者是系统开发者在工程实践中犯下了错误。例如，具体来说，Unixdomain[1] 和 ION[2] 研究了安卓网络套接字和底层内存堆接口，并通过查找缺失的权限验证来检测未受保护的公共接口。此外，IntentScope[3] 这篇论文的结果显示，一些安卓组件（例如系统服务）接受来自其他组件（例如应用程序）的组件间访问时，由于一些组件错误配置了它们的意图（Intent）过滤器，导致它们可以被未经授权的应用程序访问。据此，该论文也扩展讨论了安卓框架层的访问控制问题。此外，Zhang 等人[4] 的研究表明，在安卓系统中，当应用软件被卸载后，由于在系统服务中，部分应用软件相关的数据却没有被完全移除，这将导致这些残留数据暴露在隐私泄露的威胁之下。

（2）研究操作系统中的访问控制问题

这类漏洞的基本原理与传统系统安全类似，即系统未对关键敏感行为施加足够的安全检查。但是，在移动系统这个新环境下，由于其开放互联的特性，开发者很难细粒度地对各类涉及用户隐私的敏感行为进行充分保护，导致这类漏洞非常容易出现，而且漏

洞危害巨大。一般情况下，这类工作主要关注安卓系统框架中的访问控制问题和移动应用中的隐私泄露问题。例如，Felt 等人[5]的研究揭示，安卓系统中大量存在的预装应用可能有害于安卓权限模型。具体来说，这些预装应用通常包含大量权限，但同时也可能包含一些漏洞。通过利用这些漏洞，其他低权限应用软件就可以它们为跳板来获取高权限的资源。此外，Kratos[6]比较了不同安卓系统服务间权限是否保持一致。通过在系统服务间找到功能相似的接口，再去比较这些接口间权限检查是否一致，它们发现了很多诸如权限提升和拒绝服务攻击（DOS）相关的漏洞。与之类似，AceDroid[7]利用相同思路比较经过第三方厂商定制化后，相似系统服务接口间是否会保持较为一致的权限检查。

（3）研究二进制安全

具体涉及二进制（Binary）程序分析技术、二进制漏洞挖掘技术以及对二进制安全研究生态的综合分析等。在二进制程序分析方面，优秀的国际研究工作呈现出两大趋势，其一是安全分析技术与深度学习技术的结合，其二是对传统分析技术的再思考。对于第一点，如 Duan 等人[8]在工作 DEEPBINDIFF 中，利用深度学习技术对基本块（Basic Block）进行嵌入（Embedding），并进一步利用在代码相似度检测中。而 Guo 等人[9]在工作 DEEPVSA 中，同样利用深度学习技术优化了传统的 Value-Set Analysis 解决方案。对于第二点，Lu 等人[10]着眼于二进制安全领域中的基础分析技术，即间接调用（Indirect Call）的分析。通过利用程序中的结构体信息，极大增加了分析的准确性。此外，Subarno Banerjee 等人[11]，在其工作 Iodine 中，通过回滚与预测的方案，优化了动态污点分析技术的开销。在二进制漏洞挖掘技术上，除去目前广受关注的模糊测试技术与符号化执行技术，近年来的国际研究工作主要着力于发现各类漏洞的特征，结合传统的程序分析技术进行静态的检测。如 Wang 等人[12]总结出了一类内核中漏洞的特征（对敏感变量缺乏二次检测），并根据该特征进行内核漏洞挖掘。如 Xu 等人[13]通过对内核中 Double-fetch 漏洞进行建模，进而在此基础上提出了新型的漏洞检测方案。同时，国际研究工作也逐渐放眼于整个 Binary 安全研究生态，如数个研究工作以公开 CVE 报告为目标，揭露了漏洞披露过程中的不足之处。具体而言，Mu 等人[14]关注 CVE 报告可复现性的问题，通过分析调研提出了优化 CVE 报告流程与形式的优质建议。

（4）研究漏洞挖掘的关键技术

近几年国际相关研究的一大特点就是针对模糊测试技术（Fuzzing）的关注度非常高。尽管模糊测试技术在工业界取得了巨大的成功，挖掘出了大量的漏洞，但依旧面临着生成的测例质量较差，针对性弱，无法求解较复杂的约束等诸多问题。为了解决这些问题，国际研究者从优化测例生成算法，定向模糊测试，提升求解能力等多方面对已有方案进行改进。Han[15]，You[16]，Wang[17]等人通过对结构体信息和语法信息的分析，生成结构化的高质量输入。Böhme[18]，Chen[19]等人通过调整激励算法，引导模糊测试执行指定的代码片段。Yun[20]，Cho[21]，Chen[22]等人引入了混合执行，符号求解等技术来求解复杂的约束条件；而 Aschermann[23]，Peng[24]等人则基于一些复杂约束的特征，通过对程序转换，动态分析等其他程序分析技术来辅助测例的生成。模糊测试技术除了被用于测

试用户态的程序和软件，还被用于测试其他程序。操作系统内核是世界上最为重要的程序之一，许多工作[25-28]探索了如何用模糊测试挖掘内核中的漏洞。此外，近年来，该技术也被越来越多的应用于测试其他新兴领域的应用。Chen 等人[29]利用模糊测试挖掘 IoT 设备中的内存漏洞；He 等人[30]将该技术用于测试基于区块链技术的智能合约；Gao 等人[31]用模糊测试评估了深度神经网络的安全性；Han[15]等人则关注了 JS 引擎的安全性。最后，有一些研究难点是使用传统程序分析技术很难解决的，为此，研究者引入了机器学习的方法来解决这些难题。He 等人[30]使用机器学习训练高效的输入生成策略；Godefroid[32]等人讨论了如何用机器学习的方法生成符合语法规则的 PDF 测例；She 等人[33]使用神经网络模型来求解模糊测试过程中所遇到的约束。

此外，另外一项发展迅速的漏洞挖掘技术就是符号化执行技术。其通过将程序输入转化为符号化值对程序逻辑进行探索，并收集不同执行路径中对于输入的数据约束，从而生成满足复杂检测条件的输入。符号化执行技术的一个核心挑战在于如何提高其探索效率，避免执行路径过多而导致的“路径爆炸”问题。同时，研究者也利用其生成输入的能力，将其与模糊测试技术结合，提出了新型的混合测试方法。Wong[34]使用自然语言处理技术从应用文档和源代码的注释中提取程序输入的限制信息，并利用这些信息来使得符号化执行专注于程序中对合法输入的处理逻辑的分析，提高了自动化测试样例的生成效率。Pietro[35]提出了 HEX 语言来描述程序执行过程中的堆结构，其系统还能动态监测符号化执行过程中堆数据的变化并对其进行评估，从而减少对于无效状态的分析，提高分析效率。Su[36]首次提出了利用动态符号化执行技术来进行数据流检测的框架，并设计了有导向性的路径搜索策略来快速找出所有数据流可能的覆盖路径。为了提高符号化执行的代码覆盖率，Christakis[37]通过人工和自动化标注代码，将符号化执行流导向未被检验的代码，减少了搜索空间。针对符号化执行的路径爆炸问题，Qiu[38]提出了增量化的符号化执行技术，其对于每一个分析过的函数生成对应的记忆树，并在之后的执行中对已有的函数记忆树进行复用，将对于同一函数的重复分析简化为对于记忆树的遍历，提高了分析效率。Driller[39]是首个将符号化执行技术与模糊测试技术结合提出混合测试技术，并以此进行程序漏洞挖掘的研究。通过二者的结合，使符号化执行技术生成复杂输入的能力和模糊测试技术的程序探索能力互为补充。QSYM[40]更进一步，发现了影响混合测试效率的关键因素，并以此对传统符号化执行技术进行精简。

2.2 安全防护与补丁检测技术

安全漏洞研究的另一个关键问题就是如何对安全漏洞进行修复。作为修补软件漏洞最主要的方式，软件开发商需要及时将补丁或安全防护应用于所有受影响的软件版本。然而，受限于开源软件的开发及使用特点，位于下游的软件开发商很难及时地修补上游软件中出现的相关漏洞。因此，为了避免用户遭受已知漏洞的攻击，检测软件是否及时地应用相关安全防护是一个十分重要的领域内问题，也是近几年国际研究的一个新兴热点。具体来说，主要包含以下三个方面的研究成果。

（1）研究控制流完整性

利用非内存安全语言 C/C ++ 编写的程序通常具有非常高的性能表现，然而相较于现代程序开发语言，他们都缺少相关的安全保证，程序中存在的漏洞会引起十分严重的后果，内存破坏漏洞作为其中最常见的漏洞类型，通常使得攻击者可以控制程序的执行流程。因此，研究人员提出通过控制流完整性保护策略来帮助程序免受控制流劫持攻击。控制流劫持攻击通常会将程序原本的控制流修改到一个非程序预期的地址，因此控制流完整性保护通过限制间接跳转时的目的地址来阻止程序中出现的非预期跳转。受限于程序分析技术很难静态分析出一个间接跳转的准确目的地址，现有控制流完整性保护的实现在间接跳转时通常会允许一个目的地址集合。目的地址集合的大小决定了控制流完整性的保护粒度，因此一些工作尝试通过程序执行过程中通用的语义信息或上下文信息缩减目的地址集合大小。Ren 等人[41]提出利用路径敏感的指向性分析来缩小间接跳转的目的地址集合，提升保护的精确性。Mustakimur[42]等人则提出了一种新的上下文敏感的控制流完整性保护策略。作者将间接跳转发生时函数调用栈作为上下文信息切分跳转地址集合，并根据不同的函数调用序列长度为不同的间接跳转提供不同粒度的保护。虽然这两个工作可以在一定程度上减小程序中所有间接跳转目的地址集合的平均大小，但很难将集合大小的上界有效减少。Mustakimur[43]等人在现有函数调用栈信息的基础上，利用间接跳转地址被赋值时的上下文信息对目的地址集合实现了更均匀的划分，有效减少目的地址集合的上界。除了直接利用上下文信息，Hu 等人[44]则对原程序进行插桩记录执行时的关键上下文信息。通过结合上下文信息，解释执行编译过程中识别的间接跳转相关敏感指令获得间接跳的唯一目的地址，若实际执行时所得地址与该地址不同，则认为程序控制流完整性遭到了破坏。

除了针对普通用户态程序设计的控制流完整性保护，Mario 等人[45]还尝试利用硬件辅助的方式为嵌入式设备中的程序提供控制流完整性保护。从攻击者角度，Andrea 等人[46]对 Windows 系统上的控制流完整性保护技术（CFG）进行了评估。通过分析，作者发现现有技术的设计实现并未与 windows 库有效地结合，从而导致攻击者可以找出特定的代码片段绕过控制流完整性保护。在此基础上，作者在不改变现有防御方案设计的情况下，对现有技术的短板提出了可能的解决方案。

控制流完整性保护技术在实际实现过程中会对原程序进行大量修改，这些修改会对原程序的执行造成潜在的影响。Xu 等人[47]利用不同种类的程序对现有的控制流完整性保护技术实现的适用性进行了分析。研究发现，现有保护技术实现无法适用于包含多线程、及时编译、脱壳、回调等特性的程序，并总结了后续相关技术在由设计到实现时需要注意的问题。

（2）研究内存保护

除了控制流完整性保护，一些研究工作尝试直接为内存及内存分配过程提供安全性保护，从而避免用户态程序和操作系统内核受到针对内存破坏漏洞的攻击。主要涉及以下几个方面：

安全的内存分配器。应用程序层面的内存分配器为程序开发提供了便捷的内存使用

接口，一个安全的内存分配器可以为应用程序的内存安全提供有效的防护。因此，一些研究人员尝试针对不同的漏洞类型提出相应的设计安全的内存分配器[49,51,53,55]。如针对释放后重用攻击，Sam 等人[49]通过隔离释放后的数据来防止之后对其的使用。而 Nathaniel 等人[51]设计实现了一个高效的能力废除系统来阻止释放之后的使用。Liu 等人[55]则通过同步线程定期中和程序中的悬挂指针，以便从根本上防止对释放后利用漏洞的利用，并通过对象源头追踪技术找出程序中存在的释放后利用漏洞。除了针对特定漏洞的防护设计，一些研究人员通过提出新的内存分配机制来提供更全面的保护。Sam 等人[53]增加了分配对象地址的随机化程度，并以极低的开销集成了现有堆分配器中所有层面的安全防护，从而保护程序免受不同类型的堆漏洞影响。

内存隔离技术及内存访问限制。除了在内存分配及销毁时提供安全性保护，内存隔离技术及内存访问限制[48,50,54]作为内存使用时一个重要的技术手段，为程序执行的稳定性及内存数据的安全性提供了有力的保障。Sergej 等人[48]深入分析了 x86 架构中内存隔离的缺陷，并利用虚拟化技术，对客户机提供了可选择的内存保护原语。Hojoon 等人[54]则利用 x86 架构的中间权限层为用户态程序的执行提供了额外的受权限保护的内存区域用来存放程序执行中的敏感数据。国内研究人员 Wang 等人[50]利用监督者模式访问控制技术实现进程内内存隔离，从而保护敏感数据免受内存破坏漏洞的影响。

硬件辅助。除了软件层面的技术，一些工作通过硬件辅助为传统内存保护技术提供了新的思路。例如 Benjamin 等人[56]利用处理器中的微指令实现了对时序攻击的保护及地址随机化等系统安全防御策略。Tommaso 等人[52]对 X86 ISA 进行了扩展，设计实现了一个新的内存访问权限，通过特定的访问指令对具有特定访问权限的内存页进行访问保护。

嵌入式设备保护。随着物联网和智能家居的普及，嵌入式设备程序运行的安全性也逐渐受到更多的关注。一些工作尝试利用嵌入式设备的有限特性为其提供传统主机所能拥有的保护。例如 Donghyun 等人[57]利用 ARM Cortex-M 的特性为嵌入式设备的内存提供了仅可执行权限。而为了在嵌入式设备上保护程序免受控制流劫持攻击，Naif 等人[58]将函数返回地址的存放位置由传统的可写入内存改为可读可执行内存从而实现返回地址的完整性保护。

（3）研究漏洞补丁检测

作为最重要的开源软件代表，Linux 内核被大量下游开发商所使用，定制化的内核运行在了成千上万的设备上。国外的研究者首次针对此类问题展开了相关的研究工作。Feng[284]等人以安全研究人员人工进行补丁存在性分析的思路为参考，通过分析补丁源码所产生的句法差异，生成二进制层面的补丁签名，并通过匹配目标内核与签名来判断补丁存在性。除了开源软件，开源库的使用也存在相应的问题。例如 JAVA 库可以极大地简化软件开发商的开发难度，但由于兼容性等问题，厂商并不会积极地更新软件中所有使用到的开源库，这使得用户会受到旧版本库中相关漏洞的威胁。

相较于 Java 程序和系统内核，厂商在生成用户态二进制程序时有更多的编译选项可供选择，这使得用户态二进制程序的形式更为多样。Qian 等人[285]针对用户态二进制程序中常用的加密库 Openssl 提出了相应的补丁存在性检测方法。通过枚举组合不同的编译

选项、编译器和 Openssl 库版本生成大量参考二进制程序，作者利用程序追踪技术从中提取补丁相关的执行路径特征，并将其训练为补丁存在性检测判别神经网络。

2.3 网络攻击及检测技术

当前网络技术的应用在给人们带来巨大便利的同时，也让人们处在网络安全隐患威胁当中。尤其是随着计算机技术和网络技术应用范围的不断扩大，网络安全方面存在的漏洞越来越多，在这种情况下，如何检测网络的安全性，尽早发现漏洞，规避漏洞成为目前国内外研究和关注的热点。对于网络技术的安全研究，大致可分为两部分：网络结构安全与 Web 安全。

（1）研究网络结构安全

网络结构的安全问题作为网络安全中的传统问题，近些年来也产生了很多研究，大致主要涉及以下几个方面：

DNS（Domain Name System）作为互联网的“中枢神经系统”，是互联网上最为关键的基础设施，如果 DNS 的安全没有得到标准的防护及应急措施，即使网络主机安全防护措施级别再高，攻击者也可以轻而易举的通过攻击 DNS 服务器使网络陷入瘫痪。DNS 系统面临的安全威胁主要包括 DDoS（Distributed Denial of Service）攻击和 DNS 欺骗。目前最广泛使用的针对 DDoS 攻击的防御方法是利用大量的服务器和中间件设备构建高性能的流量清洗中心，但由于硬件设备的成本高，灵活性差，无法快速应对快速发展的攻击手段。随着近些年软件定义网络（SDN）和网络功能虚拟化（NFV）技术的发展，国外一些研究团队提出利用 SDN/NFV 更加灵活和弹性的抵御 DDoS 攻击，Bohatei[59] 利用 NFV 技术基于攻击组成，弹性的调整防御虚拟机的数量，并利用 SDN 将可疑的流量引导向合适的虚拟机。此外，Afek[60] 在 Openflow Switch 上实现了反欺骗技术，对故意消耗服务器资源的欺骗性流量进行过滤。这些方法提高了解决方案的灵活性，但引入了额外开销，导致性能降低。对于 DNS 欺骗的防御方法相对比较丰富，包括提出对 DNS 参数或架构的优化，P2P 的域名交叉引用技术，以及利用密码学方法来提高 DNS 安全性，如 Li[61] 提出利用面向字符的加密算法防御 DNS 缓存中毒攻击。

CDN（Content Delivery Network）通过在网络各处放置节点服务器，在现有的互联网基础之上构建了一层智能虚拟网络，能够解决因分布、带宽和服务器性能带来的访问延迟问题，使内容传输得更快、更稳定。目前大量的网站的使用使得 CDN 已经成为一个至关重要的网络基础设施。为了在这个蓬勃发展的市场竞争，各家 CDN 服务商尽可能地为用户提供灵活的配置选项，以降低部署的工作量。然而，过度的便利往往会引入意想不到的安全缺陷。国外研究团队主要关注 CDN 的应用为现有网络环境引入的安全性威胁。Levy[62] 等人发现 CDN 的使用打破了 Web 安全协议所依赖的数据完整性假设，并提出 Stickler 指导网站与用户如何在网络中进行安全通信。同时，Cangialosi[63] 发现由于 CDN 的使用，负责处理用户敏感数据的网站管理方会将自己的私钥共享给第三方，经过大规模研究调查发现，该现象在当前网络环境中十分普遍。

SDN（Software Defined Network）通过分离网络设备的控制面与数据面，向上将应用及程序接口提供给应用层，从而构建了开放可编程的网络环境，向下将路由策略下发到路由器，实现网络设备集中管理。解决了传统网络缺乏统一管理、可编程可扩展能力不足、灵活性不够高等缺点。然而，SDN 在带来控制集中性和开放性的同时，也引进了新的安全挑战。国外对于 SDN 的安全性研究比较丰富，主要是从 SDN 应用程序的安全缺陷着手，如 Ujcich[64]提出了跨应用中毒攻击，在 SDN 中一个低权限的应用可以通过修改控制器中的共享数据对象欺骗高权限的应用代表其执行高权限操作。也有一些研究者提出了 SDN 的自动化分析框架，如 DELTA[65]能够重现在不同测试环境中发布的 SDN 攻击，并以此去发现新的漏洞。此外，许多研究团队对于 SDN 上恶意主机发起的各种攻击提出了相应的对策，如 Wang[66]提出了 FloodGuard，利用少量的开销，实现对够导致 SDN 网络基础设施过载的攻击的防御。

（2）研究 Web 安全

而对于 Web 应用程序的攻击，我们可以将其漏洞利用类型分为如下几类：任意文件上传攻击、恶意代码注入攻击、CSRF 攻击等（见表 1）。那么针对这些常见的 Web 漏洞，近几年来国际研究学者们也在其相应的安全检测上有了一些成果。本部分将以 Web 应用程序漏洞检测任务为主，对这些方法做简明扼要的介绍和阐述。

表 1

Web 应用程序安全检测技术	精准漏洞检测	文件上传漏洞检测	文献［67］
		CSRF 漏洞检测	文献［68］
		XSS 漏洞检测	文献［69］
	通用漏洞检测	静态分析	文献［70］
		动静结合分析	文献［71］
	新型攻击检测	CPDos 攻击检测	文献［72］
		Web 缓存欺骗检测	文献［73］

针对精准 Web 漏洞中任意文件上传漏洞的检测，Taekjin Lee 等人[67]进行了一次系统的分析，设计了一款名为 FUSE 的工具，可以检测目标网站是否存在任意文件上传的危险。任意文件上传是一种危险的 Web 攻击。这种攻击手段可以将伪造的恶意代码脚本文件上传到目标 Web 服务器，而后利用该文件达到控制服务器的目的。因此对该漏洞的安全检测显得尤为重要。FUSE 在 33 个真实的 Web 应用程序中检测出了 15 个相关危害漏洞，对于任意文件上传漏洞的检测做出了一定的贡献。

而针对精准 Web 漏洞中 CSRF 攻击的检测，Giancarlo Pellegrino 等人[68]设计了名为 Deemon 的工具。CSRF 攻击是一种网络的攻击方式，它在 2007 年曾被列为互联网 20 大安全隐患之一，也被称为“One Click Attack”或者 Session Riding，通常缩写为 CSRF 或者 XSRF。其通过伪装来自受信任用户的请求来利用受信任的网站。由于针对其安全检测工具的稀疏，因此 CSRF 攻击具有较高的危险性。而 Deemon 利用动态建模和较好的建模，通过重放的方式获取用户行为，并分析其一系列的操作是否更改目标服务器的安全

状态，以此来检测网站是否易受 CSRF 攻击。研究人员利用 Deemon 对现有的 10 个主流开源的 Web 应用程序进行检测，发现其存在 29 个安全相关的状态改变请求，有 17 个属于 CSRF 漏洞，因此在 CSRF 攻击的防护上也具有一定贡献。

同样的，对于精准 Web 漏洞中 XSS 攻击的检测，Sebastian Lekies 等人[69]提出了新颖的攻击技术和检测方式。XSS 攻击全称跨站脚本攻击（Cross Site Scripting），恶意攻击者通常会向 Web 页面中插入恶意 JavaScript 代码，当用户浏览该页时，嵌入其中的恶意代码会被执行，从而达到恶意攻击用户的目的。而 Sebastian Lekies 别出心裁地提出可以使用代码重用攻击，利用一些合法代码库中的代码片段进行 JavaScript 的串联攻击。这在对 XSS 攻击检测的相关研究中具有一定的奠基作用。

除了精准漏洞检测外，对于 Web 安全的研究上，还有许多通用 Web 漏洞检测的研究成果存在。我们知道漏洞检测技术一般可分为静态分析、动态分析、动静结合分析三种模式。在静态分析方面，Michael Backes 等人[70]提出了工具 PHPJoern，其通过将 php 源码转换为 AST，并根据 AST 生成 CPG 图，同时借助于 neo4j 对图遍历的简便性，可以快速地检测 Web 应用程序中是否存在漏洞隐患。基于 PHPJoern 较高的分析效率，研究人员在 1854 个应用程序上检测出了大量的 Web 漏洞安全隐患，在 Web 安全的检测上具有一定的奠基作用；在动静结合分析方面，Abeer Alhuzali 等人[71]提出了工具 Navex，其在 PHPJoern 的基础上做了相应的改进，同时结合程序动态运行提取约束，并进行约束求解，在检测漏洞的同时可以生成漏洞利用的攻击脚本，基于动静结合的思想，该工具有效地减少了漏洞的误报，提升了分析效率。

除此之外，对于诸如 Web cache 漏洞的新型攻击也正在兴起，在国际上同样也存在一定的研究成果。Hoai Viet Nguyen 等人[72]提出了 Cache-Poisoned Denial-of-Service 攻击的检测方法，其通过修改 http header 的一些属性，利用 CDN 和服务器之间存在的语义差别进行攻击检测，判断目标网站是否存在利用 Web 缓存投毒进行 DOS 的攻击。而 Seyed Ali Mirheidari 等人[73]也提出了 Web 缓存欺骗的漏洞检测方案。检测思路为通过让受害者访问指定链接，来查看是否可以泄露其隐私信息。

2.4 区块链智能攻防技术

区块链本质上是一种分布式去中心化的数据库技术，其起源于 2008 年中本聪发表的《比特币：一种点对点的电子现金系统》。作为比特币的底层技术，区块链被看作是各种已经相对非常成熟的传统技术的结合性创新产物，如分布式系统，密码学等，同时具备了多种以往同类技术所不具备的安全特性，如去中心化，不可篡改，不可否认，匿名性等等。区块链的产生在最初并没有受到太多关注，但随着人们对其越来越了解，区块链受到的关注也越来越多，且随着以区块链为底层技术的数字货币市值的水涨船高，相应地，对区块链进行的攻击也与日俱增。

区块链隐私与安全一般又可以细分为很多不同的领域。这主要是由于区块链包含的功能组件或者说架构成分非常之多，具体涉及 P2P 分布式网络、数据库、密码学、共识

协议、虚拟机等等，这些方面中任何一部分受到攻击都可能带来巨大破坏或严重损失，因此从攻击的角度来说，这给攻击者带来了较多的攻击面。另一方面，目前国内外区块链隐私与安全相关的学术研究工作也主要集中于这两个方向。图 1 展示了目前国内外在区块链隐私与安全方面主要研究的分布情况。

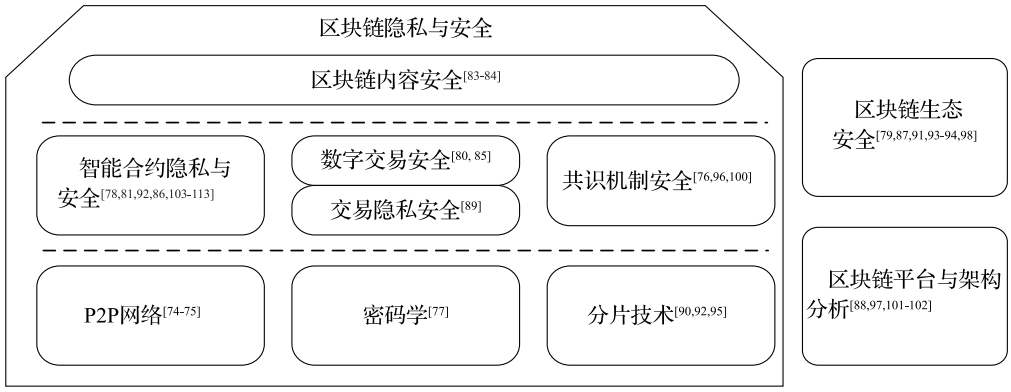

图 1

本节将优先阐述国际上关于区块链隐私与安全的研究，涵盖区块链底层至应用层以及区块链生态等方面。

（1）研究区块链协议与架构

在区块链架构底层方面，公共区块链主要有 P2P 网络、密码学以及分片技术等关键组件。对于区块链底层，一方面，其容易遭受如劫持边界网关协议所导致双花攻击的问题，另一方面也存在系统每秒确认交易数量过低，即吞吐率低下问题。基于此，Apostolaki 等人[74]详细分析总结了针对比特币网络的路由攻击，揭示了网络攻击对比特币带来的严重危害。随后，为了应对此类网络攻击，Marti 等人[75]在第二年相应提出了 SABRE 这一安全且具有良好弹性的中继网络，来对抗针对比特币网络的路由攻击。而在系统性能方面，Kokoris-Kogias 等人[90]研究的则是通过分片技术，来在保证区块链去中心化和安全性的前提下，将区块链的性能提升到 Visa 此类传统交易金融系统处理交易的性能级别。相应的，Luu 等人[92]在结合分片技术提升区块链系统吞吐率的同时，考虑在拜占庭场景下，最多可以容许不超过四分之一算力以上的恶意算力存在时保证区块链安全，使得区块链同时具备了可扩展性与安全性。Zamani 等人[95]则进一步降低了拜占庭场景下对系统安全假设的前提，即可以允许区块链网络内不超过三分之一的节点为拜占庭节点时，通过分片技术实现区块链性能的提升，同时依靠一个可验证安全的重配置机制来确保稳定性，另外此项工作相比于先前的相关工作不需要引入任何可信的预设置步骤，具备更强的实际部署能力。

其次，区块链核心层面上主要包含智能合约、交易以及共识机制这三个方面。其中智能合约方面的研究最多，其主要包含智能合约功能完善与智能合约隐私与安全两个方面。

（2）研究区块链智能合约

智能合约相关的研究主要针对智能合约本身代码或逻辑的缺陷，重点关注的是智能

合约的漏洞研究以及隐私方面的保护。比特币初始设计时只提供了脚本功能，相对于支持图灵完备的智能合约来说，这种脚本功能无法满足将传统垂直行业的应用逻辑进行上链的需求。因此 Bartoletti 等人[78]提出 BitML 这一专门用来编写比特币平台上的智能合约语言，同时提供了相应的编译器能够将对应的智能合约编译成对应比特币网络内的标准交易，以此来实现比特币平台上的智能合约功能。出于同样的目的，Das 等人[82]则研究通过链下可信执行环境来执行合约，并将执行的最终结果写入链上，从而给比特币赋能智能合约功能。

智能合约隐私与安全则主要关注智能合约本身的漏洞以及隐私问题。

智能合约漏洞方面，Grech 等人[103]系统地研究了以太坊中 out-of-gas 的漏洞，解释了 out-of-gas 常见的几种成因，如变长数组的使用等，以及相应的影响。Kalra 等人[104]采用了抽象表示以及符号化模型检查来判断以太坊智能合约中是否存在特定类型的漏洞。Krupp 等人[105]则进一步提出利用智能合约的字节码生成合约的控制流图，并通过符号执行来解决关键路径与状态变化路径的约束，从而达到自动生成漏洞利用的目的。采用类似方法的还有 Luu 等人[106]，但其发现了一些新的以太坊智能合约漏洞，并通过结合针对这些新型的合约漏洞进行监测的逻辑模块来实现识别出智能合约漏洞的目标。同时为了防护先前工作没有办法保护的以太坊上的已部署合约，Rodler 等人[107]提出基于合约动态运行时监控和验证来帮助这些已发布的合约针对性的防御可重入攻击。此外，Torres 等人[108]发现以太坊中存在类似于传统金融行业钓鱼骗局的蜜罐合约，这类合约专门以看似存在漏洞的方式吸引受害者付费调用，从而造成损失，而此项研究则是首次揭示出这一问题。Tsankov 等人[109]研究的是通过对智能合约进行符号化依赖分析，判断合约有无遵从或是违背安全属性来分辨当前合约有无漏洞，这也体现出来智能合约安全方面同运行时验证机制的结合，这也是目前有前景的研究方向之一。Perez 等人[110]系统化的研究了以太坊智能合约的计费机制。他们指出以太坊内一些指令与其计费存在较小的关联性，如 CPU 和内存操作等，导致以太坊智能合约可以在攻击者付出较低成本的情况下被 DoS 攻击。相应的，作者也提出了短期与长期的修复策略，如调整可能被利用的操作码的 gas 收费等。

智能合约隐私问题方面，Cheng 等人[111]提出利用可信执行环境来保障智能合约执行的数据隐私安全，即将合约执行完全置入可信执行环境中，从而确保合约执行的安全以及合约执行中涉及的数据不为外界所知。更具一般性的，Kosba 等人[112]研究的则是通过作者所提出的 Hawk 这一框架以及可信执行环境或是多方安全计算来方便用户快速编写满足隐私性要求的智能合约。

区块链核心层内的交易方面相关研究主要包含基于区块链进行数字交易协议本身的安全以及区块链交易的隐私问题。

前者属于应用层面上结合了区块链进行安全数字支付的协议，Campanelli 等人[80]研究了现有基于零知识的有条件支付内存在的问题，如买家未付款便可知晓商品信息等，相应地作者提出了对应改良的方案，即基于零知识的有条件服务支付协议，用于安全高效的售卖数字服务。但零知识证明会带来诸多问题，例如算法开销过大等问题，

Dziembowski 等人[85]相应设计了基于智能合约而不是零知识证明来进行公平易物的方案。与此类似的方案一般也以通过智能合约实施基于哈希锁定的方式来实现没有第三方存在的情况下卖家买家间的公平交易。

后者本质上关注的是区块链平台内自身交易存在的隐私问题。Kerber 等人[89]研究通过非交互式零知识证明以及作者所提出的密钥保密前向安全加密来保证密钥的隐秘性以及交易接收者的匿名性，同时也确保攻击者不能通过重新发布过去的 PoS 共识协议下的区块来获取任何有用的信息。相应的为了提供交易匿名性，工业界内 Zcash、Monero 等知名的专注于区块链交易隐私保护的区块链平台也相继推出，并且也占据相当大的市场比重。

（3）研究区块链共识机制

此外，作为区块链平台核心的共识机制也是国际上研究的重点。在 PoW 共识协议耗费电力资源、影响系统吞吐量等缺陷日益明显的情况下，关于 PoS 等新型的共识机制从未停止。Badertscher 等人[76]提出了首个在通用可组合安全分析模型下被形式化证明为安全的 PoS 协议。而 Zhang 等人[96]则针对已有的 PoW 共识协议及其各种在原有 Pow 共识协议上进行改良后的变种共识机制进行了分析，而在研究的最后，作者指出原始的 PoW 协议仍然是最好的，已有的诸多 PoW 变种则并没有它们所宣传的那样好。Ekparinya 等人[100]则针对企业界经常采用的 PoA 共识机制进行了研究，作者选定了以太坊 Parity 与 Geth 内对应的 PoA 共识机制，即 Aura 以及 Clique 进行了分析，同时提出了克隆攻击，展示出了在 PoA 共识机制情况下如何通过类似于边界网关协议劫持等方式实现区块链网络的分割，并发动克隆攻击，从而完成双花，最后作者相应给出了解决方案以及推荐的 PoA 共识参数。

同时值得注意的是区块链本身在内容上的安全问题。换言之，区块链内也存在如色情、暴力、恐怖主义等不良信息，如何在不违背区块链数据不可更改的原则下实现对不良信息的去除也是学术界的研究热点之一。Derler 等人[83]通过基于策略访问的变色龙哈希来在不违背区块链规则的情况下修改区块链内违法或不合适的内容。Deuber 等人[84]则提出的是另一种思路，其核心思想是通过投票来决定是否要接受区块链内容修订的提案，如果一个提案得到了多数人的提议，则该提议生效，且该提议涉及的区块链内容会被删除掉。

（4）研究区块链生态问题

最后，除了上述关于区块链内部各组件的安全问题外，区块链的生态安全与区块链架构和平台的分析也是国际上的研究重点。在生态方面，主要是劫持挖矿，即攻击者通过向受害者浏览的网页端嵌入挖矿脚本来非法利用受害者的计算资源进行挖矿并获利。Bijmans 等人[79]大规模研究了网页端劫持挖矿，并揭示了此种攻击的诸多方面特征以及获利情况。Lee 等人[91]则揭示了数字货币在暗网的滥用情况，并指出由于数字货币具有更好的匿名性和难追溯特性，因此在暗网内使用非常广泛。Xu 等人[93]则观测到传统金融市场内倒买倒卖乱象在数字货币交易内的翻版问题，并以案例讲解的方式详细分析了整个流程。Yousaf 等人[94]则进行了对跨多种数字货币平台交易的追踪，并且识别出了多种跨数字货币交易模式，同时也验证了跨多个区块链平台追踪交易的可行性。在区块链

架构和平台分析方面，Kappos 等人[88]研究了以交易隐私著称的 Zcash 区块链平台的匿名性分析，主要是发现大部分交易并没有使用 zcash 的匿名交易特点，同时即使使用了 zcash 的匿名交易，攻击者仍然可以通过可识别的使用模式来减少 zcash 整体的匿名集合问题。Tramèr 等人[97]则主要针对 Monero 和 Zcash 这两种专注隐私保护的区块链平台进行了分析，作者利用如根据零知识证明证据生成的时间，或者能够确定是否某一个 P2P 节点收到了任意一笔交易的支付等信息，对目标区块链平台进行侧信道攻击。这些说明了现有注重隐私保护的区块链平台仍存在着诸多隐私安全问题，因此区块链平台的隐私安全仍然任重而道远。

2.5　人工智能及其安全

近年来，随着深度学习（Deep Learning，DL）的发展，人工智能（AI）技术取得了重大的突破，越来越多的新 AI 技术被提出并被应用在各个领域，例如图像分类，机器翻译，人脸识别等。深度学习有效的一个主要因素就是其复杂的多层非线性结构，帮助其可以有效地学习出数据内在的特征。有一些工作尝试把这样的新 AI 技术应用在安全问题上，例如恶意软件检测、异常行为检测、防御网络攻击等。相关研究发现，这样的新技术在相关安全问题上得到了巨大的效果提升，超过了原经典方法。此外，AI 技术本身的安全和隐私问题也逐步引起社会各界的广泛关注与讨论。在接下来的内容中，本部分将对这两个方面的国际研究分别进行阐述。

（1）研究利用人工智能解决安全问题

表 2 展示了利用人工智能解决安全问题的国际研究中大致涉及的几个关键研究方向：

表　2

AI 模型在安全问题上的应用	恶意软件检测	有监督类方法	文献［126-128］
		无监督类方法	文献［129-130］
		混合类方法	文献［131-132］
	异常行为检测		文献［114，139］
	密码破解		文献［140］
	程序分析		文献［142］
	内存取证		文献［143］
	防御网络攻击		文献［141］
AI 模型应用于安全问题的挑战	可解释性问题		文献［137-138］
	模型老化问题		文献［133-134］
	对抗样本问题		文献［136］

恶意软件检测。恶意软件（Malware）指那些在未明确提示用户或者未经用户许可的情况下在用户终端上安装运行并侵害用户合法权益的软件，例如广告、勒索、间谍软件等[123]。针对恶意软件检测这个问题，经典方法可以划分为静态方法和动态方法，其中

静态方法通常在软件运行之前就可以通过静态分析技术提取出软件的特征，然后通过特征分析或者特征库对比的方式来判断软件是否会存在恶意行为[125]；而动态方法则是收集程序在运行时候的日志等信息，检测到异常行为之后迅速地做拦截[124]。

但是随着恶意软件的不断发展，现有的检测规则越来越难以适应海量的恶意软件家族，针对这个问题，一些工作提出用 AI 技术来应对复杂的检测情况。根据不同的分类角度，我们可以对这些 AI 方法做出不同的分类，例如以特征的形式来看，目前的方法也可以划分为静态方法和动态方法；从使用平台上划分，可以划分为 Windows、Android、PDF 和 Flash 等。本文将从使用方法上来对目前 AI 方法做划分。详细而言，这些方法通常可以划分为有监督和无监督两类：1）有监督类方法需要从每个程序中提取出相应的动态或者静态特征，然后构造一个训练集，其中训练集由若干个（特征，是否为恶意软件）这样的二元组组成。通过不断的学习和调整，AI 模型将学会这些特征与是否为恶意软件之间的联系。在接下来的预测过程中，给定一个新的软件并提取特征之后，就可以交给 AI 模型来预测这个软件是否是恶意软件。2）无监督类方法则不直接用 AI 模型来判别恶意行为，而是利用 AI 技术来对数据或者特征做处理或者简化，然后再基于 AI 模型得到的新特征来做判别。在接下来的内容中，我们将举几个例子来介绍这两类方法。

在有监督类方法中，Nix 等人[126]着眼于安卓恶意软件检测，通过静态分析工具，获取每个 APP 的 System API 调用，然后把这个软件调用过的 API 输入到一个卷积神经网络（Convolutional Neural Network，CNN）中，预测这个软件是否包含恶意行为。作者使用 1000 个左右的 APP 作为训练集，并在 200 个 APP 上做测试，准确率达到了 99.4%。Saxe 等人[128]着眼于 windows 平台上的恶意软件，提把 PE 包的元信息、调用函数和字节码分布等信息输入到一个四层神经网络中，然后预测这个二进制程序是否存在恶意行为。Zhihua Cui 等人[127]提出把二进制程序文件转化为图像，然后交给 CNN 模型训练和预测。在有监督类方法中，比较影响效果的就是输入特征的质量，以及训练集的好坏。

无监督类方法没有训练和预测的过程，需要利用 AI 模型给出的信息来辅助判别。Nur 等人[129]用软件在 Windows 注册表上的信息作为静态特征，然后把所有软件的特征用 K-Means 算法做聚类，根据聚类结果和人工判别，就可以筛选出哪些软件包含恶意行为。Mu Zhang 等人[130]则针对安卓 APP 的 API 调用图，先利用专家知识构建一个数据库，包含了若干个典型的恶意行为调用子图，接下来再利用图算法，计算某个 APP 的调用图与数据库中的恶意行为调用图的相似度，若相似度高则判别为恶意软件。近年来，一些工作提出结合有监督和无监督两类方法来提升判别的准确度，例如 Shuangshuang Xue 等人[131]提出先用无监督方法自编码器（AutoEncoder，AE）把输入特征做降维，再把降维后的特征给有监督模型做训练和预测。Mariconti 等人[132]则提出用主成分分析（Principal components analysis，PCA）对输入特征降维。这类方法可以帮助模型过滤不必要的信息，提升判别准确度。

其他安全问题。除了恶意软件检测，AI 模型也可以应用在其他经典安全任务上。与恶意软件类方法类似，我们可以利用有监督模型直接做判别，也可以用无监督模型对数据做分析和处理。例如在有监督模型中，Min Du 等人[139]提出用循环神经网络

（Recurrent Neural Network，RNN）作为判别器，输入依时间变化的系统日志，经过训练后就可以判别系统中是否存在异常行为。Melicher 等人[140]基于 RNN 实现了一个密码破解器，用以猜出用户的密码。Putchala 等人[141]也基于 RNN，实现物联网中的入侵检测。在无监督模型中，Xiaojun Xu 等人[142]用图嵌入（Graph Embedding）的方法，为每个函数学习出一个实数向量，然后通过向量之间的距离来描述函数之间的相似度。Antonis 等人[143]提出用贝叶斯网络来做内存取证，观察是否有危险行为。

AI 模型应用于安全问题遇到的挑战。虽然 AI 技术相对于经典方法取得了巨大的进步，但是这类方法也带来了新的问题，本文将以恶意软件检测为例，对其中三个问题做描述。第一个就是模型老化（Model Aging）问题，随着时间的推移，安卓恶意软件的发展也越来越快，其躲避检查的能力也会得到提升，这就使得原先的 AI 模型失效。Jordaney 等人[133]发现，一个经过训练的 AI 模型，如果一直不更新模型，在两年的时间后，模型的准确度将从 80% 降到 20%。针对这个问题，Ke Xu 等人[134]提出了一个自我进化算法，在不需要新标签的情况下，可以让模型从新数据中学习出新的规律，防止模型老化。第二个问题就是对抗样本（Adversarial Sample）问题，在其他的 AI 任务中，给定一个训练好的模型，攻击者可以根据模型的特点，在样本上加入轻微的扰动，从而使得判别模型失效[135]，类似地，Grosse 等人[136]发现在恶意软件检测上 AI 模型也存在这样的对抗样本。第三个问题是 AI 模型的可解释性，由于 AI 模型的复杂性，在使用过程中我们通常都把其当成黑箱，但是这样就失去了一定的可靠性，为了更加放心地使用模型，我们需要一个更具有可解释性的方法。针对这个问题，Arp 等人[137]提出了一个方法，可以在检测安卓恶意软件时同时告诉用户为什么把其判别为恶意。

（2）研究人工智能系统本身的安全与隐私

图 2 展示了本部分所总结归纳的该领域的主要研究方向和攻防场景。

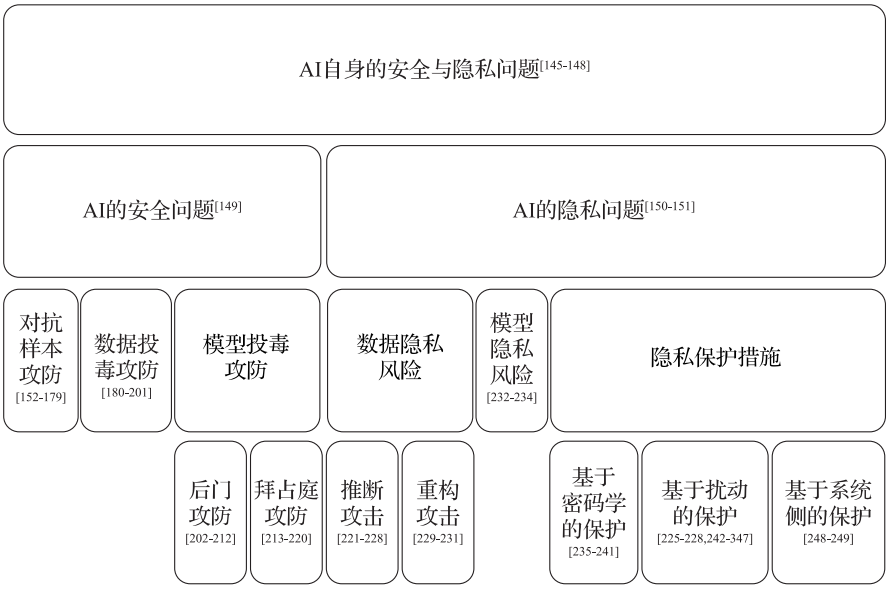

图　2

对抗样本攻防。针对 AI 系统的对抗样本（adversarial example）攻击首先在 2014 年由 Szegedy 等人首先提出，在文献［152］中，Szegedy 等人提出了第一种基于伪牛顿优化技术 L-BFGS 的对抗样本生成算法，并实际展示了如何利用该算法在多种图像识别模型上对正常数据添加人眼无法识别的噪声从而造成模型误判。该工作一经提出便引起了国内外学界和工业界对于对抗样本攻防的持续关注。

攻击侧的研究工作主要集中于两个研究方向：1. 如何提升对抗样本生成算法效率、成功率；2. 研究对抗样本攻击在图像识别之外的其他应用领域上的威胁。为了提升 Szegedy 等人算法的生成效率，Goodfellow 等人随即提出了一种被称作快速梯度符号法（Fast Gradient Sign Method；FGSM）的基于梯度下降的高效对抗样本生成算法[153]。该方法仅利用损失函数对输入数据的梯度就可以找到高效的扰动噪声，后续算法被提出以进一步提升 FGSM 的通用性[175,156]、隐蔽性[155]和准确度[154,157]，例如 Kurakin 等人提出了迭代地运用 FGSM 算法来找到攻击效果更好的扰动[154]并应用于物理世界之中[175]。同时，一些研究者尝试着在其他任务上生成对抗样本，包括语音[158]、文本[161]、视频[162]、移动应用[164]等多种任务。在语音识别的任务上，Carlini 等人证明了音频空间中也存在对抗样本，给定任意自然波形，他们提出了一种方法，可以生成人耳听不见的噪声加在波形上，从而影响语音识别模型的识别结果[158]。

现有的对抗样本防御工作主要可以分为两类：基于信息不透明原则的防御机制和基于对抗训练的防御机制。一方面，基于信息不透明原则的防御机制主要采用隐藏或混淆（obfuscate）模型结构的方法来尝试防御传统的基于模型梯度的对抗样本生成方法[168-170]。例如，Dhillon 通过对神经网络内部的激活函数增加运行时随机性来提升模型鲁棒性[169]。而 Athalye 等人[167]提出一种基于梯度估计的新型对抗样本生成算法一举攻破了几乎目前所有基于隐藏或混淆模型结构的前沿防御方法，给未来对抗样本的防御研究提出了更多开放问题。另一方面，Madry 等人[171]于 2017 年提出了基于对抗训练的防御机制，将深度学习模型的学习目标从学习输入数据与输出标签间对应关系替换成学习输入数据的邻域与输出标签之间的对应，该方法最早源于 Goodfellow 等人 2014 年提出的将生成的对抗样本加入训练集中以增强习得的模型鲁棒性的直观方法[153]。然而在实际应用中，基于对抗训练的防御方法要求对深度学习模型求解双层优化问题（bilevel optimization problem），从而带来了极大的计算开销和理论困难[173]，亟待解决[173-174]。

数据投毒攻防。数据投毒通过篡改训练集中的训练样本，以破坏模型的性能或诱使模型的异常行为。在实际中，由于部分智能系统的准入门槛较低，攻击者因而能够以极低成本伪装成一般用户或经由众包平台假扮数据标注人员向系统提交精心设计的恶意数据来污染 AI 模型的训练数据集。在现有文献资料中，数据投毒攻击比对抗样本的出现时间更早，且而不同于对抗样本主要针对深层神经网络模型，数据投毒攻击对几乎所有的机器学习模型都可能造成不良影响。

2012 年，Biggio 等人率先在包括支持向量机（SVM）和 logistic 回归在内的线性分类器上提出了数据中毒攻击的概念[180]，并讨论了如何通过标签扰动或特征扰动来破坏这类线性模型。随后，国内外研究人员致力于设计和评估各类现有模型和系统上的数据中

毒攻击，包括矩阵分解[181]、回归[194]、图像识别[185]、知识图谱[182]等。同时，他们也开始着眼于对数据篡改方法进行改良，以提升该类数据投毒攻击的隐蔽性。一方面，Koh等人在2017年率先提出了基于统计学的影响函数的新型数据投毒攻击技术，攻击者通过影响函数可以精确地预测当训练集中某些样本的特定扰动对于最终模型的影响，实验表明该方法只需对10个数据点进行投毒以造成模型对于99%以上目标样本的误判[188]；另一方面，除了减少需要进行篡改的数据量，Shafahi等人[183]和Saha等人[186]分别独立提出了无须对数据的标签进行修改的干净标签攻击（clean-label attack），该方法通过对数据集加入在特征空间上与目标误判类接近的其他类样本从而实现对模型的决策边界造成干扰，诱发其异常行为。由于这类样本数据本身和标签保持一致，即使通过人工检测也难以识别，从而该类方法大大增强了数据投毒攻击的隐蔽性。

由于数据污染攻击最早在经典机器学习算法上被提出[180]，因而现有防御工作也主要集中于线性回归[191,192,194]，支持向量机分类[195-196]和非参数估计[197-198]等传统模型上进行。总体来看，数据污染防御的基本思想在于，在训练过程开始前对原始训练集预先进行数据清洗，排除数据与标签不匹配的训练样本，随后让AI模型在清洗后的数据集上进行重训练。常见具体数据清洗方法主要包括：1）设置特定先验规则排除异常数据[201]；2）利用多模型集成学习（ensemble learning）多数投票表决可疑数据[199-200]；3）利用权威的第三方分类模型筛去低置信度数据[197]。然而，这类数据清洗方法对于干净标签攻击是反制能力微乎其微。近年来，少数研究工作也提出通过估计样本影响进行数据清理[189-190]：然而，实验表明在CIFAR-10上用带有随机污染数据训练时，该方法仅能够将被污染的模型准确性提高约2%[189]。

模型投毒攻防。除了数据投毒这类通过污染训练数据造成学习模型的异常行为的攻击之外，近年来一些研究工作指出，潜在攻击者在迁移学习或者分布式学习的情形下能够直接篡改模型的部分模型参数，以造成学习模型在训练过程中无法收敛、在模型上线后整体准确度下降或对特定样本产生误判等恶性后果。在现有相关文献中，后一类对造成对特定样本产生误判的攻击也被称为后门攻击，而前一类造成模型无法收敛或整体准确度下降的攻击也被称作拜占庭攻击。

1）后门攻防相关：后门攻击在迁移学习、模型复用以及分布式学习场景中都有可能发生，主要攻击目标在于通过篡改训练完成或已经上线的学习模型参数，使该模型对一部分攻击者预先选择的特定目标输入进行误分类。在相关文献中，这类特定目标输入通常也被称作“触发器”（Trigger）。除了上述主要攻击目标，后门攻击还要求被篡改的模型在除触发器之外的正常数据上的分类准确度几乎不受影响，从而不易被系统的使用者发现，以提升后门攻击的隐蔽性。根据所采用的不同后门构造技术，现有攻击主要可分为基于数据[202,204,205,212]和基于模型的后门攻击。例如，Gu等人[202]提出在正常数据上增加一些特定的像素模式来形成触发器数据，将其添加到正常训练数据集中，随后在本地利用该数据集训练出后门模型后，替换可能被受害者将复用或者用于迁移学习的模型来源。基于模型的后门攻击思路则在2018年由Liu等人首先提出，主要针对深层神经网络模型。这类攻击主要利用了深层神经网络的过参数化（over-parametrization）性质，通过

人为增加触发器在网络中部分激活神经元的关联权重以增强触发器与目标误判类之间的因果性[203]。

由于基础的后门攻击具有良好的攻击效果以及隐蔽性，因此相应的防御工作面临着极大的挑战。后门攻击的防御策略可以进一步根据其应用的时间点划分为训练中的防御策略与训练后的防御策略。关于训练中的防御策略，现有的研究工作主要致力于利用模型的行为差异区分后门样本与正常样本，进而检测并剔除训练集中的后门样本，并主要通过重训练的方式修复模型[206-209]。例如，Chen 等人[206]利用了神经网络中神经元的激活频度在处理不同类型样本时存在的差异，通过应用聚类方法划分模型训练集，从而实现对训练集中的后门样本的检测与剔除。然而，这一类型的防御策略由于需要通过重训练完成模型修复，会带来极大的时间、空间和计算开销，不适合资源不足的防御者部署。关于训练后的防御策略，现有的研究工作主要致力于发掘后门模型与正常模型内部的特征差异与行为差异，进而判断目标模型是否被注入后门，并主要通过裁剪微调的方式修复模型[202,210,211]。2017 年，Gu 等人[202]发现在神经网络中的后门神经元在处理正常数据与处理后门数据有明显的激活频度特征，并基于此设计了 pruning 算法完成对后门神经元的检测与删除。例如，Liu 等人[210]通过综合 fine-tuning 算法改进 pruning 算法并提出了 fine-pruning 算法，解决了后门攻击者通过绑定后门与正常神经元来迫使防御者平衡模型安全与模型性能的问题。

2）拜占庭攻防相关：在 AI 安全中，拜占庭攻击主要指分布式学习系统中攻击者控制了部分工作节点后，通过对参数服务器发送错误梯度乃至随机噪声以进行攻击，诱使主节点的梯度合并规则异常，导致参数更新决策错误，以扰乱深度学习模型的训练过程。2017 年，Blanchard 等人率先指出，攻击者理论上仅需要控制一个工作节点，便能轻易诱使基于算术平均的经典梯度合并规则输出攻击者想要的任意参数更新方向[217]。后续相关研究工作提出了多种不同的梯度篡改策略，如 Guerraoui 等人于 2018 年提出的单分量攻击[213]以造成诸如模型的学习过程无法收敛、模型的整体表现大幅弱化等恶性后果。随着联邦学习[237]等新型分布式学习范式的出现，近期一些研究者也开始研究同时具有拜占庭攻击与后门攻击特征的针对联邦学习的攻击[221]。

由于近期一系列研究工作指出梯度攻击对于分布式深度学习系统具有重大威胁，相应的防御工作也在学术界上陆续开展。通过分析错误梯度对模型学习过程影响，现有研究工作主要采用多数表决原则以过滤或抵消错误梯度的影响，从而使参数更新方向尽可能接近正确梯度方向[214-217]。例如，经典的 Krum 方法[217]通过计算主节点接收到的每个梯度与其他梯度的相似度以过滤掉一些疑似错误梯度，并证明当恶意节点的比例小于 50% 时，Krum 算法能够保证分布式学习过程收敛。然而 2020 年 Fang 等人通过优化技术构造出了这类基于统计的防御算法的反例从而证实了现有防御算法并非绝对安全[218]。从更为普遍的层面来看，由于基于过滤或均值抵消的防御方法都可以直观理解为梯度间的“多数表决机制”的不同实现，一旦当攻击者控制了多数工作节点之后，这些防御机制大多会失效，甚至反过来会助长错误梯度的攻击效果。针对这些问题，近年 Xie 等人[219]和 Pan 等人[220]分别独立提出利用分布式学习系统的辅助信息来达到在工作节点占

多数的情形下的拜占庭鲁棒性，其中前者利用在训练集上计算的损失函数下降值用于评判各工作节点的可信度。

AI 的隐私问题。隐私问题作为信息安全的一个重要方面，随着 AI 系统的不断发展和落地，同样演化出了一些 AI 系统所特有的隐私风险。作为以机器学习为主要技术的当今 AI 系统，数据和模型也成了隐私侵犯和隐私保护的主要博弈对象，近年来吸引着国内外研究者开始对其进行广泛深入的探究。

1）数据隐私风险：一方面，现实世界中，尤其对于医疗、金融、安防等相关领域，数据集、数据集的全局属性，乃至数据集中是否包含特定数据样本等信息都可能具有高度敏感性。近年来，一些研究者提出了多种不同新型的攻击手段从不同层面揭示了基于机器学习的 AI 系统的各类数据隐私问题。根据攻击的目的性不同，现有工作大致可以分为推断攻击和重构攻击两类。

推断攻击的主要目标是判断模型训练使用的数据集是否满足某种谓词逻辑，通常被认为是一种二分类任务。根据推断目的的不同，现有推断攻击工作又主要分为成员推断攻击[222-225]和属性推断攻击[226-227]。成员推断攻击主要试图揭示某些特定数据样本是否在已知模型的未知训练数据当中，由 Shokri 等人于 2017 年率先提出[223]，根据训练集内样本和外部样本之间在给定模型上的分类置信度之间的差异来推断特定数据是否存在于训练集中。不同于成员推断攻击，属性推断攻击的目标更为粗粒度，攻击者主要希望判断训练集是否具有某些特定的全局属性（例如用于训练人脸识别的数据是否具有模型的），由 Ganju 等人于 2018 年提出并多种深度学习模型上实现了该类攻击[226]。该方法主要通过利用具有或不具有该目标属性的训练集对分类器进行训练，随后训练一个额外的二分类器学习模型参数的差异与训练集是否具有该目标属性之间的关联。

重构攻击的主要目的在于恢复训练集中的具有代表性的部分或者全部数据[229-231]，最早可追溯到 2015 年 Fredrikson 等人[229]首次提出的模型反演攻击（model inversion attack）。在该工作中，他们通过数值优化技术，搜索最大化特定分类结果概率的数据点作为属于该类的“原型图片”，从而实现对学习任务信息的窃取。在 2019 年，Salem 等人[231]和 Zhu 等人[230]在模型反演攻击的基础上进行了更为细粒度的改良，几乎同时提出了数据重构攻击以在当前训练轮次中恢复对应的小批量内的每个数据样本。

2）模型隐私风险：另一方面，在一些工业应用场景中，AI 系统中所部署的模型为相关机构组织的知识产权，具有高度私密性。然而，近年来的研究发现，攻击者仍然能够通过一些侧信道窃取 AI 系统的模型训练后参数[232]、模型结构[233]、超参数[234]等敏感信息。例如，Duddu 等人[233]另辟蹊径，巧妙利用在线深度学习 API 的运算时间，应用强化学习技术准确推断了 API 背后的神经网络的层数、卷积层大小乃至具体的网络结构。

3）隐私保护措施：尽管上述多种针对训练数据的攻击类型具有不同的攻击目标和手段，就防御侧而言，为了增强 AI 系统的隐私性，现有研究的大体思路不尽相同：对攻击者用于窃取隐私的信息源增加隐私保护机制。对于不同的攻击类型，攻击者信源可能为模型输出、模型参数、梯度等多方面。从算法层面而言，现有工作主要通过将这些信源作为实数值向量进行隐私提升，采取的方式大致可分为：基于密码学的防护和基于扰动

的防护。另有部分研究工作从系统层面考虑隐私防护，通过软硬件沙盒等其他安全机制来实现信息源的隐私性。

基于密码学的防护主要集中在联邦学习这一新兴场景中，现有方法主要基于同态加密技术或密钥共享技术实现。密码学中的同态加密技术为分布式学习的安全聚合和安全推断需求提供了自然的支持[237,239,240]，例如近年由微众银行开发的 FATE 框架中即采用了 Paillier 加法同态加密实现[237]。除了同态加密技术。2017 年，Bonawitz 等人提出了一种基于密钥分享的安全多方计算协议以实现在分布式学习过程中的单个计算节点上传的梯度信息对其他计算设备及参数服务器不可见，从而一定程度上缓解了数据重构攻击的风险，并于 2019 年由来自 Google 的同组研究人员在工业级的联邦学习应用中实现[238]。尽管两种基于密码学的技术路线均能够保障梯度聚合的中间结果对于参数服务器不可见，有效规避服务提供者存在泄密的可能，基于同态加密的梯度聚合算法往往具有计算量大的特征，而基于密钥共享的协议则往往存在通讯开销过高的局限性。

基于扰动的防护则通过对待保护信息源增加合理的噪声扰动以降低数据的敏感性。相较于基于密码学的防护，基于扰动的防护通常具有更加高效灵活的优势，但往往无法具有可证明的安全性，且使用者往往需要在隐私性和模型效果上进行人为平衡的局限性。从技术层面来看，这类方法可进一步划分为 Naïve 扰动、差分隐私扰动和基于学习的扰动。Naïve 扰动通常包括例如量化[225]、Dropout[227]等手段。基于差分隐私的扰动则相对 Naïve 扰动具有更为严格的量化隐私保障，这通常体现在差分隐私所增加的噪声规模与数据可区分性之间的关系，例如在最初的成员推断攻击、模型反演攻击、属性推断攻击中均有评估差分隐私技术对于这类攻击的反制效果。除了最初由 Dwork 等人提出的泛用差分隐私定义[246]，事件级差分隐私[245]、局部差分隐私[244]、成员级差分隐私[243]等定义变种也被陆续提出并应用于 AI 隐私的语境中以提供不同粒度的差分隐私保护。除此之外，一些研究者也通过将差分隐私结合到传统的随机梯度下降过程实现了学习过程中的差分隐私保证[243]。基于学习的扰动则主要将隐私保护任务构建成机器学习任务以学习目的性的隐私防护模块。例如，Shokri 等人于 2015 年提出了一种基于适应性部分分量上传的分布式学习协议以减少梯度中可能包含的敏感信息[242]；Salamatian 等人采用人为设定假象攻击者的方式，在对抗训练框架下寻找有效的隐私保护映射[247]。在上述三类方法中，差分隐私通常具有相对最为严格的隐私性保障，以及可以预期的隐私性与模型效果的平衡关系，而同时，也正是由于差分隐私的高度泛用性，在某些特定的任务上，基于学习的扰动可以提供更为确切有效的防护，甚至在一定程度上可以在几乎不损害模型效果的同时，达到更具优势的隐私保护效果[242]。

现有基于系统侧的防护主要针对模型通过一些特殊的软硬件机制实现模型学习及推断过程的隔离性。例如，在 2016 年，Ohrimenko 等人[248]和 Hunt 等人[249]分别独立为基于支持 SGX 特性的处理器的各类机器学习及深度学习模型，如支持向量机、卷积神经网络、序列到序列机器翻译模型等提出了一种基于硬件沙盒的隐私提升方法。这类基于软硬件沙盒的技术尤其对于对抗敏感信源为模型参数的数据隐私攻击以及模型隐私攻击有较强的防御效果，然而由于其对于特殊硬件机制的依赖性，其对于不同平台的兼容性和

可扩展性相较于算法层的防御而言相对薄弱。

3 国内研究进展

本章节主要介绍国内相关研究进展，也从系统安全漏洞智能挖掘技术、安全防护与补丁检测技术、网络攻击及检测技术、区块链智能攻防技术以及人工智能及其安全这5个角度进行整理。

3.1 系统安全漏洞智能挖掘技术

近几年，国内有关系统安全漏洞的智能挖掘技术研究也进步飞速，尤其在移动端操作系统安卓上，先后有数篇文章发表在国际顶会上。例如，ASV[250]在安卓系统服务种的并发控制机制中发现了一个设计缺陷，并证明通过客户端应用程序内的一个代码循环，就可能导致系统服务端受到DOS攻击。而buzzer[251]和Invetter[252]这两篇文章也在安卓系统服务中指出并找到大量的不正确的输入验证，以及由此导致的安全漏洞问题。具体来说，系统服务作为安卓系统提供给软件开发者的接口，负责管理系统资源并检查应用软件是否有权限访问系统资源。但是这种机制忽略了一个关键问题，即应用软件的输入是否可信。具体来说，应用软件在申请权限时并不需要告诉用户该权限会和哪些输入数据一起使用。因此，对于任意系统服务接口，在获取权限后，应用软件都可以传递恶意构造的数据结构。此时，若系统服务并没有对输入数据做足够验证，就会导致安全问题。经过研究发现，该问题不仅在安卓系统服务中真实存在，而且安卓特有的权限模型的也无形中加重了这个问题。具体体现为三个原因。首先，为了减轻应用软件开发者负担，安卓提供了开发者工具包（Android SDK）对权限模型进行包装。但该SDK同时对某些系统接口的参数进行了预设置，例如，将用于身份验证的软件包名这一参数设置为当前应用软件的包名。因为在应用软件实际运行时SDK的代码和开发者代码实际处在同一进程中，开发者其实可以不采用SDK中的默认参数配置，重写并恶意构造对应参数，例如，将软件包名这一参数填写为安卓系统“android”。其次，在系统服务侧，系统接口在做安全检查时严重依赖权限模型，即简洁的通过调用特定系统接口进行权限检查，而非细粒度的考虑当前接口的安全需求。所以，往往缺乏关键的输入验证，甚至直接信任输入的参数。这样恶意软件就可以通过构造输入参数的方式，获取到本不属于它的资源。例如，利用软件包名“android”在后台弹出任意窗口进行钓鱼攻击。再次，华为、小米等第三方厂商对安卓系统定制化时，也只着眼于权限检查，不做甚至删除原有的输入验证，更加加剧了这个问题。

此外，模糊测试技术作为国际上目前非常热门的漏洞挖掘技术，在国内也吸引了大量的关注。为了提升模糊测试生成测例的质量，提高模糊测试的效率，国内研究者也从多方面进行了优化。Chen等人[253-254]采用了一些数学方法来对复杂约束进行求解；

Lv[255]，Yue[256]等人优化了突变和调度算法以更高效的生成测例；Chen[257]等人通过整合多个不同策略的模糊测试器，并根据不同场景选择合适的组合来提升测试的效率。此外，国内研究者同样也将模糊技术迁移到了一些其他有趣的测试场景中。Liu[258]等人将安卓原生系统服务作为模糊测试的对象；Cao[259]等人用基于差分重放的测试技术挖掘 Windows 内核中的未初始化变量漏洞；Chen[260]等人关注于挖掘 JVM 中的漏洞；Jiang[261]等人探索了如何将模糊测试用于测试智能合约中的漏洞。国内研究者在机器学习于模糊测试的结合领域也取得了进展。Zong 等人[262]通过深度学习来过滤生成的低价值的输入，从而提高了模糊测试的效率。

与模糊测试技术的火热相对比，国内关于符号化执行技术和二进制安全的研究则明显少了许多。在近年来为数不多的研究里，FUZE[263]成功地将符号化执行技术用于发现、分析和评估对于漏洞利用有帮助的系统调用，从而为内核 UAF 漏洞的利用提供便利。而针对二进制安全，国内研究工作者同样尝试将深度学习技术用于安全领域的问题分析。如郑炜等人[264]，将深度学习技术应用于安全缺陷报告的文本分析中，该课题也是 Binary 安全研究生态中的关键一环。国内研究工作中，亦有 Binary 分析技术的应用，分析与优化。如傅立国等人[265]，在二进制翻译的场景下构建了新的基于后继关系的形式化模型，为二进制翻译技术的优化研究提供了理论支撑。又如卢帅兵等人[266]，利用了动态二进制翻译和插桩的方式，进行低成本的函数调用跟踪。

3.2 安全防护与补丁检测技术

相较于国外研究，国内针对安全防护与补丁检测技术的研究还停留在初级阶段，主题相对单一且只针对特定漏洞类型。

在控制流完整性保护和内存保护领域，针对传统的缓冲区溢出漏洞，邵思豪等人[267]对现有攻击技术进行了详细的分析，并讨论了缓冲区溢出分析领域未来的研究方向。对于代码复用攻击，张贵民等人[268]提出了一种基于运行时代码随机化的防御方法。通过实时监控攻击者企图获取或利用代码片段的行为，并检测到攻击行为时，对代码进行函数块级的随机化变换，从而阻止代码复用攻击的实现。

在漏洞补丁检测技术方面，Jiang[269]提出了一套在开源软件中检测是否存在对应漏洞补丁的技术框架。具体来说，开源内核经过第三方厂商的修改被应用于大量设备中。然而厂商通常会忽视上游版本中公开的漏洞修补补丁，或者推迟对漏洞的修补。更为严重的是，很多厂商并不会公开相关设备的漏洞修补信息，有些甚至会给出错误信息。然而，设备漏洞修补信息对于诸如政府和企业等极度重视安全的用户是非常重要的。因此一个可靠的第三方定制化内核的补丁存在性检测工具显得尤为重要。目前，最先进的补丁存在性检测工具（fiber）是通过代码签名匹配的方式判断目标内核中是否应用了相关补丁。然而，这种方式很难解决实际应用中所出现的问题。因此，通过对第三方厂商公开的有限的源码及对应的内核进行分析，该研究发现，第三方厂商通常会对源码进行定制化修改，并使用非标准的编译参数生成最终发行的内核。通常这些修改会影响补丁周

围的代码，甚至是补丁代码本身的特征，这极大降低了基于代码特征签名的检测方式在现实场景下的准确性。因此他们提出了 PDiff，一个高效且可靠的针对下游内核进行补丁存在性检测系统。不同于提取代码特征，PDiff 首先对上游应用补丁前、后的内核分别总结出补丁相关的语义信息。通过将目标内核中的语义分别与应用补丁前、后的补丁相关语义进行比较，判断出与目标内核更为相近的版本，并据此给出目标内核的补丁存在性检测结果。与先前工作不同的是，PDiff 同时以补丁前、后两个内核作为参考，并通过计算语义间相似度的方式极大程度上减小了由于定制化和编译带来的影响。

3.3 网络攻击及检测技术

国内研究团队在对 CDN 自身缺陷的研究上比较深入，Chen[270] 发现 CDN 存在受到 DoS 的风险，攻击者可以构建请求转发循环，使得一个网络请求可以被重复甚至无限期的处理，消耗大量资源，进行 DoS 攻击。Guo[271] 对 CDN 的安全性进行研究，发现 CDN 无法对客户提供的域名或 IP 地址进行所有权校验，利用这样的设计缺陷，恶意用户可以滥用 CDN，以服务商意想不到的方式实现对第三方资源的访问。此外，在 CDN 应用所引入的安全威胁方面，Guo[272] 提出由于 CDN 转发机制实现及协议的脆弱性，攻击者可以通过构造合法的请求实现对 CDN 之后的网站进行 Dos 攻击。CDN 的引入目的是提高网络性能同时保护主机免受网络流量攻击，HTTP 范围请求机制设计的初衷是为了尽可能地减少不必要的网络传输，然而 Li 等人[275] 发现，CDN 遇上 HTTP 范围请求，为网络环境引入了新的安全威胁。作者提出了一种新型的 HTTP 放大攻击，RangeAMP 攻击，攻击者不仅可以耗尽 CDN 后源服务器的带宽资源，还可以增加代理节点的带宽资源占用。作者在 13 个流行的 CDN 中测试攻击在实际部署环境中的危害性，发现所有测试的 CDN 都受到 RangeAMP 攻击的威胁。

国内研究团队对于 SDN 的安全性也有一些丰富的研究，对于 SDN 应用程序的漏洞研究，Cao[273] 发现 SDN 中应用程序安装流规则过程中存在校验漏洞—缓冲包劫持，利用该漏洞攻击者能够有效地绕过目前所有的防御系统。此外，Cao[274] 还提出了 CrossPath 攻击，利用控制流量和数据流量中的共享链接扰乱 SDN 的控制通道。在防御侧，虽然 SDN 将控制面与数据面分离，提供了更灵活的网络流量管理方式，但控制面与数据面的通信链路存在受到 DoS 攻击的风险，攻击者通过利用大量的 table-miss 包能够使 SDN 的延迟和丢包率极大增加。针对这种攻击场景，Shang 等人[276] 提出了一种独立于协议的防御框架，FloodDefender，能够有效地消除这类 DoS 攻击，将该框架置于控制器平台与控制器应用程序之间，通过 table-miss 管理，包过滤以及流规则管理，实现对数据面和控制面资源的保护，实验证明 FloodDefender 能够在 SDN 受到攻击时以低于 0.5% 的计算开销将延迟控制在 18ms 内，并保证丢包率不超过 5%。

国内研究学者们同样在 Web 应用程序的安全检测上也做了一些建设。对于精准漏洞检测，李彪[277] 等人提出了一种简易、高效、非破坏性的 SQL 注入漏洞检测方法，并结合多线程技术、广度优先搜索策略，对 SQL 注入漏洞进行检测。而在 XSS 漏洞精准检测

上，潘瑾琨[278]等人针对浏览器扩展的特殊性，引入了 DOM 作为攻击面的漏洞类型，同时针对 Web 应用中广泛存在的正则表达式问题，提出了一种面向正则表达式增强的 XSS 漏洞检测技术，并在浏览器扩展 Greasemonkey 的用户脚本中，成功地检测出了 58 个源于 DOM 的跨站脚本漏洞。而吕成成[279]等人将自适应随机测试方法应用到 XSS 漏洞检测上，提出了 Payload 选择算法 XSSART，提高了对 XSS 漏洞检测的效率。除此之外，在 Web 应用程序安全中，对 WebShell 进行检测也是国内研究学者挖掘的一个热点问题。WebShell 是一种恶意可执行代码脚本，会对网络服务会产生巨大的恶意作用，攻击者可将 WebShell 上传至受害者服务器，达到持久化控制的目的。Tian 等人[280]首次将 CNN 应用到恶意 WebShell 检测上，通过对 HTTP Request 的研究和 word2vec 的创新，他们提出了 CNN-WebShell 模型，并在经典分类器比较中拥有较好的表现。Wang，Jiabao[281]等人在此基础上引入了 LSTM 算法进行优化，并解决在 HTTP Request 流量分割时忽略了单词间关系的问题，有效地提升了检测准确率。而 Fang，Yong[282]等人更换了对 WebShell 的检测的切入点，其将检测重点放在了文件检测上。在原有的静态特征基础上，根据 Opcode 的特点进行了特征提取，使用 fastText 和随机森林算法相结合的 FRF-WD 模型，其准确率相较于不考虑 Opcode 特征与 fastText 提高了 8.95%。Cui 等人[283]同时也对随机森林算法进行了优化，将其与梯度提升迭代决策树算法相结合，相较于主流 WebShell 查杀工具 D Shield、WebShellKiller 等都拥有更高的准确率。

3.4 区块链智能攻防技术

相比与国际上关于区块链隐私与安全的研究，国内研究整体处于较为落后的态势，但已有起步态势。

在区块链底层方面，国内相关研究较少，Bai 等人[77]提出了一种分析以及检查 PoW 共识协议的方法，同时针对以太坊中被认为是可以抵抗专用集成电路挖矿的 Equihash 算法，设计了相应的对抗求解器，最后也给出了对专用集成电路友好以及对抗所涉及的核心因素。

在区块链核心层内的智能合约方面，Chen 等人[81]细致地从以太坊中符合 ERC20 标准的代币合约的三个方面，即标准接口、标准事件日志以及实际合约状态修改，来检测以太坊中合约实现与标准不一致带来的问题。Fu 等人[86]则专注于以太坊版本实现上潜在存在的漏洞，并且创新性的利用以太坊虚拟机不同编程语言实现版本间行为上不一致的检测，来自动发现这些不同实现版本潜在的漏洞问题。同时，Jiang 等人[113]研究的则是利用模糊测试的思想来发掘智能合约漏洞，其核心思想是通过智能合约的 abi 来生成测试用例，然后通过定义测试预言机来检测漏洞，并且作者通过插装 EVM 来记录合约执行日志，同时通过分析这些日志来挖掘漏洞。

在区块链架构和平台分析方面，国内的徐蜜雪等人[101]提出借鉴拟态的动态异构冗余架构和密码抽签的思想，结合安全性定义和参数选择规则来实现拟态区块链这一安全区块链模型方案，十分的具有创新性。此外，国内叶聪聪等人[102]提出了一种根据区块

链结构来评估和检测安全性的方法，其核心是依据每个结构到达稳定状态的概率来评估系统的安全性，这种检测安全思路也是非常独到的。

在区块链生态方面，国内的 Hong 等人[87]设计了 CMTracker 来从哈希计算以及 Javascript 调用栈两个角度对网页端劫持挖矿进行了大规模的精确检测、评估与分析研究。Zhou 等人[98]则大规模地对目前以太坊内部署的智能合约内存在攻防情况进行了全面细致的评估与分析，揭示了以太坊内智能合约攻防不断演进的攻防现状。

3.5 人工智能及其安全

在人工智能及其安全领域，相比于国际研究，虽然国内研究成果数量相对较少，也已能够覆盖该领域的各个方面。

在利用人工智能解决安全问题方面，国内研究人员在成共将人工智能运用到安全场景上的同时，也开始考虑模型优化和模型解释性的问题。例如，国内研究人员郭文博等人[138]提出了一个模型，可以对任意一个 AI 判别模型的判别结果做解释，输出这个模型在判别时主要着眼于哪部分数据，例如在恶意 PDF 检测中，可以输出因为哪一个特征而判别一个 PDF 是恶意。丁岱宗等人[144]着眼于时间序列预测中的极值点，建立了理论框架，研究为什么极值点会对深度学习模型有较大的影响，然后基于极值理论提出了一个更具有鲁棒性的模型。

在对抗样本攻防方向中也取得了不少成果。图像数据方面，Ma 等人对基于卷积神经网络的活体检测系统实现了 DeepFool 的改良算法用于提升对抗扰动的隐蔽性[166]；文本数据方面，Ren 等人提出用语义网找到同义词来替换原单词，从而生成对抗样本，干扰文本分类的结果[159]。类似地，Jin 等人也通过替换词的方法生成对抗样本，影响文本蕴含任务结果[160]，后续 Li 等人在多种实际应用场景中进行了相关攻击方法的系统化评估[176]；Wang 等人则通过估计词句对于模型的影响对中文情感分类器设计了对抗样本生成算法[165]。Zhang 等人[179]则验证了神经网络的可解释性方法对于对抗样本的脆弱性。在防御侧，Li 等人[178]对文本分类模型提出了一种基于多模态特征的对抗样本防御算法，Ling 等人[177]实现了一种用于评估商用 AI 系统对抗样本鲁棒性的评测系统。

在数据投毒攻防方向，Ji 等人[187]对干净标签攻击方法进行了改良，从而进一步提升了其攻击效果，并评估了其对于多种应用场景的实际威胁；Pang 等人[192]对遭到数据投毒后的各类真实系统进行了对抗样本脆弱性分析。

在模型投毒攻防方向，Pan 等人[220]提出利用分布式学习系统的辅助信息来达到在工作节点占多数的情形下的拜占庭鲁棒性，通过强化学习技术，设计基于神经网络结构的适应性梯度聚合模块，通过分析并挖掘节点提交梯度的历史记录，并基于相应的学习过程中的额外辅助信息，以动态评估各参与学习过程的计算节点的可信度，主动学习拜占庭节点的恶意行为模式，从而同步调整梯度聚合策略，以保持 AI 分布式系统在高恶意节点比例情况下的训练过程安全，并基于统计学习理论给出了该方法拜占庭鲁棒性的形式化证明。

在 AI 隐私方向，Pan 等人[228]通过构建量化评估算法率先对 8 种包括谷歌、脸书、百度等大型 IT 公司开发的商用通用语言模型的隐私漏洞进行了系统化评估，并在国际上首次证实了上述这些通用语言模型在基于云服务的文本智能系统应用中均可能通过未加密的文本高维特征暴露用户隐私，易于遭受攻击者通过逆向工程推测与用户身份、地理、生物、医疗密切相关的敏感信息，并提出多种新型防护措施用以提升上述各类商用模型的隐私性。Xu 等人提出了一种基于同态加密技术的对于 K 近邻算法的隐私保护机制[241]。

此外，为了向国内学界普及 AI 系统安全与隐私地相关研究进展，近年来国内多个研究课题组也陆续在核心期刊上发表了多篇相关的领域综述文献[145-147,150-151]。

4 国内外研究进展比较

从前文介绍中不难发现，在网络安全领域，整体上来说，相对于国外研究，国内研究仍处于起步阶段，其研究成果较少，涉及领域较为分散。尤其在传统的漏洞挖掘相关技术领域，例如模糊测试和符号化执行。但是，在这之中我们也发现，国内研究在某些关键技术领域已经取得突破，其成绩与水平已经不亚于相关国际研究。例如，在移动安全和网络结构安全上的研究。此外，令人幸喜的是，在人工智能及其安全这一领域上，国内研究已经可以做到覆盖主流方向，并在关键问题上取得突破。人工智能是网络安全领域的一个关键技术突破点，也是一个新兴热点。在这一网络安全子领域上，国内研究的发展速度很快。在可预见的未来，也将会涌现出更多更有价值的研究工作。下面我们针对系统安全漏洞智能挖掘技术、安全防护与补丁检测技术、网络攻击及检测技术、区块链智能攻防技术以及人工智能及其安全这 5 个方面分别比较国内外的研究进展。

在系统安全漏洞智能挖掘技术方面，国内研究目前主要涉及了移动操作系统中的应用层和系统框架层的漏洞研究，但是相对于国际研究，对于底层的 Linux 内核以及系统框架中的 Native 部分研究较少。在漏洞挖掘关键技术上，国内近年的研究也很少涉及符号化执行等。这反映出，国内研究其实在传统的系统安全领域积累薄弱，在涉及二进制与 Native 等方面的研究不够深入。只关心了在操作系统中与用户交互的部分，而忽略了系统内部核心的运行机制。但是，不可否认的是，国内在移动安全领域的研究水平已经很高，在系统应用层和框架层的研究成果很丰富。后续应当注意往系统底层拓展，早日实现移动操作系统全软件栈的系统性研究。

在安全防护与补丁检测技术方面，从前文可以看出，国外对控制流完整性和内存保护的相关研究已经较为成熟，但不幸的是，国内对控制流完整性保护的研究还未完全起步。而且，国外研究除了改进技术的固有问题还尝试找出更多样的应用场景，并在一定阶段后对现有技术进行了对比分析。在这一领域，国内只在某些关键技术例如补丁检测上取得了突破，目前亟须在更多问题上产生高质量研究成果。

在网络攻击及检测技术方面，相对于国际研究，国内研究在数量和质量上都取得了

不错的成绩。例如，与网络结构安全相关的研究在近几年中发表了多篇高水平论文，甚至影响到了相关网络标准的修订。但是，国内研究也存在一些瑕疵。例如，国内对于Web 应用程序检测的研究手段目前还处于一个较为初级的阶段。较多工作依旧停留在对攻击测试用例的组合测试，算法优化，对于新型攻击技术的提出和检测还较为空缺。同时在 Web 应用程序通用漏洞检测方面内容匮乏，不仅支持的漏洞类型偏少，同时大多无法进行 exp 的自动生成，需要人为辅助对漏洞点进行校验，相较于国际研究中 Navex 等工具的表现有着较大的上升空间。

在区块链智能攻防技术方面，国内外研究方向对比差异是比较明显的，如表 3 所示。

表　3

国际研究方向	国内研究方向
区块链网络	哈希算法
分片技术	智能合约
智能合约	区块链架构和平台
数字交易	区块链生态
区块链交易隐私	
共识机制	
区块链内容	
区块链生态	
区块链平台与架构	

由上可见，国内外研究方向侧重点略有不同，国际上主流的研究方向基本覆盖区块链及协议安全的诸多研究面。相比之下，国内则更侧重于应用层面的安全问题，对于区块链及协议安全的其他底层研究层面，如密码学相关协议（零知识证明等）、网络、共识协议、分片技术等方面的研究则相对较少或缺失，这也代表着国内未来区块链技术安全领域发展的新需求。区块链的应用前景是广阔的，这更要求我们能够掌握区块链的核心技术，这其中自然也包括区块链各方各面的安全技术，因此国内相对国际落后的研究状况亟须改变。近年来国内高水平区块链及协议安全方面的论文发表呈现增多趋势，这显示出国内区块链及协议安全方面的研究已经有迎头赶上的态势，但预计在未来一段时间内，国内外研究进展仍然会保持国内追赶国外的这一态势，相应的，那些现有国内研究不足或缺失的领域则是作为相关科研人员的我们需要重点攻克的。

在人工智能及其安全领域，与国外研究相比，国内研究取得了相对不错的成绩。作为一个新兴研究领域，就起步时间而言，国内的相关研究从 2018 年开始产生萌芽，于近两年逐渐在国内外会议期刊上出现，而国外学界于 2014 年前后便已出现对 AI 系统的数据投毒、对抗样本和隐私问题的相关文献发表；就成果数量而言，国内研究仍相对滞后，在国际顶级会议刊物上发表的国内机构作为主要作者的相关工作不足 10%；就研究方向而言，国内现有相关研究成果仍主要集中于对抗样本攻防问题，近一两年逐渐开始出现部分探索其他重要方向的研究成果，尤其是模型投毒攻防和 AI 隐私问题，仍未引起国内学界的重视，相比之下，国外在该方向上则具有较为深厚的技术积累。尽管如此，随着 AI 系统与社会生活的结合愈发紧密，国内学界已逐渐开始认识到 AI 系统的安全和隐私问题研究的重要性和必要性，近两年在国际上也出现越来越多国内学者做出的具有影响力的研究工作，而且也基本涵盖了该领域的关键方向。在可预见的未来，该领域会涌现出更多高质量的国内研究。

5 发展趋势与展望

网络安全发展的一个关键方向就是人工智能技术，主要涉及利用人工智能解决安全问题与人工智能系统本身的安全与隐私问题两个方向。同时，通过对近年网络安全相关研究的整理也发现，在其他领域也热衷于通过引入人工智能来解决某些关键问题。如在漏洞挖掘中，将代码特征识别与机器学习结合，以实现漏洞代码的自动分类等。但是，传统网络安全问题依然不能忽视。

在系统安全漏洞智能挖掘技术上，一方面要重视智能化与自动化，充分发挥人工智能的优势，以节省人力，将漏洞挖掘从依靠人力的低效模式，向借助人工智能的自动模式靠近。另一方面，也不应忽视漏洞挖掘关键技术的突破。例如，通过优化模糊测试和符号化执行技术，使这些传统技术能更快速高效的定位漏洞代码，再尝试引入人工智能，进一步优化减少其中的人工干预，才能最终使得漏洞挖掘能同时兼顾其效益与速度。

在安全防护与补丁检测技术上，作为一个重要的保护策略，控制流完整性保护还并未得到非常广泛的部署和使用，其中一个最重要的原因是保护所需的开销十分巨大，因此如何减少保护所用开销是该防御技术的一个重要发展方向。现有丰富的硬件辅助技术是一个可能的选择，例如已经有工作[43]尝试使用 Intel MPX 和 TSX 技术减少间接跳转时的监控和分析开销。但诸如 Intel PT 技术，虽然可以允许操作系统内核高效记录上下文信息，如何将其使用在用户态程序的保护实现上还有待研究。另外，提高保护技术的适用性、选取更精细的上下文信息进一步提高目的地址集合切分的精确性仍是控制流完整性保护技术接下来发展的重点方向。此外，国内外对于补丁存在性检测的研究还处于初步阶段。除了进一步提出针对不同软件类型的补丁存在性检测工具外，还需要思考通用的检测思路，例如从不同类型的软件中提取语言无关的特征与补丁源码进行匹配。另外，与深度学习技术的结合也可以作为进一步提高检测效率和准确率的方式。虽然国内已有研究人员将机器学习与补丁检测相结合[284]，但仍受限于使用场景。作为更深入的探索方向，研究人员需要进一步探索如何在训练集有限的情况下，从补丁程序中总结出具有代表性的语义信息，并用此训练出高精度的检测模型用于检测。

在网络攻击及检测技术上，由于开发者代码风格和网络协议的多样性，服务器后端语言生态的复杂性，该方向研究中需要打开视野，不局限于当今热门的各类网络技术和服务器后端语言，灵活的适应技术的发展，将传统的漏洞检测技术创新修正，并将其发展为适合当下环境的新型技术。网络检测技术应由多种方案相辅相成，由内而外地解决问题，在系统评估的基础上对薄弱环节进行安全加固，保护关键节点，建立起全球范围的合作协调机制，从而保证互联网安全协调发展。

在区块链智能攻防技术上，在对已有国内外研究分析的基础上，区块链及协议安全方向在未来会更多地关注共识协议的安全、分片技术的安全、链上链下以及跨链时所涉及的交易数据隐私与安全等方向。同时随着区块链内各模块技术的不断发展和演进，区

块链架构或者协议也可能出现与其设计初衷不再能很好融合甚至是互斥的情况，例如随着分片技术的研究与部署，跨分片的交易可能会给原有区块链模型带来更多安全性上的挑战。在未来的发展趋势上，区块链本身会呈现内部组件不断演化的趋势，如采用能抗量子攻击的格密码技术以及更高效安全的共识机制等，同时不同区块链平台会通过侧链或者多链等方式，朝着互融互通打破数据孤岛的方向前进，这些都能够从现有的区块链发展中总结出来，因此这些方向也会是未来区块链及协议安全研究所应关注的重点。目前，我国已经确定要大力推广区块链在各行各业的应用。2019 年 10 月 24 日，习近平总书记在中央政治局第十八次集体学习时强调把区块链作为核心技术自主创新的重要突破口，加快推动区块链技术和产业创新发展。结合目前国内相关的研究发展态势，可以预见我国在区块链及协议安全领域的发展未来是光明的。

在人工智能及其安全上，从整体发展程度来看，国内外现有 AI 系统安全与隐私研究均仍处于早期阶段，研究内容仍主要集中于发现 AI 系统的各类安全隐私风险，相应防御侧研究仍相对滞后，未来针对各类 AI 安全和隐私问题的防御侧工作将成为一个主要研究热点。例如，在对抗样本方面，在 Athalye 等人[167]一举攻破了几乎目前所有基于隐藏或混淆模型结构的前沿防御方法之后，如何有效地防御对抗样本、如何形式化验证 AI 系统的对抗样本鲁棒性、如何平衡对抗样本鲁棒性和系统性能等仍然是该领域的重要开放问题；在投毒攻击方面，随着干净标签攻击等极具隐蔽性的攻击算法的提出，现有的防御方法再次遭到了冲击，国内外学术界仍未找到有效的反制手段；在隐私问题方面，随着 AI 系统的各层面的隐私问题不断被揭示，未来研究一方面应进一步扩大对于 AI 系统的隐私泄露评估，逐步形成系统化的隐私评估框架，另一方面也应不断完善各类隐私提升机制，寻求在用户隐私和 AI 系统服务质量的平衡。最后，随着联邦学习等新型分布式 AI 系统逐渐应用于医疗金融等实际场景，未来学界也应加大对于分布式 AI 系统所面对的各类可能攻击的研究力度，完善相关的防御手段以保障分布式 AI 系统在实际应用中的安全性和隐私性。

6 结束语

本文从网络信息安全智能技术与应用的研究进展与趋势这一角度出发，通过归纳分析近年来在网络安全领域的两百多篇研究工作，回顾和阐述近几年国际和国内相关的网络安全智能技术研究，详细分析信息安全新技术和新应用的发展，并结合我国实际情况针对网络安全的未来研究提出几点启示和展望，以帮助国内科研工作者迎头赶上国外相关研究。具体来说，本文分别从系统安全漏洞智能挖掘技术、安全防护与补丁检测技术、网络攻击及检测技术、人工智能及其安全等 5 个角度对相关工作进行整理和比较。经过详细对比分析，本文发现，国内研究目前已在部分关键技术领域取得突破，在国际顶级会议上发表了一批高质量论文，例如对网络安全中网络 DNS 的研究，对系统安全中移动安全的研究以及对人工智能及其安全的研究等等。但是，与国外研究相比，国内研究起

步较晚，而且成果一般较为分散，在某一领域内的数个方向上均有少数研究。此外，随着近年人工智能研究的火热，越来越多的网络安全研究人员开始涉足这个领域，使得近年来网络信息安全智能技术得到快速发展，虽然其研究成果与国外相比仍然很少，但是这一领域内的研究已经能覆盖很多关键性方向，且已经有少数高质量论文发表。

未来几年仍是网络安全及其智能技术的高速发展时期。对此，我们认为，智能技术将是网络安全技术的一个重要突破点。但是，应该注意的是，这里的关键部分依然是安全，即利用智能技术解决安全问题和人工智能系统本身的安全问题两个维度。科研工作应着重分析其问题的安全属性，而不是仅仅将人工智能的相关工作包装在网络安全的外壳之下。此外，随着互联网技术的进一步发展，用户隐私将会进一步暴露在网络上。用户隐私数据和数字资产的保护也会一直成为一个话题的研究重点。工控系统、智能电网、智能家居等物联网技术也是十分重要的研究方向。但是对普通科研人员来说，其门槛较高，需要相关专业设备。最后，漏洞挖掘相关技术仍旧会是研究热点。例如，近几年非常火热的模糊测试技术（Fuzzing）同时受到学界和工业界的追捧。值得一提的是，符号化执行技术作为漏洞挖掘的关键技术之一，国内鲜有研究，也希望国内能尽快在这一领域取得关键性突破。

参考文献

[1] Shao Y, Ott J, Jia Y J, et al. The misuse of android unix domain sockets and security implications[C]. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016: 80-91.

[2] Zhang H, She D, Qian Z. Android ion hazard: The curse of customizable memory management system [C]. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016: 1663-1674.

[3] Jing Y, Ahn G J, Doupé A, et al. Checking intent-based communication in android with intent space analysis[C]. Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. 2016: 735-746.

[4] Zhang X, Ying K, Aafer Y, et al. Life after App Uninstallation: Are the Data Still Alive? Data Residue Attacks on Android[C]. NDSS. 2016.

[5] Felt A P, Wang H J, Moshchuk A, et al. Permission Re-Delegation: Attacks and Defenses[C]. USENIX security symposium. 2011, 30: 88.

[6] Shao Y, Chen Q A, Mao Z M, et al. Kratos: Discovering Inconsistent Security Policy Enforcement in the Android Framework[C]. NDSS. 2016.

[7] Aafer Y, Huang J, Sun Y, et al. AceDroid: Normalizing Diverse Android Access Control Checks for Inconsistency Detection[C]. NDSS. 2018.

[8] Duan Y, Li X, Wang J, et al. DEEPBINDIFF: Learning Program-Wide Code Representations for Binary Diffing[C]. In NDSS 2020.

[9] Guo W, Mu D, Xing X, et al. {DEEPVSA}: Facilitating Value-set Analysis with Deep Learning for

Postmortem Program Analysis[C]. 28th {USENIX} Security Symposium ({USENIX} Security 19). 2019: 1787-1804.

[10] Lu K, Hu H. Where does it go? refining indirect-call targets with multi-layer type analysis[C]. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019: 1867-1881.

[11] Banerjee S, Devecsery D, Chen P M, et al. Iodine: fast dynamic taint tracking using rollback-free optimistic hybrid analysis[C]. 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019: 490-504.

[12] Wang W, Lu K, Yew P C. Check it again: Detecting lacking-recheck bugs in os kernels[C]. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018: 1899-1913.

[13] Xu M, Qian C, Lu K, et al. Precise and scalable detection of double-fetch bugs in OS kernels[C]. 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018: 661-678.

[14] Mu D, Cuevas A, Yang L, et al. Understanding the reproducibility of crowd-reported security vulnerabilities[C]. 27th {USENIX} Security Symposium ({USENIX} Security 18). 2018: 919-936.

[15] Han H S, Oh D H, Cha S K. CodeAlchemist: Semantics-Aware Code Generation to Find Vulnerabilities in JavaScript Engines[C]. NDSS. 2019.

[16] You W, Wang X, Ma S, et al. Profuzzer: On-the-fly input type probing for better zero-day vulnerability discovery[C]. 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019: 769-786.

[17] Wang J, Chen B, Wei L, et al. Superion: Grammar-aware greybox fuzzing[C]. 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE). IEEE, 2019: 724-735.

[18] Böhme M, Pham V T, Nguyen M D, et al. Directed greybox fuzzing[C]. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017: 2329-2344.

[19] Chen H, Xue Y, Li Y, et al. Hawkeye: Towards a desired directed grey-box fuzzer[C]. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018: 2095-2108.

[20] Yun I, Lee S, Xu M, et al. {QSYM}: A practical concolic execution engine tailored for hybrid fuzzing[C]. 27th {USENIX} Security Symposium ({USENIX} Security 18). 2018: 745-761.

[21] Cho M, Kim S, Kwon T. Intriguer: Field-level constraint solving for hybrid fuzzing[C]. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019: 515-530.

[22] Chen Y, Li P, Xu J, et al. SAVIOR: Towards Bug-Driven Hybrid Testing[C]. 2020 IEEE Symposium on Security and Privacy (SP). 15-31.

[23] Aschermann C, Schumilo S, Blazytko T, et al. REDQUEEN: Fuzzing with Input-to-State Correspondence[C]. NDSS. 2019, 19: 1-15.

[24] Peng H, Shoshitaishvili Y, Payer M. T-Fuzz: fuzzing by program transformation[C]. 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018: 697-710.

[25] Schumilo S, Aschermann C, Gawlik R, et al. kafl: Hardware-assisted feedback fuzzing for {OS} kernels[C]. 26th {USENIX} Security Symposium ({USENIX} Security 17). 2017: 167-182.

[26] Pailoor S, Aday A, Jana S. Moonshine: Optimizing {OS} fuzzer seed selection with trace distillation[C]. 27th {USENIX} Security Symposium ({USENIX} Security 18). 2018: 729-743.

[27] Xu W, Moon H, Kashyap S, et al. Fuzzing file systems via two-dimensional input space exploration[C]. 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019: 818-834.

[28] Kim K, Jeong D R, Kim C H, et al. HFL: Hybrid Fuzzing on the Linux Kernel[J].

[29] Chen J, Diao W, Zhao Q, et al. IoTFuzzer: Discovering Memory Corruptions in IoT Through App-based Fuzzing[C]. NDSS. 2018.

[30] He J, Balunović M, Ambroladze N, et al. Learning to fuzz from symbolic execution with application to smart contracts[C]. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019: 531-548.

[31] Gao X, Saha R K, Prasad M R, et al. Fuzz Testing based Data Augmentation to Improve Robustness of Deep Neural Networks[J].

[32] Godefroid P, Peleg H, Singh R. Learn&fuzz: Machine learning for input fuzzing[C]. 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2017: 50-59.

[33] She D, Pei K, Epstein D, et al. NEUZZ: Efficient fuzzing with neural program smoothing[C]. 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019: 803-817.

[34] Wong, Edmund, et al. Dase: Document-assisted symbolic execution for improving automated software testing[C]. Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on. Vol. 1. IEEE, 2015.

[35] Braione, Pietro, Giovanni Denaro, and Mauro Pezzè. Symbolic execution of programs with heap inputs [C]. Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering. ACM, 2015.

[36] Su, Ting, et al. Combining symbolic execution and model checking for data flow testing[C]. Proceedings of the 37th International Conference on Software Engineering-Volume 1. IEEE Press, 2015.

[37] Christakis, Maria, Peter Müller, and Valentin Wüstholz. Guiding dynamic symbolic execution toward unverified program executions [C]. Proceedings of the 38th International Conference on Software Engineering. ACM, 2016.

[38] Qiu, Rui, et al. Compositional symbolic execution with memoized replay[C]. Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on. Vol. 1. IEEE, 2015.

[39] Stephens N, Grosen J, Salls C, et al. Driller: Augmenting Fuzzing Through Selective Symbolic Execution [C]. NDSS. 2016, 16(2016): 1-16.

[40] Yun I, Lee S, Xu M, et al. {QSYM}: A practical concolic execution engine tailored for hybrid fuzzing [C]. 27th {USENIX} Security Symposium ({USENIX} Security 18). 2018: 745-761.

[41] Ding R, Qian C, Song C, et al. Efficient protection of path-sensitive control security[C]. 26th {USENIX} Security Symposium ({USENIX} Security 17). 2017: 131-148. [2]Enforcing Unique Code Target Property for Control-Flow Integrity, 18.

[42] Khandaker M, Naser A, Liu W, et al. Adaptive Call-site Sensitive Control Flow Integrity[C]. 2019 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2019: 95-110.

[43] Khandaker M R, Liu W, Naser A, et al. Origin-sensitive control flow integrity[C]. 28th {USENIX} Security Symposium ({USENIX} Security 19). 2019: 195-211. [5]Sponge-Based Control-Flow Protection for IoT Devices.

[44] Hu H, Qian C, Yagemann C, et al. Enforcing unique code target property for control-flow integrity[C]. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018: 1470-1486.

[45] Werner M, Unterluggauer T, Schaffenrath D, et al. Sponge-based control-flow protection for iot devices [C]. 2018 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2018: 214-226.

[46] Biondo A, Conti M, Lain D. Back To The Epilogue: Evading Control Flow Guard via Unaligned Targets [C]. NDSS. 2018.

[47] Xu X, Ghaffarinia M, Wang W, et al. {CONFIRM}: Evaluating Compatibility and Relevance of Control-flow Integrity Protections for Modern Software[C]. 28th {USENIX} Security Symposium ({USENIX} Security 19). 2019: 1805-1821.

[48] Proskurin S, Momeu M, Ghavamnia S, et al. xMP: Selective Memory Protection for Kernel and User Space[C]. 2020 IEEE Symposium on Security and Privacy (SP). 2020: 584-598.

[49] Ainsworth S, Jones T M. MarkUs: Drop-in use-after-free prevention for low-level languages[C]. 2020 IEEE Symposium on Security and Privacy (SP). 2020: 860-860.

[50] Wang Z, Wu C, Xie M, et al. SEIMI: Efficient and Secure SMAP-Enabled Intra-process Memory Isolation[J].

[51] Filardo N, Gutstein B F, Woodruff J, et al. Cornucopia: Temporal Safety for CHERI Heaps[C]. 2020 IEEE Symposium on Security and Privacy (SP). 2020: 1507-1524.

[52] Frassetto T, Jauernig P, Liebchen C, et al. {IMIX}: In-Process Memory Isolation EXtension[C]. 27th {USENIX} Security Symposium ({USENIX} Security 18). 2018: 83-97.

[53] Silvestro S, Liu H, Liu T, et al. Guarder: A tunable secure allocator[C]. 27th {USENIX} Security Symposium ({USENIX} Security 18). 2018: 117-133.

[54] Lee H, Song C, Kang B B. Lord of the x86 rings: A portable user mode privilege separation architecture on x86[C]. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018: 1441-1454.

[55] Liu D, Zhang M, Wang H. A robust and efficient defense against use-after-free exploits via concurrent pointer sweeping[C]. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018: 1635-1648.

[56] Kollenda B, Koppe P, Fyrbiak M, et al. An exploratory analysis of microcode as a building block for system defenses[C]. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018: 1649-1666.

[57] Kwon D, Shin J, Kim G, et al. uXOM: Efficient eXecute-Only Memory on {ARM} Cortex-M[C]. 28th {USENIX} Security Symposium ({USENIX} Security 19). 2019: 231-247.

[58] Almakhdhub N S, Clements A A, Bagchi S, et al. μRAI: Securing Embedded Systems with Return Address Integrity[J].

[59] S. K. Fayaz, Y. Tobioka, V. Sekar, and M. Bailey, Bohatei: Flexible and elastic DDoS defense[C]. in USENIX Security Symposium, 2015, pp. 817-832.

[60] Y. Afek, A. Bremler-Barr, and L. Shafir, Network anti-spoofing with SDN data plane[C]. in INFOCOM 2017-IEEE Conference on Computer Communications, IEEE. IEEE, 2017: 1-9.

[61] Li, Xue Jun, Maode Ma, and Narayanan Arjun. An Encryption Algorithm to Prevent Domain Name System Cache Poisoning Attacks[C]. In 2019 29th International Telecommunication Networks and Applications Conference (ITNAC), pp. 1-6. IEEE, 2019.

[62] Levy, Amit, Henry Corrigan-Gibbs, and Dan Boneh. Stickler: Defending against malicious content distribution networks in an unmodified browser[C]. IEEE Security & Privacy 14, no. 2 (2016): 22-28.

[63] Cangialosi, Frank, Taejoong Chung, David Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove, and Christo Wilson. Measurement and analysis of private key sharing in the https ecosystem[C]. In

Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 628-640. 2016.

[64] Ujcich, Benjamin E., Samuel Jero, Anne Edmundson, Qi Wang, Richard Skowyra, James Landry, Adam Bates, William H. Sanders, Cristina Nita-Rotaru, and Hamed Okhravi. Cross-app poisoning in software-defined networking[C]. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 648-663. 2018.

[65] Lee, Seungsoo, Changhoon Yoon, Chanhee Lee, Seungwon Shin, Vinod Yegneswaran, and Phillip A. Porras. DELTA: A Security Assessment Framework for Software-Defined Networks[C]. In NDSS. 2017.

[66] Wang, Haopei, Lei Xu, and Guofei Gu. Floodguard: A dos attack prevention extension in software-defined networks[C]. In 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 239-250. IEEE, 2015.

[67] Lee, Taekjin, et al. FUSE: Finding File Upload Bugs via Penetration Testing[C]. 2020 Network and Distributed System Security Symposium. Network & Distributed System Security Symposium, 2020.

[68] Pellegrino, Giancarlo, et al. Deemon: Detecting CSRF with dynamic analysis and property graphs[C]. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017.

[69] Lekies, Sebastian, et al. Code-reuse attacks for the web: Breaking cross-site scripting mitigations via script gadgets[C]. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017.

[70] Backes, Michael, et al. Efficient and flexible discovery of php application vulnerabilities[C]. 2017 IEEE european symposium on security and privacy (EuroS&P). IEEE, 2017.

[71] Alhuzali, Abeer, et al. {NAVEX}: Precise and Scalable Exploit Generation for Dynamic Web Applications[C]. 27th {USENIX} Security Symposium ({USENIX} Security 18). 2018.

[72] Nguyen, Hoai Viet, Luigi Lo Iacono, and Hannes Federrath. Your cache has fallen: Cache-poisoned denial-of-service attack[C]. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019.

[73] Mirheidari, Seyed Ali, et al. Cached and confused: Web cache deception in the wild[J]. arXiv preprint arXiv: 1912.10190 (2019).

[74] Apostolaki M, Zohar A and Vanbever L, 2017, May. Hijacking bitcoin: Routing attacks on cryptocurrencies [C]. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 375-392). IEEE. MariaApostolaki, GianMarti, JanMüller, andLaurentVanbever. 2018.

[75] Apostolaki M, Marti G, Müller J and Vanbever L, 2018. SABRE: Protecting bitcoin against routing attacks[J]. arXiv preprint arXiv: 1808.06254.

[76] Badertscher C, Gaži P, Kiayias A, Russell A and Zikas V, 2018, January. Ouroboros genesis: Composable proof-of-stake blockchains with dynamic availability[C]. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 913-930).

[77] Bai X, Gao J, Hu C and Zhang L, 2019. Constructing an Adversary Solver for Equihash[C]. In NDSS.

[78] Bartoletti M and Zunino R, 2018, January. BitML: a calculus for Bitcoin smart contracts. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 83-100).

[79] Bijmans H L, Booij T M. and Doerr C, 2019. Inadvertently making cyber criminals rich: A comprehensive study of cryptojacking campaigns at internet scale[C]. In 28th {USENIX} Security Symposium ({USENIX} Security 19) (pp. 1627-1644).

[80] Campanelli M, Gennaro R, Goldfeder, S. and Nizzardo, L., 2017, October. Zero-knowledge contingent payments revisited: Attacks and payments for services[C]. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 229-243).

[81] Chen T, Zhang Y, Li Z, Luo X, Wang T, Cao R, Xiao X and Zhang X, 2019, November. TokenScope: Automatically detecting inconsistent behaviors of cryptocurrency tokens in Ethereum[C]. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (pp. 1503-1520).

[82] Das P, Eckey L, Frassetto T, Gens D, Hostáková K, Jauernig P, Faust S and Sadeghi A R, 2019. Fastkitten: Practical smart contracts on bitcoin[C]. In 28th {USENIX} Security Symposium ({USENIX} Security 19) (pp. 801-818).

[83] Derler D, Samelin K, Slamanig D and Striecks C, 2019. Fine-Grained and Controlled Rewriting in Blockchains: Chameleon-Hashing Gone Attribute-Based[C]. IACR Cryptol. ePrint Arch., 2019, p. 406.

[84] Deuber D, Magri B and Thyagarajan S A K, 2019, May. Redactable blockchain in the permissionless setting[C]. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 124-138). IEEE.

[85] Dziembowski S, Eckey L and Faust, S, 2018, January. Fairswap: How to fairly exchange digital goods [C]. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 967-984).

[86] Fu Y, Ren M, Ma F, Jiang Y, Shi H and Sun J, 2019. Evmfuzz: Differential fuzz testing of ethereum virtual machine[J]. arXiv preprint arXiv: 1903.08483.

[87] Hong G, Yang Z, Yang S, Zhang L, Nan Y, Zhang Z, Yang M, Zhang Y, Qian Z and Duan H, 2018, January. How you get shot in the back: A systematical study about cryptojacking in the real world[C]. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 1701-1713).

[88] Kappos G, Yousaf H, Maller M and Meiklejohn S, 2018. An empirical analysis of anonymity in zcash [C]. In 27th {USENIX} Security Symposium ({USENIX} Security 18) (pp. 463-477).

[89] Kerber T, Kiayias A, Kohlweiss M and Zikas V, 2019, May. Ouroboros crypsinous: Privacy-preserving proof-of-stake[C]. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 157-174). IEEE.

[90] Kokoris-Kogias E, Jovanovic P, Gasser L, Gailly N, Syta E and Ford B, 2018, May. Omniledger: A secure, scale-out, decentralized ledger via sharding[C]. In 2018 IEEE Symposium on Security and Privacy (SP) (pp. 583-598). IEEE.

[91] Lee S, Yoon C, Kang H, Kim Y, Kim Y, Han D, Son S and Shin S, 2019, February. Cybercriminal minds: an investigative study of cryptocurrency abuses in the dark web[C]. In Network and Distributed System Security Symposium (pp. 1-15). Internet Society.

[92] Luu L, Narayanan V, Zheng C, Baweja K, Gilbert S and Saxena P, 2016, October. A secure sharding protocol for open blockchains[C]. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 17-30).

[93] Xu J and Livshits B, 2019. The anatomy of a cryptocurrency pump-and-dump scheme[C]. In 28th {USENIX} Security Symposium ({USENIX} Security 19) (pp. 1609-1625).

[94] Yousaf H, Kappos G and Meiklejohn S, 2019. Tracing transactions across cryptocurrency ledgers[C]. In 28th {USENIX} Security Symposium ({USENIX} Security 19) (pp. 837-850).

[95] Zamani M, Movahedi M and Raykova M, 2018, January. Rapidchain: Scaling blockchain via full

sharding[C]. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 931-948).

[96] Zhang R and Preneel B, 2019, May. Lay down the common metrics: Evaluating proof-of-work consensus protocols' security[C]. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 175-192). IEEE.

[97] Tramèr F, Boneh D and Paterson K G, 2020. Remote Side-Channel Attacks on Anonymous Transactions [C]. IACR Cryptol. ePrint Arch., 2020, p. 220.

[98] Frank J, Aschermann C, Holz T, Hu S M, Zhang Z, Sagonas K, Song L, Somorovsky J, Wang G, Zhou X and Liu Y, 2020. An Ever-evolving Game: Evaluation of Real-world Attacks and Defenses in Ethereum Ecosystem[C]. In 29th {USENIX} Security Symposium ({USENIX} Security 20).

[99] Yu H, Nikolic I, Hou R and Saxena P, 2018. OHIE: blockchain scaling made simple[J]. arXiv preprint arXiv: 1811.12628.

[100] Ekparinya P, Gramoli V and Jourjon G, 2019. The attack of the clones against proof-of-authority[J]. arXiv preprint arXiv: 1902.10244.

[101] 徐蜜雪, 苑超, 王永娟, 付金华, & 李斌. (2019). 拟态区块链——区块链安全解决方案[J]. 软件学报(6).

[102] 叶聪聪, 李国强, 蔡鸿明, & 顾永跟. (2018). 区块链的安全检测模型[J]. 软件学报, 029 (005), 1348-1359.

[103] Grech N, Kong M, Jurisevic A, Brent L, Scholz B and Smaragdakis Y, 2018. Madmax: Surviving out-of-gas conditions in ethereum smart contracts[C]. Proceedings of the ACM on Programming Languages, 2(OOPSLA), pp. 1-27.

[104] Kalra S, Goel S, Dhawan M and Sharma S, 2018, February. ZEUS: Analyzing Safety of Smart Contracts [C]. In NDSS.

[105] Krupp J and Rossow C, 2018. teether: Gnawing at ethereum to automatically exploit smart contracts [C]. In 27th {USENIX} Security Symposium ({USENIX} Security 18) (pp. 1317-1333).

[106] Luu L, Chu D H, Olickel H, Saxena P and Hobor A, 2016, October. Making smart contracts smarter [C]. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 254-269).

[107] Rodler M, Li W, Karame G O and Davi L, 2018. Sereum: Protecting existing smart contracts against re-entrancy attacks[J]. arXiv preprint arXiv: 1812.05934.

[108] Torres C F and Steichen M, 2019. The art of the scam: Demystifying honeypots in ethereum smart contracts[C]. In 28th {USENIX} Security Symposium ({USENIX} Security 19) (pp. 1591-1607).

[109] Tsankov P, Dan A, Drachsler-Cohen D, Gervais A, Buenzli F and Vechev M, 2018, January. Securify: Practical security analysis of smart contracts[C]. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 67-82).

[110] Perez D and Livshits B, 2019. Broken metre: Attacking resource metering in evm[J]. arXiv preprint arXiv: 1909.07220.

[111] Cheng R, Zhang F, Kos J, He W, Hynes N, Johnson N, Juels A, Miller A and Song D, 2019, June. Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts[C]. In 2019 IEEE European Symposium on Security and Privacy (EuroS&P) (pp. 185-200). IEEE.

[112] Kosba A, Miller A, Shi E, Wen Z and Papamanthou C, 2016, May. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts[C]. In 2016 IEEE symposium on security and

privacy (SP) (pp. 839-858). IEEE.

[113] Jiang B, Liu Y and Chan W K, 2018, September. Contractfuzzer: Fuzzing smart contracts for vulnerability detection[C]. In 2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE) (pp. 259-269). IEEE.

[114] Hu W, Liao Y, & Vemuri V R (2003, June). Robust anomaly detection using support vector machines [C]. In Proceedings of the international conference on machine learning (pp. 282-289).

[115] Choi Y H, Liu P, Shang Z, Wang H, Wang Z, Zhang L,... & Zou, Q. (2019). Using Deep Learning to Solve Computer Security Challenges: A Survey[J]. arXiv preprint arXiv: 1912.05721.

[116] He K, Zhang X, Ren S, & Sun J (2016). Deep residual learning for image recognition[C]. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[117] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez, A N,... & Polosukhin I (2017). Attention is all you need[C]. In Advances in neural information processing systems (pp. 5998-6008).

[118] Taigman Y, Yang M, Ranzato M A, & Wolf L (2014). Deepface: Closing the gap to human-level performance in face verification[C]. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1701-1708).

[119] LeCun Y, Bengio Y, & Hinton G (2015). Deep learning[C]. nature, 521(7553), 436-444.

[120] 张玉清, 董颖, 柳彩云, 雷柯楠, 孙鸿宇. (2018). 深度学习应用于网络空间安全的现状、趋势与展望[J]. 计算机研究与发展, 55(06), 3-28.

[121] McGraw G, & Morrisett G (2000). Attacking malicious code: A report to the infosec research council [J]. IEEE software, 17(5), 33-41.

[122] 刘剑, 苏璞睿, 杨珉, 和亮, 张源, 朱雪阳, 林惠民. (2018). 软件与网络安全研究综述[J]. 软件学报, 29(1), 42-68.

[123] 中国互联网协会公布恶意软件定义(征求意见稿), https://www.isc.org.cn/hdzt/feyrj/listinfo-4190.html.

[124] Enck W, Gilbert P, Han S, Tendulkar V, Chun B G, Cox L P,... & Sheth, A. N. (2014). TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones[J]. ACM Transactions on Computer Systems (TOCS), 32(2), 1-29.

[125] Christodorescu M, Jha S, & Kruegel C (2007, September). Mining specifications of malicious behavior [C]. In Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering (pp. 5-14).

[126] Nix R, & Zhang J (2017, May). Classification of android apps and malware using deep neural networks [C]. In 2017 International joint conference on neural networks (IJCNN) (pp. 1871-1878). IEEE.

[127] Cui Z, Xue F, Cai X, Cao Y, Wang G G, & Chen J (2018). Detection of malicious code variants based on deep learning[C]. IEEE Transactions on Industrial Informatics, 14(7), 3187-3196.

[128] Saxe J, & Berlin K (2015, October). Deep neural network based malware detection using two dimensional binary program features[C]. In 2015 10th International Conference on Malicious and Unwanted Software (MALWARE) (pp. 11-20). IEEE.

[129] Rosli N A, Yassin W, Faizal M A, & Selamat S R. Clustering Analysis for Malware Behavior Detection using Registry Data.

[130] Zhang M, Duan Y, Yin H, & Zhao Z (2014, November). Semantics-aware android malware classification using weighted contextual api dependency graphs[C]. In Proceedings of the 2014 ACM

SIGSAC conference on computer and communications security (pp. 1105-1116).

[131] Xue S, Zhang L, Li A, Li X Y, Ruan C, & Huang W (2018, April). Appdna: App behavior profiling via graph- based deep learning [C]. In IEEE INFOCOM 2018- IEEE Conference on Computer Communications (pp. 1475-1483). IEEE.

[132] Mariconti E, Onwuzurike L, Andriotis P, De Cristofaro E, Ross G, & Stringhini G (2016). Mamadroid: Detecting android malware by building markov chains of behavioral models[J]. arXiv preprint arXiv: 1612. 04433.

[133] Jordaney R, Sharad K, Dash S K, Wang Z, Papini D, Nouretdinov I, & Cavallaro L (2017). Transcend: Detecting concept drift in malware classification models[C]. In 26th {USENIX} Security Symposium ({USENIX} Security 17) (pp. 625-642).

[134] Xu K, Li Y, Deng R, Chen K, & Xu J (2019, June). Droidevolver: Self- evolving android malware detection system[C]. In 2019 IEEE European Symposium on Security and Privacy (EuroS&P) (pp. 47-62). IEEE.

[135] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, & Fergus R (2013). Intriguing properties of neural networks[J]. arXiv: Computer Vision and Pattern Recognition.

[136] Grosse K, Papernot N, Manoharan P, Backes M, & McDaniel P (2017, September). Adversarial examples for malware detection[C]. In European Symposium on Research in Computer Security (pp. 62-79). Springer, Cham.

[137] Arp D, Spreitzenbarth M, Hubner M, Gascon H, Rieck K, & Siemens C E R T (2014, February). Drebin: Effective and explainable detection of android malware in your pocket[C]. In Ndss (Vol. 14, pp. 23-26).

[138] Guo W, Mu D, Xu J, Su P, Wang G, & Xing X (2018, January). Lemna: Explaining deep learning based security applications[C]. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 364-379).

[139] Du M, Li F, Zheng G, & Srikumar V (2017, October). Deeplog: Anomaly detection and diagnosis from system logs through deep learning[C]. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 1285-1298).

[140] Melicher W, Ur B, Segreti S M, Komanduri S, Bauer L, Christin N, & Cranor L F (2016). Fast, lean, and accurate: Modeling password guessability using neural networks[C]. In 25th {USENIX} Security Symposium ({USENIX} Security 16) (pp. 175-191).

[141] Putchala M K (2017). Deep learning approach for intrusion detection system (ids) in the internet of things (iot) network using gated recurrent neural networks (gru).

[142] Xu X, Liu C, Feng Q, Yin H, Song L, & Song D (2017, October). Neural network- based graph embedding for cross- platform binary code similarity detection[C]. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 363-376).

[143] Michalas A, & Murray R (2017, October). MemTri: A memory forensics triage tool using bayesian network and volatility[C]. In Proceedings of the 2017 International Workshop on Managing Insider Security Threats (pp. 57-66).

[144] Ding D, Zhang M, Pan X, Yang M, & He X (2019, July). Modeling extreme events in time series prediction[C]. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1114-1122).

[145] 陈宇飞, 沈超, 王骞, 李琦, 王聪, 纪守领, 李康, 管晓宏. 人工智能系统安全与隐私风险[J]. 计算机研究与发展, 2019, 56(10): 2135-2150.

[146] 何英哲, 胡兴波, 何锦雯, 孟国柱, 陈恺. 机器学习系统的隐私和安全问题综述[J]. 计算机研究与发展, 2019, 56(10): 2049-2070.

[147] 纪守领, 杜天宇, 李进锋, 沈超, 李博. 机器学习模型安全与隐私研究综述[J]. 软件学报. http://www. jos. org. cn/1000-9825/0000. htm.

[148] Papernot N, McDaniel P, Sinha A and Wellman M P, 2018, April. SoK: Security and privacy in machine learning[C]. In 2018 IEEE European Symposium on Security and Privacy (EuroS&P) (pp. 399-414). IEEE.

[149] Biggio B and Roli F, 2018. Wild patterns: Ten years after the rise of adversarial machine learning[C]. Pattern Recognition, 84, pp. 317-331.

[150] 谭作文, 张连福. 机器学习隐私保护研究综述[J]. 软件学报. http://www. jos. org. cn/1000-9825/6052. html.

[151] 刘睿瑄, 陈红, 郭若杨, 赵丹, 梁文娟, 李翠平. 机器学习中的隐私攻击与防御[J]. 软件学报, 2020, 31(3): 866-892. http://www. jos. org. cn/1000-9825/5904. htm.

[152] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I and Fergus R, 2013. Intriguing properties of neural networks[J]. arXiv preprint arXiv: 1312. 6199.

[153] Goodfellow I J, Shlens J and Szegedy C, 2014. Explaining and harnessing adversarial examples. arXiv preprint arXiv: 1412. 6572.

[154] Kurakin A, Goodfellow I and Bengio S, 2016. Adversarial examples in the physical world[J]. arXiv preprint arXiv: 1607. 02533.

[155] Moosavi-Dezfooli S M, Fawzi A and Frossard P, 2016. Deepfool: a simple and accurate method to fool deep neural networks[C]. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2574-2582).

[156] Wiyatno R and Xu A, 2018. Maximal jacobian-based saliency map attack[J]. arXiv preprint arXiv: 1808. 07945.

[157] Carlini N and Wagner D, 2017, May. Towards evaluating the robustness of neural networks[C]. In 2017 ieee symposium on security and privacy (sp) (pp. 39-57). IEEE.

[158] Carlini N and Wagner D, 2018, May. Audio adversarial examples: Targeted attacks on speech-to-text [C]. In 2018 IEEE Security and Privacy Workshops (SPW) (pp. 1-7). IEEE.

[159] Ren S, Deng Y, He K and Che W, 2019, July. Generating natural language adversarial examples through probability weighted word saliency[C]. In Proceedings of the 57th annual meeting of the association for computational linguistics (pp. 1085-1097).

[160] Jin D, Jin Z, Tianyi Zhou J and Szolovits P, 2019. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment[J]. arXiv, pp. arXiv-1907.

[161] Cheng M, Yi J, Chen P Y, Zhang H and Hsieh C J, 2020. Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples[C]. In AAAI (pp. 3601-3608).

[162] Zajac M, Zołna K, Rostamzadeh N and Pinheiro P O, 2019, July. Adversarial framing for image and video classification[C]. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, pp. 10077-10078).

[163] Pierazzi F, Pendlebury F, Cortellazzi J and Cavallaro L, 2019. Intriguing Properties of Adversarial ML

Attacks in the Problem Space[J]. arXiv preprint arXiv: 1911.02142.

[164] Yang W, Kong D, Xie T and Gunter C A, 2017, December. Malware detection in adversarial settings: Exploiting feature evolutions and confusions in android apps[C]. In Proceedings of the 33rd Annual Computer Security Applications Conference (pp. 288-302).

[165] 王文琦, 汪润, 王丽娜, 唐奔宵. 面向中文文本倾向性分类的对抗样本生成方法[J]. 软件学报, 2019, 30(8): 2415-2427. http://www.jos.org.cn/1000-9825/5765.htm.

[166] 马玉琨, 毋立芳, 简萌, 刘方昊, 杨洲. 一种面向人脸活体检测的对抗样本生成算法[J]. 软件学报, 2019, 30(2): 469-480. http://www.jos.org.cn/1000-9825/5568.htm.

[167] Athalye A, Carlini N and Wagner D, 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples[J]. arXiv preprint arXiv: 1802.00420.

[168] Guo C, Rana M, Cisse M and Van Der Maaten L, 2017. Countering adversarial images using input transformations[J]. arXiv preprint arXiv: 1711.00117.

[169] Dhillon G S, Azizzadenesheli K, Lipton Z C, Bernstein J, Kossaifi J, Khanna A and Anandkumar A, 2018. Stochastic activation pruning for robust adversarial defense[J]. arXiv preprint arXiv: 1803.01442.

[170] Xie C, Wang J, Zhang Z, Ren Z and Yuille A, 2017. Mitigating adversarial effects through randomization [J]. arXiv preprint arXiv: 1711.01991.

[171] Madry A, Makelov A, Schmidt L, Tsipras D and Vladu A, 2017. Towards deep learning models resistant to adversarial attacks[J]. arXiv preprint arXiv: 1706.06083.

[172] Wong E and Kolter Z, 2018, July. Provable defenses against adversarial examples via the convex outer adversarial polytope[C]. In International Conference on Machine Learning (pp. 5286-5295).

[173] Al-Dujaili A, Srikant S, Hemberg E and O'Reilly, U. M., 2019, February. On the application of Danskin's theorem to derivative-free minimax problems[C]. In AIP Conference Proceedings (Vol. 2070, No. 1, p. 020026). AIP Publishing LLC.

[174] Tsipras D, Santurkar S, Engstrom L, Turner A and Madry A, 2018. Robustness may be at odds with accuracy[J]. arXiv preprint arXiv: 1805.12152.

[175] Kurakin A, Goodfellow I and Bengio S, 2016. Adversarial machine learning at scale[J]. arXiv preprint arXiv: 1611.01236.

[176] Li J, Ji S, Du T, Li B and Wang T, 2018. Textbugger: Generating adversarial text against real-world applications[J]. arXiv preprint arXiv: 1812.05271.

[177] Ling X, Ji S, Zou J, Wang J, Wu C, Li B and Wang T, 2019, May. Deepsec: A uniform platform for security analysis of deep learning model[C]. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 673-690). IEEE.

[178] Li J, Du T, Ji S, Zhang R, Lu Q, Yang M and Wang T, 2020. TextShield: Robust Text Classification Based on Multimodal Embedding and Neural Machine Translation[C]. In 29th {USENIX} Security Symposium ({USENIX} Security 20).

[179] Zhang X, Wang N, Shen H, Ji S, Luo X and Wang T, 2020. Interpretable deep learning under fire [C]. In 29th {USENIX} Security Symposium ({USENIX} Security 20).

[180] Biggio B, Nelson B and Laskov P, 2012. Poisoning attacks against support vector machines[J]. arXiv preprint arXiv: 1206.6389.

[181] Li B, Wang Y, Singh A and Vorobeychik Y, 2016. Data poisoning attacks on factorization-based collaborative filtering[C]. In Advances in neural information processing systems (pp. 1885-1893).

[182] Zhang H, Zheng T, Gao J, Miao C, Su L, Li Y and Ren K, 2019. Data poisoning attack against knowledge graph embedding[J]. arXiv preprint arXiv: 1904.12052.

[183] Shafahi A, Huang W R, Najibi M, Suciu O, Studer C, Dumitras T and Goldstein T, 2018. Poison frogs! targeted clean-label poisoning attacks on neural networks[C]. In Advances in Neural Information Processing Systems (pp. 6103-6113).

[184] Laishram R and Phoha V V, 2016. Curie: A method for protecting SVM Classifier from Poisoning Attack [J]. arXiv preprint arXiv: 1606.01584.

[185] Suciu O, Marginean R, Kaya Y, Daume III H and Dumitras T, 2018. When does machine learning {FAIL}? generalized transferability for evasion and poisoning attacks[C]. In 27th {USENIX} Security Symposium ({USENIX} Security 18) (pp. 1299-1316).

[186] Saha A, Subramanya A and Pirsiavash H, 2019. Hidden trigger backdoor attacks[J]. arXiv preprint arXiv: 1910.00033.

[187] Ji Y, Zhang X, Ji S, Luo X and Wang T, 2018, January. Model-reuse attacks on deep learning systems [C]. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 349-363).

[188] Koh P W and Liang P, 2017. Understanding black-box predictions via influence functions[J]. arXiv preprint arXiv: 1703.04730.

[189] Hara S, Nitanda A and Maehara T, 2019. Data Cleansing for Models Trained with SGD[C]. In Advances in Neural Information Processing Systems (pp. 4213-4222).

[190] Steinhardt J, Koh P W W and Liang P S, 2017. Certified defenses for data poisoning attacks[C]. In Advances in neural information processing systems (pp. 3517-3529).

[191] Nguyen N H and Tran T D, 2012. Robust lasso with missing and grossly corrupted observations[C]. IEEE transactions on information theory, 59(4), pp. 2036-2058.

[192] Pang R, Shen H, Zhang X, Ji S, Vorobeychik Y, Luo X, Liu A and Wang T, 2020, January. A Tale of Evil Twins: Adversarial Inputs versus Poisoned Models[C]. In Proceedings of the ACM Conference on Computer and Communications Security.

[193] Bhatia K, Jain P and Kar P, 2015. Robust regression via hard thresholding[C]. In Advances in Neural Information Processing Systems (pp. 721-729).

[194] Jagielski M, Oprea A, Biggio B, Liu C, Nita-Rotaru C and Li B, 2018, May. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning[C]. In 2018 IEEE Symposium on Security and Privacy (SP) (pp. 19-35). IEEE.

[195] Awasthi P, Balcan M F and Long P M, 2014, May. The power of localization for efficiently learning linear separators with noise[C]. In Proceedings of the forty-sixth annual ACM symposium on Theory of computing (pp. 449-458).

[196] Klivans A R, Long P M and Servedio R A, 2009. Learning Halfspaces with Malicious Noise[J]. Journal of Machine Learning Research, 10(12).

[197] Diakonikolas I, Kamath G, Kane D, Li J, Moitra A and Stewart A, 2019. Robust estimators in high-dimensions without the computational intractability[C]. SIAM Journal on Computing, 48(2), pp. 742-864..

[198] Lai K A, Rao A B and Vempala S, 2016, October. Agnostic estimation of mean and covariance[C]. In 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS) (pp. 665-674).

IEEE.

[199] Biggio B, Corona I, Fumera G, Giacinto G and Roli F, 2011, June. Bagging classifiers for fighting poisoning attacks in adversarial classification tasks[C]. In International workshop on multiple classifier systems (pp. 350-359). Springer, Berlin, Heidelberg.

[200] Cretu G F, Stavrou A, Locasto M E, Stolfo S J and Keromytis A D, 2008, May. Casting out demons: Sanitizing training data for anomaly sensors[C]. In 2008 IEEE Symposium on Security and Privacy (sp 2008) (pp. 81-95). IEEE.

[201] Newell A, Potharaju R, Xiang L and Nita-Rotaru C, 2014, November. On the practicality of integrity attacks on document-level sentiment analysis[C]. In Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop (pp. 83-93).

[202] Gu T, Liu K, Dolan-Gavitt B and Garg S, 2019. Badnets: Evaluating backdooring attacks on deep neural networks[C]. IEEE Access, 7, pp. 47230-47244.

[203] Liu Y, Ma S, Aafer Y, Lee W C, Zhai J, Wang W and Zhang X. 2017. Trojaning attack on neural networks[C].

[204] Su J, Vargas D V and Sakurai K, 2019. One pixel attack for fooling deep neural networks[C]. IEEE Transactions on Evolutionary Computation, 23(5), pp. 828-841.

[205] Chen X, Liu C, Li B, Lu K and Song D, 2017. Targeted backdoor attacks on deep learning systems using data poisoning[J]. arXiv preprint arXiv: 1712.05526.

[206] Chen B, Carvalho W, Baracaldo N, Ludwig H, Edwards B, Lee T, Molloy I and Srivastava B, 2018. Detecting backdoor attacks on deep neural networks by activation clustering[J]. arXiv preprint arXiv: 1811.03728.

[207] Tran B, Li J and Madry A, 2018. Spectral signatures in backdoor attacks[C]. In Advances in Neural Information Processing Systems (pp. 8000-8010).

[208] Wang B, Yao Y, Shan S, Li H, Viswanath B, Zheng H and Zhao B Y, 2019, May. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks[C]. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 707-723). IEEE.

[209] Udeshi S, Peng S, Woo G, Loh L, Rawshan L and Chattopadhyay S, 2019. Model agnostic defence against backdoor attacks in machine learning[J]. arXiv preprint arXiv: 1908.02203.

[210] Liu K, Dolan-Gavitt B and Garg S, 2018 September Fine-pruning: Defending against backdooring attacks on deep neural networks[C]. In International Symposium on Research in Attacks, Intrusions, and Defenses (pp. 273-294). Springer, Cham.

[211] Q Qiao X, Yang Y and Li H, 2019. Defending neural backdoors via generative distribution modeling [C]. In Advances in Neural Information Processing Systems (pp. 14004-14013).

[212] Yao Y, Li H, Zheng H and Zhao B Y, 2019, November. Latent backdoor attacks on deep neural networks[C]. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (pp. 2041-2055).

[213] Mhamdi E M E, Guerraoui R and Rouault S, 2018. The hidden vulnerability of distributed learning in byzantium[J]. arXiv preprint arXiv: 1802.07927.

[214] Yin D, Chen Y, Ramchandran K and Bartlett P, 2018. Byzantine-robust distributed learning: Towards optimal statistical rates[J]. arXiv preprint arXiv: 1803.01498.

[215] Feng J, Xu H and Mannor S, 2014. Distributed robust learning. arXiv preprint arXiv: 1409.5937.

[216] Chen Y, Su L and Xu J, 2017. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent[J]. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 1 (2), pp. 1-25.

[217] Blanchard P, Guerraoui R and Stainer J, 2017. Machine learning with adversaries: Byzantine tolerant gradient descent[C]. In Advances in Neural Information Processing Systems (pp. 119-129).

[218] Fang M, Cao X, Jia J and Gong N Z, 2020. Local model poisoning attacks to Byzantine-robust federated learning[C]. In 29th {USENIX} Security Symposium ({USENIX} Security 20).

[219] Xie C, Koyejo S and Gupta I, 2019, May. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance[C]. In International Conference on Machine Learning (pp. 6893-6901).

[220] Pan X, Zhang M, Wu D, Xiao Q, Ji S and Yang M, 2020. Justinian's GAAvernor: Robust Distributed Learning with Gradient Aggregation Agent[C]. In 29th {USENIX} Security Symposium ({USENIX} Security 20).

[221] Bagdasaryan E, Veit A, Hua Y, Estrin D and Shmatikov V, 2020, June. How to backdoor federated learning[C]. In International Conference on Artificial Intelligence and Statistics (pp. 2938-2948).

[222] Nasr M, Shokri R and Houmansadr A, 2018, January. Machine learning with membership privacy using adversarial regularization[C]. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 634-646).

[223] Shokri R, Stronati M, Song C and Shmatikov V, 2017, May. Membership inference attacks against machine learning models[C]. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 3-18). IEEE.

[224] Song L, Shokri R and Mittal P, 2019, November. Privacy risks of securing machine learning models against adversarial examples[C]. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (pp. 241-257).

[225] Salem A, Zhang Y, Humbert M, Berrang P, Fritz M and Backes M, 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models[J]. arXiv preprint arXiv: 1806.01246.

[226] Ganju K, Wang Q, Yang W, Gunter C A and Borisov N, 2018, January. Property inference attacks on fully connected neural networks using permutation invariant representations[C]. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 619-633).

[227] Melis L, Song C, De Cristofaro E and Shmatikov V, 2019, May. Exploiting unintended feature leakage in collaborative learning[C]. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 691-706). IEEE.

[228] Pan X, Zhang M, Ji S and Yang M, 2020. Privacy Risks of General-Purpose Language Models[C]. In 2020 IEEE Symposium on Security and Privacy (SP) (pp. 1471-1488).

[229] Fredrikson M, Jha S and Ristenpart T, 2015, October. Model inversion attacks that exploit confidence information and basic countermeasures[C]. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (pp. 1322-1333).

[230] Zhu L, Liu Z and Han S, 2019. Deep leakage from gradients[C]. In Advances in Neural Information Processing Systems (pp. 14774-14784).

[231] Salem A, Bhattacharya A, Backes M, Fritz M and Zhang Y, 2019. Updates-leak: Data set inference and reconstruction attacks in online learning[J]. arXiv preprint arXiv: 1904.01067.

[232] Tramèr F, Zhang F, Juels A, Reiter M K and Ristenpart T, 2016. Stealing machine learning models via prediction apis[C]. In 25th {USENIX} Security Symposium ({USENIX} Security 16) (pp. 601-618).

[233] Duddu V, Samanta D, Rao D V and Balas V E, 2018. Stealing neural networks via timing side channels [J]. arXiv preprint arXiv: 1812.11720.

[234] Wang B and Gong N Z, 2018, May. Stealing hyperparameters in machine learning[C]. In 2018 IEEE Symposium on Security and Privacy (SP) (pp. 36-52). IEEE.

[235] Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan H B, Patel S, Ramage D, Segal A and Seth K, 2017, October. Practical secure aggregation for privacy-preserving machine learning[C]. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 1175-1191).

[236] Halevi S, Lindell Y and Pinkas B, 2011, August. Secure computation on the web: Computing without simultaneous interaction[C]. In Annual Cryptology Conference (pp. 132-150). Springer, Berlin, Heidelberg.

[237] Yang Q, Liu Y, Chen T and Tong Y, 2019. Federated machine learning: Concept and applications [C]. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), pp. 1-19.

[238] Bonawitz K, Eichner H, Grieskamp W, Huba D, Ingerman A, Ivanov V, Kiddon C, Konečný J, Mazzocchi S, McMahan H B and Van Overveldt T, 2019. Towards federated learning at scale: System design[J]. arXiv preprint arXiv: 1902.01046.

[239] Kumar N, Rathee M, Chandran N, Gupta D, Rastogi A, & Sharma R (2019). CrypTFlow: Secure TensorFlow Inference[J]. arXiv: Cryptography and Security.

[240] Kumar N, Rathee M, Chandran N, Gupta D, Rastogi A and Sharma R, 2019. Cryptflow: Secure tensorflow inference[J]. arXiv preprint arXiv: 1909.07814.

[241] 徐剑, 王安迪, 毕猛, 周福才. 支持隐私保护的 k 近邻分类器[J]. 软件学报, 2019, 30(11): 3503-3517. http://www.jos.org.cn/1000-9825/5573.htm.

[242] Shokri R and Shmatikov V, 2015, October. Privacy-preserving deep learning[C]. In Proceedings of the 22nd ACM SIGSAC conference on computer and communications security (pp. 1310-1321).

[243] Abadi M, Chu A, Goodfellow I, McMahan H B, Mironov I, Talwar K and Zhang L, 2016, October. Deep learning with differential privacy[C]. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 308-318).

[244] Duchi J C, Jordan M I and Wainwright M J, 2013, October. Local privacy and statistical minimax rates [C]. In 2013 IEEE 54th Annual Symposium on Foundations of Computer Science (pp. 429-438). IEEE.

[245] Dwork C, Naor M, Pitassi T and Rothblum G N, 2010, June. Differential privacy under continual observation [C]. In Proceedings of the forty-second ACM symposium on Theory of computing (pp. 715-724).

[246] Dwork C, 2008, April. Differential privacy: A survey of results[C]. In International conference on theory and applications of models of computation (pp. 1-19). Springer, Berlin, Heidelberg.

[247] Salamatian S, Zhang A, du Pin Calmon F, Bhamidipati S, Fawaz N, Kveton B, Oliveira P and Taft N, 2015. Managing your private and public data: Bringing down inference attacks against your privacy[C]. IEEE Journal of Selected Topics in Signal Processing, 9(7), pp. 1240-1255.

[248] Ohrimenko O, Schuster F, Fournet C, Mehta A, Nowozin S, Vaswani K and Costa M, 2016. Oblivious

multi-party machine learning on trusted processors[C]. In 25th {USENIX} Security Symposium ({USENIX} Security 16) (pp. 619-636).

[249] Hunt T, Zhu Z, Xu Y, Peter S and Witchel E, 2018. Ryoan: A distributed sandbox for untrusted computation on secret data[C]. ACM Transactions on Computer Systems (TOCS), 35(4), pp. 1-32.

[250] Huang H, Zhu S, Chen K, et al. From system services freezing to system server shutdown in android: All you need is a loop in an app[C]. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015: 1236-1247.

[251] Cao C, Gao N, Liu P, et al. Towards analyzing the input validation vulnerabilities associated with android system services[C]. Proceedings of the 31st Annual Computer Security Applications Conference. 2015: 361-370.

[252] Zhang L, Yang Z, He Y, et al. Invetter: Locating insecure input validations in android services[C]. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018: 1165-1178.

[253] Chen P, Chen H. Angora: Efficient fuzzing by principled search[C]. 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018: 711-725.

[254] Chen P, Liu J, Chen H. Matryoshka: fuzzing deeply nested branches[C]. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019: 499-513.

[255] Lyu C, Ji S, Zhang C, et al. {MOPT}: Optimized mutation scheduling for fuzzers[C]. 28th {USENIX} Security Symposium ({USENIX} Security 19). 2019: 1949-1966.

[256] Yue T, Wang P, Tang Y, et al. EcoFuzz: Adaptive Energy-Saving Greybox Fuzzing as a Variant of the Adversarial Multi- Armed Bandit[C]. 29th {USENIX} Security Symposium ({USENIX} Security 20). 2020.

[257] Chen Y, Jiang Y, Ma F, et al. Enfuzz: Ensemble fuzzing with seed synchronization among diverse fuzzers[C]. 28th {USENIX} Security Symposium ({USENIX} Security 19). 2019: 1967-1983.

[258] Liu B, Zhang C, Gong G, et al. {FANS}: Fuzzing Android Native System Services via Automated Interface Analysis[C]. 29th {USENIX} Security Symposium ({USENIX} Security 20). 2020.

[259] Cao M, Hou X, Wang T, et al. Different is Good: Detecting the Use of Uninitialized Variables through Differential Replay[C]. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019: 1883-1897.

[260] Chen Y, Su T, Su Z. Deep differential testing of JVM implementations[C]. 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE). IEEE, 2019: 1257-1268.

[261] Jiang B, Liu Y, Chan W K. Contractfuzzer: Fuzzing smart contracts for vulnerability detection[C]. 2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2018: 259-269.

[262] Zong P, Lv T, Wang D, et al. FuzzGuard: Filtering out Unreachable Inputs in Directed Grey- box Fuzzing through Deep Learning[J].

[263] Wu W, Chen Y, Xu J, et al. {FUZE}: Towards facilitating exploit generation for kernel use-after-free vulnerabilities[C]. 27th {USENIX} Security Symposium ({USENIX} Security 18). 2018: 781-797.

[264] 郑炜, 陈军正, 吴潇雪, 陈翔, 夏鑫. 基于深度学习的安全缺陷报告预测方法实证研究[J]. 软件学报, 2020, 31(05): 1294-1313.

[265] 傅立国, 庞建民, 王军, 张家豪, 岳峰. 二进制翻译正确性及优化方法的形式化模型[J]. 计算机

研究与发展, 2019, 56(09): 2001-2011.

[266] 卢帅兵, 张明, 林哲超, 李虎, 况晓辉, 赵刚. 基于动态二进制翻译和插桩的函数调用跟踪[J]. 计算机研究与发展, 2019, 56(02): 421-430.

[267] 邵思豪, 高庆, 马森, 等. 缓冲区溢出漏洞分析技术研究进展[J]. Journal of Software, 2018, 29(5).

[268] 张贵民, 李清宝, 曾光裕, 等. 运行时代码随机化防御代码复用攻击[J]. 软件学报, 2019(9): 14.

[269] PDiff: Semantic-based Patch Presence Testing for Downstream Kernels. Zheyue Jiang, Yuan Zhang, Jun Xu, Qi Wen, Zhenghe Wang, Xiaohan Zhang, Xinyu Xing, Min Yang, Zhemin Yang[C]. In Proceedings of the 27th ACM Conference on Computer and Communications Security, CCS, Orlando, USA, November 9-13, 2020 (conditionally accepted).

[270] Chen, Jianjun, Xiaofeng Zheng, Hai-Xin Duan, Jinjin Liang, Jian Jiang, Kang Li, Tao Wan, and Vern Paxson. Forwarding-Loop Attacks in Content Delivery Networks[C]. In NDSS. 2016.

[271] Guo, Run, Jianjun Chen, Baojun Liu, Jia Zhang, Chao Zhang, Haixin Duan, Tao Wan, Jian Jiang, Shuang Hao, and Yaoqi Jia. Abusing CDNs for fun and profit: Security issues in CDNs' origin validation [C]. In 2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS), pp. 1-10. IEEE, 2018.

[272] Guo, Run, Weizhong Li, Baojun Liu, Shuang Hao, Jia Zhang, Haixin Duan, Kaiwen Shen, Jianjun Chen, and Ying Liu. Cdn judo: Breaking the cdn dos protection with itself[C]. NDSS, 2020.

[273] Cao, Jiahao, Renjie Xie, Kun Sun, Qi Li, Guofei Gu, and Mingwei Xu. When Match Fields Do Not Need to Match: Buffered Packet Hijacking in SDN[C]. In Proc. of the Network and Distributed System Security Symposium (NDSS' 20). 2020.

[274] Cao, Jiahao, Qi Li, Renjie Xie, Kun Sun, Guofei Gu, Mingwei Xu, and Yuan Yang. The crosspath attack: Disrupting the {SDN} control channel via shared links[C]. In 28th {USENIX} Security Symposium ({USENIX} Security 19), pp. 19-36. 2019.

[275] Weizhong Li, Kaiwen Shen, Run Guo, Baojun Liu, Jia Zhang, Haixin Duan, Shuang Hao, Xiarun Chen, Yao Wang. CDN Backfired: Amplification Attacks Based on HTTP Range Requests[C]. DSN 2020 (best paper nominee).

[276] Shang, Gao, Peng Zhe, Xiao Bin, Hu Aiqun, and Ren Kui. FloodDefender: Protecting data and control plane resources under SDN-aimed DoS attacks[C]. In IEEE INFOCOM 2017-IEEE Conference on Computer Communications, pp. 1-9. IEEE, 2017.

[277] 李彪. SQL 注入漏洞检测系统的设计与实现[D]. MS thesis. 北京工业大学, 2019.

[278] 潘瑾琨. 跨站脚本漏洞检测技术研究[D]. Diss. 国防科技大学, 2017.

[279] 吕成成. 面向 WEB 应用程序的输入功能测试与 XSS 漏洞检测[D]. MS thesis. 中国科学技术大学, 2019.

[280] Tian, Yifan, et al. CNN-WebShell: malicious web shell detection with convolutional neural network[C]. Proceedings of the 2017 VI International Conference on Network, Communication and Computing. 2017.

[281] Wang, Jiabao, Zhenji Zhou, and Jun Chen. Evaluating CNN and LSTM for web attack detection[C]. Proceedings of the 2018 10th International Conference on Machine Learning and Computing. 2018.

[282] Fang, Yong, et al. Detecting WebShell based on random forest with fasttext[C]. Proceedings of the 2018 International Conference on Computing and Artificial Intelligence. 2018.

[283] Cui, Handong, et al. WebShell detection based on random forest-gradient boosting decision tree algorithm[C]. 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC).

IEEE, 2018.
[284] Feng Q, Zhou R, Zhao Y, et al. Learning binary representation for automatic patch detection[C]. 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2019: 1-6.
[285] Zhang H, Qian Z. Precise and accurate patch presence test for binaries[C]. 27th {USENIX} Security Symposium ({USENIX} Security 18). 2018: 887-902.

作者简介

杨　珉　复旦大学计算机学院科研副院长，教授、博导，研究方向为智能系统安全。

张　磊　博士，复旦大学网络空间国际治理研究基地助理研究员，研究方向为系统安全、漏洞挖掘、区块链安全、人工智能及其安全等。

荆继武　中国科学院大学教授，研究方向主要为数据保护。

刘欣然　中科院计算技术研究所正高级工程师，博导，研究方向为网络安全、分布式系统等。

胡传平　公安部第三研究所研究员，博导，研究方向为计算机视觉和网络安全。