2023

SecXOps
安全智能分析技术白皮书

● 智能基座，开启安全分析新时代

NSFOCUS

关于绿盟科技

绿盟科技集团股份有限公司（以下简称绿盟科技），成立于 2000 年 4 月，总部位于北京。公司于 2014 年 1 月 29 日在深圳证券交易所创业板上市，证券代码：300369。绿盟科技在国内设有 50 余个分支机构，为政府、金融、运营商、能源、交通、科教文卫等行业用户与各类型企业用户，提供全线网络安全产品、全方位安全解决方案和体系化安全运营服务。公司在美国硅谷、日本东京、英国伦敦、新加坡及巴西圣保罗设立海外子公司和办事处，深入开展全球业务，打造全球网络安全行业的中国品牌。

版权声明

NSFOCUS

CONTENTS

执行摘要

随着网络空间的攻击面的延伸和拓展，网络空间攻防双方信息不对称的现象愈发明显。伴随着攻防对抗态势的升级，自动化技术、智能化技术与安全分析技术融合的安全智能分析技术已成为网络安全技术发展的必然趋势之一。

SecXOps 即 XOps for Security，以 XOps 与安全场景的融合为基础，由安全数据资产高质可信、安全模型全生命周期管理、安全模型高精度定制、安全模型自动化运营、AI 工程化持续保障五大核心技术能力组成，在保证安全性的同时，减少技术和流程的重复，实现网络安全分析自动化、智能化进阶，是未来应对网络空间高级、持续、复杂威胁与风险不可或缺的关键技术之一。

绿盟科技推出 SecXOps 安全智能分析技术白皮书《智能基座，开启安全分析新时代》，旨在对 SecXOps 概念内涵、技术优势、核心能力、关键技术和应用实践进行全面地总结与介绍，期望为读者带来全新的技术思考，助力网络安全智能分析实现自动化、智能化进阶。本技术白皮书的主要观点如下：

智能分析是网络安全分析的必然趋势：

随着网络空间攻防对抗态势不断升级演化，数字化时代的特征倒逼网络安全分析突破依赖安全专家的传统“人工”阶段，安全智能分析已成为网络安全风险治理与防控的必备条件之一。

网络安全实战场景是安全分析应用的“试金石”：

安全分析模型从实验室研究走进网络安全实战化的场景，走进常态化的日常安全监测中，面临着诸多挑战，只有以实战检测的方式来验证安全分析模型的价值，才能有效地促进安全分析能力的提升。

SecXOps 成为提升安全分析自动化和智能化水平的关键：

SecXOps 将 XOps 实践应用到网络安全分析中，以支撑安全数据治理，安全模型训练、管理和监控，为网络安全的数据分析人员、ML 工程团队、应用开发团队以及安全运营团队的协作搭建安全、兼容和经济高效的平台，从而实现基于 AI 安全模型的持续交付，完成网络安全分析技术与大数据和人工智能技术的深度协同融合，全面提升安全分析的自动化和智能化水平。

促进 SecXOps 生态建设，共同推动网络安全分析智能化：

SecXOps 技术的研究和攻防场景应用实践仍然具备较大的上升空间，在理论方法、标准制定和模型运营等方面需要进一步的研究与探索，需要技术生态的构建，营造网络安全分析智能化大时代技术氛围。

01

安全分析的发展背景和趋势

近年来，互联网、大数据和人工智能等技术都得到了飞速的发展，网络攻击的方法也越来越复杂，过去广泛、漫无目的的攻击威胁，在数年内迅速地转化为有目标、有组织、长期潜伏的多阶段组合式高级可持续威胁（Advanced Persistent Threat，APT）攻击。APT 攻击有着复杂度高、对抗性强、特征隐蔽等特点，通常由有国家背景的相关攻击组织发起，实施窃取国家机密、重要企业的有价值商业信息、破坏网络基础设施等活动，具有强烈的政治和经济目的，严重影响网络空间稳定运行，造成国民经济损失，威胁国家安全。随着 APT 攻击等高隐蔽未知威胁的出现和演进，传统安全分析技术难以满足 APT 攻击检测的要求，亟需融合多手段的检测技术来应对种类日益多样化的安全威胁攻击。在安全威胁具有更强的杀伤力与隐蔽性的形势下，结合大数据和人工智能技术的安全智能分析成为新一代安全能力的关键，是网络空间安全的重要发展方向之一。

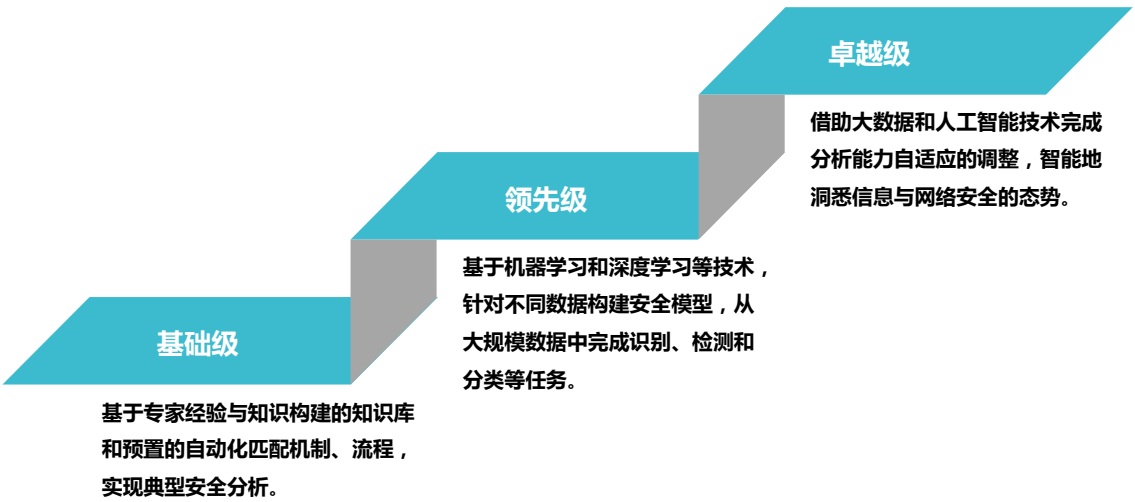

图 1　网络安全分析发展阶段

由于 APT 攻击等网络威胁利用大数据分析、自动化工具等先进技术来提升恶意攻击的效率和隐蔽性，倒逼网络安全分析突破依赖安全专家的传统“人工”阶段，进入安全智能分析阶段。安全智能分析运用人工智能技术从安全大数据中进行威胁检测分析，直接或间接地提高安全分析效率，在实际攻防实战中充当智能化助手的角色，帮安全分析员更加快速地定位威胁攻击，提升安全分析的自动化、智能化水平。回顾网络安全分析发展历程，可以将安全分析技术发展大致划分为三个阶段，包括基础级、领先级、卓越级，如图 1 所示，以下分别进行简要介绍。

- **基础级**

  基础级的安全分析技术基于专家经验与知识构建的知识库和预置的自动化匹配机制与流程，实现典型安全分析，利用安全专家的知识和解决问题的方法来分析安全数据。该阶段面向不同的应用场景，需要专家编写指定的检测规则，以列表结构、树结构、图结构简单组织的规则逻辑结构，分析场景下的攻击行为。然而，随着攻防技术的快速迭代和升级，攻防场景与流程的细化，此类分析方法逻辑的完备性在大数据场景下迎来关键挑战，针对攻击的误报率、漏报率和整体准确性性能衰减很快，难以有效自适应演化，过度依赖专家资源，可维护性低，能够支撑分析的场景愈发受限。

- **领先级**

  领先级的安全分析技术面向不同的网络安全数据，包括安全环境数据（资产、资产脆弱性、文件信息、用户信息），行为数据（网络侧检测告警、终端侧检测告警、文件分析日志、应用日志、蜜罐日志、沙箱日志），情报数据（各类外部威胁情报）以及安全知识数据（ATT&CK）等，基于机器学习和深度学习等技术，针对不同数据构建安全模型，从大规模数据中完成识别、检测和分类等任务。虽然在诸多网络安全分析的场景下基于机器学习和深度学习等安全模型取得了重要的突破，但是随着数据的变动，传统的构建安全模型分析的方法无法有效地完成模型运营，导致在面对 APT 等高级复杂攻击技战术分析时，安全模型分析的结果仍需要深度的专家参与的研判与关联分析来判定安全分析的业务价值，限制了安全分析自动化、智能化水平的提升。

- **卓越级**

  卓越级安全分析借助大数据和人工智能技术完成分析能力自适应的调整，能够更加智能地洞悉信息与网络安全的态势，更加主动、弹性地去分析新型复杂的威胁和未知多变的风险。自适应安全分析采用安全分析工程化的最佳实践提升安全分析的自动化和智能化水平，在确保可靠性、可用性和可重复性的前提下，减少技术和流程的重复，实现安全分析能力的持续交付，发挥安全分析技术在安全领域的巨大潜在价值，推进网络安全智能不断走向成熟，是分析网络空间高级、持续、复杂威胁与风险不可或缺的技术基础。

02

安全智能分析的挑战

随着各个国家的重视和布局，大数据技术和人工智能技术发展迅速，相关自动化与智能化的识别和处理能力、数据分析能力逐渐与网络安全技术进行了深度协同，对网络安全的技术、方法、应用产生了重要影响，促进了网络安全技术的变革性的进步[1]。可以预见的是安全数据采集和安全智能数据分析技术的成熟将会大幅提升网络安全威胁检测、网络安全风险评估等关键安全防御环节的效率，大幅减少对网络安全专家的依赖，有效地降低企业、组织乃至国家级关键信息基础设施、数据资产的整体安全风险[2]。因此，安全智能分析能力的提升已经成为安全能力落地、发挥网络安全防御有效性和对抗 APT 等高级威胁最直接、最关键的环节之一。面对日趋白热化、持续化的网络攻防对抗环境，安全智能分析也在多个方面面临着诸多挑战。

2.1 数据治理

企业数字化转型浪潮的来临，多源异构数据的爆发式增长，使数据治理得到了企业的普遍关注和重视。大规模数据蕴藏的巨大潜在价值吸引着攻击者对集中存储的数据进行窃取、对海量数据的管理是企业亟待解决的一项艰巨任务。

数据治理旨在解决数据在生产、管理和使用中面临的各种问题。从数据源汇入开始，在数据清洗、数据存储、数据分析、数据服务等数据生命周期涉及的所有环节中，数据治理对企业内部的数据集进行规范和定义，并结合企业自身数据现状，为各个环节提供持续的治理服务[3]。随着企业数字化程度的提升，数据治理的需求和复杂度也会增加。由于安全领域自身的特点，数据治理面临着以下挑战：

- **数据采集：**网络安全领域数据壁垒问题严重，安全数据作为敏感数据，往往分布在各个数据持有者手中，数据采集缺乏可访问性和采集渠道，导致采集的安全数据集不完整、不可靠，不足以代表安全问题的真实数据分布，数据多样性的匮乏和完整性的不足会影响到后续的数据挖掘、数据分析等数据操作的质量。
- **数据标签：**数据标签化旨在为安全数据贴上精准的标签，安全分析任务所需的真实标签不准确、不稳定或错误，将影响后续安全分析模型的整体性能，反之数据标签越丰富，越准确，后续的安全分析依据就越多，决策也越准确。安全数据标签化深度依赖企业在研究中的积累和专家知识，海量的攻击样本、复杂的攻击类型、多源异构的数据都为数据标签化带来难以回避的挑战。
- **数据规模：**安全分析需要大规模、多维度的安全数据作为基础，但处理海量数据也给安全分析带来多方面挑战，如依赖爆炸、海量告警场景下的告警疲劳等问题，以及海

量数据在数据采集、传输和存储阶段为系统带来的压力，都会降低整个安全数据治理流程的效率。

- **动态治理：**安全数据包含动态产生的终端侧、网络侧、沙箱侧、蜜罐侧的告警日志，以及威胁情报、漏洞、知识库、IT 资产等数据资产，对这些安全数据的动态治理是实现可靠数据分析的关键。为了在复杂多变的网络空间环境中制定出动态自适应的分析策略，安全数据治理对动态治理提出更高的要求。

因此，安全数据的治理需要对安全数据进行持续采集、标注、存储以及动态治理，基于大数据的治理技术确保安全数据的质量，在提高数据质量的同时减少数据分析的周期。

2.2 模型开发

近年来，随着硬件计算能力的日益强大与数据量的井喷式增长，深度学习的应用也是与日俱增。尽管经过了数十年的发展，深度学习的实用性和普适性已经有了很大的提升，例如，使用计算机视觉领域的卷积神经网络也可以对文本进行处理，并且得到比较好的效果。然而在很多场景下，不同领域的模型针对某一特定场景的任务在准确率、查准率、查全率和时间复杂度等很多指标上都有明显的鸿沟，无法将其他领域的模型直接应用到一个新的领域。

数字时代的背景下，数据和智能驱动的安全对抗，技术平台的自动化、智能化水平，攻击面的拓宽，导致网络安全分析所需采集的数据种类多、数据规模异常庞大。因此，从防御角度来看，安全分析模型需要快速检测分析多源异构安全数据，辅助安全团队进行分析。目前，安全模型构建面临着以下挑战：

- **特征提取困难：**

  人工提取面向网络安全领域的特征较为复杂，现阶段机器无法自动化高效地识别有效的特征，存在一些与安全问题无关但是能够在分类任务指标上表现良好的特征也会导致安全模型去拟合这些特征而不是去解决安全问题。例如，在网络入侵检测场景中，数据集中绝大多数攻击流量来自于一个特定网段，导致模型不学习攻击模式，而是学习特定 IP 段来检测攻击，导致了 IP 段与网络入侵的虚假相关 [4]。

- **泛化能力差：**

  在网络安全领域，不同的安全场景之间的差距较大，相关模型在学术界并没有像 CV 或 NLP 领域那样形成一个稳定的思路或流派。例如，针对加密流量的分类，从

CNN、RNN 再到一些集成算法，研究者们进行了很多尝试，仍然没有算法能够脱颖而出。

- **可解释性低：**

  基于深度学习等复杂不可解释的黑盒模型，以及低交互甚至无交互的人机交互流程设计，是各种模型在安全领域应用的重要阻碍。例如，在 web 攻击识别方面，需要分析的数据是一段形如程序命令的文本，并不同于自然语言有明显的语义上的连贯性，导致 NLP 方法在面对预测结果时也无法自圆其说，模型的分析结果没有较强的可解释性，并且一旦如果模型不可解释，就意味着该模型本身是不可知、不安全的，当受到投毒攻击、对抗样本攻击时很难进行快速处理。

因此，只有确保信息可靠性、明晰模型输入输出的因果关系，模型的预测结果才能令人信服，才能投入使用 [5]。面对日新月异的攻击手段时，安全分析模型无法对诸如零日漏洞、未知流量入侵等未知攻击进行准确检测，动态变化的网络数据进一步加大了安全分析模型增量更新的难度。

2.3 模型交付

对于一个网络安全领域的任务，机器学习或深度学习算法的选择是一件耗时很长的任务，模型选定后的调优过程也需要大量的时间成本。目前大多学术论文中的研究不会涉及算法在真实场景交付的问题。由于算法模型的分类结果依赖收集到和标定好的数据集合，在实际应用场景下，数据分布的变化、数据复杂度的提升等因素可能会导致大部分实验室中表现良好的模型面临失效的困境。真实场景数据大部分情况下无法直接获取用于模型训练，因此在模型训练阶段不能和本地环境中的数据做大量的融合适配，无法基于本地环境的特性去标定整个模型和检测点的指标，通常训练阶段会采用合成数据来扩展数据集或者使用迁移学习的方法来提升模型的普适性等，但是在真实的环境中还是会存在各种问题。例如，人工智能算法在加密流量领域的应用，当算法模型在实验室封闭数据集上得到高检测率和低误报率的效果后，在真实场景下仍然会因为环境流量的复杂导致误报率较高，并且，在此场景下对于大部分现场安服和分析人员来说，他们并不具备对加密流量的验证能力，即验证一条加密流量所需要的成本非常高，导致用户使用相关安全分析模型的意愿降低。

因此，网络安全领域一方面需要提高安全分析模型的开发效率和上线速率，另一方面也需要保证安全分析模型离开实验室环境到真实环境中后依旧是有效的。安全分析模型在上线之前应该部署于多个真实环境的实验中以发现实验室环境和单一真实环境中发现不了的问

题。此外，在安全分析模型交付期间，需要考虑真实场景中数据的时间特性、空间特性、运行时间和存储限制，以及如何与其他流程完成工程化的结合以实现业务目标。例如，在加密流量分析模型的实际应用中，需要结合加密流量实际场景研判的角度，将安全分析模型作为整个加密流量判别的一个环节，通过整个流程中其他模型关联的联合验证，以宏观的角度完成加密流量的研判。

2.4 模型运营

如今，企业组织的数字化转型带来了海量的数据，这些数据倾向于云存储而不是物理存储，这一趋势使得安全分析模型需要处理的场景也在变化。例如，为了保护在家和移动设备上的办公安全，需要新的身份管理系统去保障多源的数据和代码的安全性，机器学习等技术被引入来管理和分析安全数据，以自动化地识别威胁降低风险。随着安全分析模型在网络安全领域的应用越来越多，网络攻击手段越来越丰富，安全分析模型检测威胁难度不断增大，导致很多安全分析模型无法投入真实环境。安全分析模型一旦投入脱离单一的实验室环境，就会面临着快速迭代的真实攻击场景、持续化的网络攻防对抗环境，从而导致模型逐渐失效。安全产品的安全分析模型检测效果随着时间推移而效果下降的问题是不可一次性解决的，需要持续进行运营。以 web 攻击检测为例，对于传统的基于规则检测的模型，安全研究员对于运营过程中采集到的数据可以直接进行分析，同时对规则进行修改，更新模型。但当告警数量激增的情况下，这种传统的模型运营模式难免显得力不从心。相比之下，对于机器学习和深度学习算法，运营流程只会更加复杂。运营人员将漏报误报数据收集好后需要与算法工程师进行对接，针对不同的场景、不同的数据、不同的算法需要采用不同的数据处理方式，模型更新完成后又需要与产品开发人员对接，导致运营、研究、开发多个部门之间耦合程度过高，造成人力和时间的浪费。

因此，网络安全领域安全分析模型的应用需要打破多个部门之间的协作壁垒，建立一个良好的团队之间协作的通道，通过有效的安全分析模型的运营，在生产环境中基于模型的有效性和效率等指标 (效率指标可包括目前硬件资源利用率和延迟等) 持续地监控安全分析模型，以支撑模型可审计、可追溯，完成安全分析模型有效的运营。

2.5 AI 工程化

网络安全分析模型应用到真实场景需要考虑诸多因素，包括真实场景中算法效果的局限性，上下游数据是否具备可用性，算力是否能够支撑安全分析算法的运行实现，以及性能的消耗等，因此，需要在基于机器学习和深度学习模型在网络安全应用场景中面向数据、

算法、模型等要素建立完备的工具体系、标准化研发和管理体系。为了实现网络安全领域基于 AI 的模型的高效交付，需要利用跨数据、机器学习、人工智能和应用程序开发管道的多种最佳实践，实现 AI 工程化，支持将 AI 交付模块化、编排、发现和自治作为业务框架的组成部分。

目前，在网络安全分析的过程中，当遇到跨多个环境部署应用 AI 分析模型时，传统的虚拟机技术无法有效应对所需要的环境资源配置。在多个系统环境和框架中无缝移动和运行，并保持结果的一致性，才可以使 AI 应用的迁移更加简洁。此外，AI 工程需要用到 GPU，CPU 等诸多服务器资源，只有高效且灵活的资源动态调度方案，才可以保证在不同主机、集群之间，实现有效地调度，维护 AI 系统的稳定性，以适用于网络安全海量数据、众多业务中快速响应安全分析的请求，实现有效的网络安全威胁分析。

因此，网络安全智能分析的实现中通过环境容器化、环境迁移、模型资源动态调度管理等技术，实现 AI 工程化，以在网络安全分析中 AI 系统涉及的所有阶段提供持续集成和持续交付，支持和保障系统的可重用性、结果可再现性、代码可回滚性、数据可沿袭性和环境安全。

03

SecXOps
技术体系

本章将介绍 SecXOps 技术的核心内涵、技术优势与技术框架。

3.1 概念内涵

运维（Ops）发展至今在企业中的重要性越来越高，随着各行各业数字化规模越来越大，组件监控粒度越来越细，不断有新技术和组件被引入运维体系。运维体系不断向着多元化方向发展，XOps 即其他技术与 Ops 的融合，随着不同的 Ops 发展，XOps 已成为定义 DevOps、DataOps、MLOps、ModelOps、Platform Ops for AI 等组合的总称。Gartner 指出 XOps 的目标是使用 DevOps 的最佳实践实现效率和规模经济，在确保可靠性、可用性和可重复性的前提下，减少技术和流程的重复，实现进阶自动化 [6]。总的来说，XOps 技术促进企业组织通过数据和分析的运营技术赋能业务，推动提升业务价值。

SecXOps 即 XOps for security，以 XOps 与安全场景的融合为基础，由安全数据资产高质可信、安全模型全生命周期管理、安全模型高精度定制、安全模型自动化运营、AI 工程化持续保障五大核心技术能力组成，在保证安全性的同时，减少技术和流程的重复，实现网络安全分析自动化、智能化进阶，是未来应对网络空间高级、持续、复杂威胁与风险不可或缺的关键技术之一。

3.2 技术优势

SecXOps 将 XOps 实践扩展到网络安全领域，从安全数据治理、ML 模型管理、AI 模型管理和底层基础设施建设等各个阶段建立强大的 DevOps 实践，以支撑安全数据治理与安全模型训练、管理和监控，发挥 Ops 技术在安全领域的巨大潜在价值，为网络安全的数据分析人员、ML 工程团队、应用开发团队和安全运营团队的协作搭建安全、兼容和经济高效的平台，从而实现基于 AI 的安全系统的持续交付。其技术优势如下图所示：

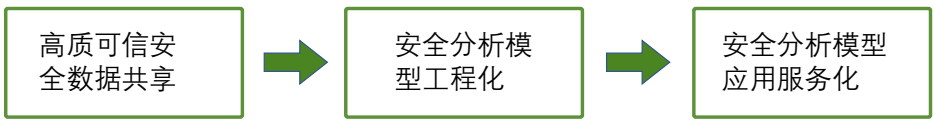

图 2　SecXOps 技术优势

- **高质可信安全数据共享**

  SecXOps 在安全数据分析过程中，通过数据治理的方案，管理不规则的数据管道，完善数据采集、数据工程、数据分析的流程，实现安全数据资产的统一化管理，包含网络流量数据、恶意样本行为数据、威胁情报数据等，协助安全数据研究，打破安全

数据壁垒难题，优化数据处理的各个环节的协作，解决数据错误和数据缺失的问题，提升安全数据资产的质量和数据的处理效率，实现自动化的安全数据交付，达到安全数据迭代交付，不断提升安全数据的价值。

- **安全分析模型工程化**

  SecXOps 通过自动化、构建模块的重复使用和业务流程的集成，可持续地操作网络安全领域的ML模型和AI模型等模块，实现网络安全应用相关的AI分析等模块的扩展。基于 XOps 技术中高性能模型敏捷交付方法，解决安全数据依赖、自动化训练、优化复杂配置的问题，协助网络安全领域模型的研究，完成安全分析模型开发、交付的自动化，做到安全模型全生命周期管理，简化端到端 ML 工作流程，加快安全分析模型从研究到工程部署的速度。同时，利用集成系统的质量指标来追踪 ML 和 AI 模型等模块性能指标，支撑 ML 和 AI 模型等模块的再训练，实现大量网络安全领域模型及时的反馈闭环，有效地保障模型的再现性、可追溯性、完整性和可整合性。

- **安全分析模型应用服务化**

  SecXOps 为网络安全领域构建自适应的系统，通过可视化、可编排的管理平台，构建安全模型资产管理和多元模型开放服务体系，实现安全模型一体化运营。在安全相关应用从开发到落地的整个生命周期中，观测和预测系统中安全模型的变化，确保安全模型的应用效果，克服安全模型高精度定制的难题，提高模型应对威胁和攻击的能力，为意外的网络安全环境变化做好准备，根据不同的运维目标、场景，结合环境和专家的反馈，完成 AI 安全分析模型的应用服务化，实现规模化的生产与利用，从而减少网络安全领域生产过程维护模型所需要的成本。

SecXOps 从数据采集、模型开发、模型部署，模型管理等方面为网络安全分析提供必要的透明度和可见性，以支撑对网络安全分析模型的有效性进行判别，在模型投入生产前以全局的视角对其结果进行审查，实现从多个维度有效衡量和跟踪网络安全领域 ML 和 AI 模型在其生命周期中业务影响和价值，避免将网络安全模型快速扩展到安全业务落地阶段后对企业造成的损失。未来 SecXOps 技术将成为在安全分析流程中形成机器效率与安全人员经验融合闭环、辅助 AI 模型在安全领域逐渐落地的不可或缺的基础技术。

3.3 核心能力

为了加快安全分析能力更全面、更深入的自动化，SecXOps 的目标在于创建一个集成的用于 Security 的 XOps 实践，提升安全分析的场景覆盖率和运营效率。SecXOps 技术并不

是 Ops 技术在安全领域的简单加和，SecXOps 技术的实现一方面取决于不同 Ops 技术自身的发展水平，另一方面，更取决于 Ops 技术与安全分析场景在核心目标、体系架构、功能需求、数据模型的各个方面适配。不同于安全运维实践中 Ops 技术与安全领域的单点结合，SecXOps 是面向安全分析的各个环节，进行系统、深入的多维融合技术方案，可以适应安全分析不同阶段、不同任务场景的应用需求，实现安全数据价值的最大化，这对传统 Ops 技术的鲁棒性、可信性、安全性提出了全新的要求。本文将 SecXOps 核心技术能力划分为五部分，包括安全数据资产高质可信、安全模型全生命周期管理、安全模型高精度定制、安全模型自动化运营、AI 工程化持续保障，如下图所示。以下首先分别对各个方面进行简要介绍，后文将进一步展开详细说明。

图 3　SecXOps 核心技术能力拆解

3.3.1　安全数据资产高质可信

随着全球数字经济的蓬勃发展，各种复杂业务越来越多，高级持续性威胁的出现和演进导致网络安全所要分析的内容急剧增加。传统的安全运营仍然深度依赖安全专家去研判，进行调查取证及分析，涉及的数据包括终端侧、网络侧、沙箱侧、蜜罐侧的告警，系统日志以及威胁情报、知识库、扫描的漏洞等。从数据层面来看，安全数据通常来源分散、语义多样、格式异构，因此，需要针对不同的数据源进行采集、预处理、打标签等工作。以往数据分析人员进行数据处理后仅将数据处理成适用于自己场景下的模型训练的输入，这种低耦合、低交互的方式导致数据的分析通常无法复用。为了应对数据本地化理解、脆弱性和新的数据用

例扩展速度慢等挑战，SecXOps 对安全数据进行有效地管理，包括数据提取、集成、转换和分析。面向网络环境数据、威胁行为数据、威胁情报数据、安全知识库等，引入机器学习和知识图谱等技术，根据攻击行为的特征与多源异构数据的特点对数据进行定义和分类，协调不同工具构建数据处理管道，基于深度学习技术进行实体消歧、实体统一等操作，提取安全数据的描述信息，将多源异构数据转换成可以连接的数据，以灵活的图数据结构将数据管道中不同阶段的数据提供给各个领域的安全专家，使其针对多源异构数据进行高效地协同组织与管理，促进专家知识储备不同的安全人员之间的合作，使得安全数据可以在动态环境中更快更智能地被利用，完成自动化的安全数据交付的设计和管理。

3.3.2 安全模型全生命周期管理

目前各种模型算法在网络安全领域中大多较难落地。因为即使在各种学术论文中，模型的检测率达到了 99.9%，一旦当安全数据为海量级别的时候，会造成大量的误报或者漏报。安全领域的建模仍需要业务人员、数据分析工程师、AI 建模工程师，安全运营研判人员投入大量的精力，在场景中进行持续的反馈、沟通、迭代和优化。各种算法模型在安全领域的落地尚处于初步阶段，大多方法存在泛化能力差、交付效率低，且在真实场景中无法解决实战的问题。随着安全对抗的升级、攻击手段的变化，以及业务需求的变动，当模型在真实网络安全环境中出现性能衰减的时候，SecXOps 基于工程化的实践，自适应调整模型参数来应对复杂多变的网络环境，以取得最佳检测效果，只有可解释、增量更新的安全分析模型，在网络安全领域才是可评估、可实战、可运营的。SecXOps 技术可靠和高效地在生产中完成机器学习模型的发布、激活、监控、管理和更新，实现安全模型全生命周期管理，在机器学习模型整个生命周期中融合模型的开发和管理阶段所需不同专家的能力，打破沟通屏障，确保各种机器学习模型在安全领域的落地。

3.3.3 安全模型高精度定制

就安全场景而言，流量分析、用户实体行为分析、样本分析、威胁关联、自动化响应等安全能力逐渐集成机器学习算法，但是在安全领域每一次针对安全事件的漏判错判都可能会造成无法挽回的损失。为了使 AI 技术在安全运营领域逐渐落地，从而减少安全运营人员的工作量，需要针对安全分析模型进行有效的 AI 模型定制。安全分析模型需要考虑模型的安全性，在真实环境上线前需要对安全分析模型进行攻击，以检测和确保安全分析模型自身的安全性。因此，安全分析需要 AI 以透明、可解释的方式输出其判断和决策，以 AI 的性能弥补人类在数据处理上的低效性、AI 的鲁棒性来适应不同的使用环境，并且保障 AI 自身的安全性。

SecXOps 设计安全分析模型的定制流程，扩展人工智能在网络安全分析中的应用，协助部署，治理和监控生产环境中的 AI 模型。通过为安全业务、开发和运营团队提供一定程度的透明度，帮助企业最大化和扩展 AI 相关计划，在网络安全检测、溯源、响应等各环节自动化任务中为业务领域专家提供可解释的结果，结合 AI 模型进行推理，丰富安全知识和网络安全领域的应用，完成威胁建模、风险分析、攻击推理，加速安全进入认知智能。

3.3.4 安全模型自动化运营

目前，由于安全分析模型从构建、部署到运营的流程涉及数据分析人员、机器学习模型构建人员以及其他 IT 部门之间的相互配合，复杂的安全分析模型运营和治理流程导致安全分析模型应用举步维艰。在其他领域的动态环境中表现出色的工程化实践能够针对不同的业务问题采用不同类型的模型，确保随着时间的推移模型预测会保持准确，并且遵守所有法规和风险要求，同时为业务领域专家提供了可解释的结果，以及在不完全依赖专家的情况下升级和降级用于推理的 AI 模型的能力。SecXOps 技术构建安全模型自动化运营体系，突破现有的管理流程，实现有效的安全分析模型规模化治理，帮助企业在实际生产环境中有效地部署和应用安全分析模型，确保在企业不断变化的环境中，安全分析模型仍然可以有效地识别和检测威胁风险，满足企业的业务目标。

3.3.5 AI 工程化持续保障

人工智能平台运营 [7] 是一种编排和扩展 AI 的方法，用于构建和交付基于 AI 的系统，涉及数据、ML、AI 和应用程序开发管道的多种最佳实践，实现分析、ML 和 AI 模型部署的管理，为基于 AI 的系统创建高效的交付模型。SecXOps 通过人工智能运营平台构建基于人工智能的安全系统，利用模块化和业务编排的底层平台来不断扩展人工智能在安全中的应用，管理端到端的 AI 安全平台，从数据到模型的管道建立一致性，在构建基于人工智能安全系统的各个阶段提供持续集成和持续交付的能力，为业务部门提供自主权，加快人工智能解决方案在网络安全领域的采用和交付。

04

SecXOps
关键技术

4.1 安全数据资产统一管理

DataOps，即 Data 和 Operations 的集成，于 2014 年首次提出。Gartner 将 DataOps 定义为“一种协作性的数据管理实践，专注于改进组织内数据管道的通信、集成和自动化”[7]。

DataOps 是一种面向流程的自动化方法，适用于从数据采集到生成数据分析报告的整个数据生命周期，通过创建对数据、数据模型和相关组件的可预测交付和变更管理，实现更快的价值交付。在整个数据生命周期中，DataOps 提供一个所有成员无缝协作的环境，协助数据团队将数据孤岛转变为敏捷、高速、自动化的数据供应链，持续改善和优化整个数据管道，最大限度发挥数据价值。

DataOps 在产品开发的各个方面保障数据完整性，旨在提高数据分析质量和缩短数据分析周期，提高企业使用数据的效率，降低使用数据的门槛和成本，让数据为企业带来更多价值。

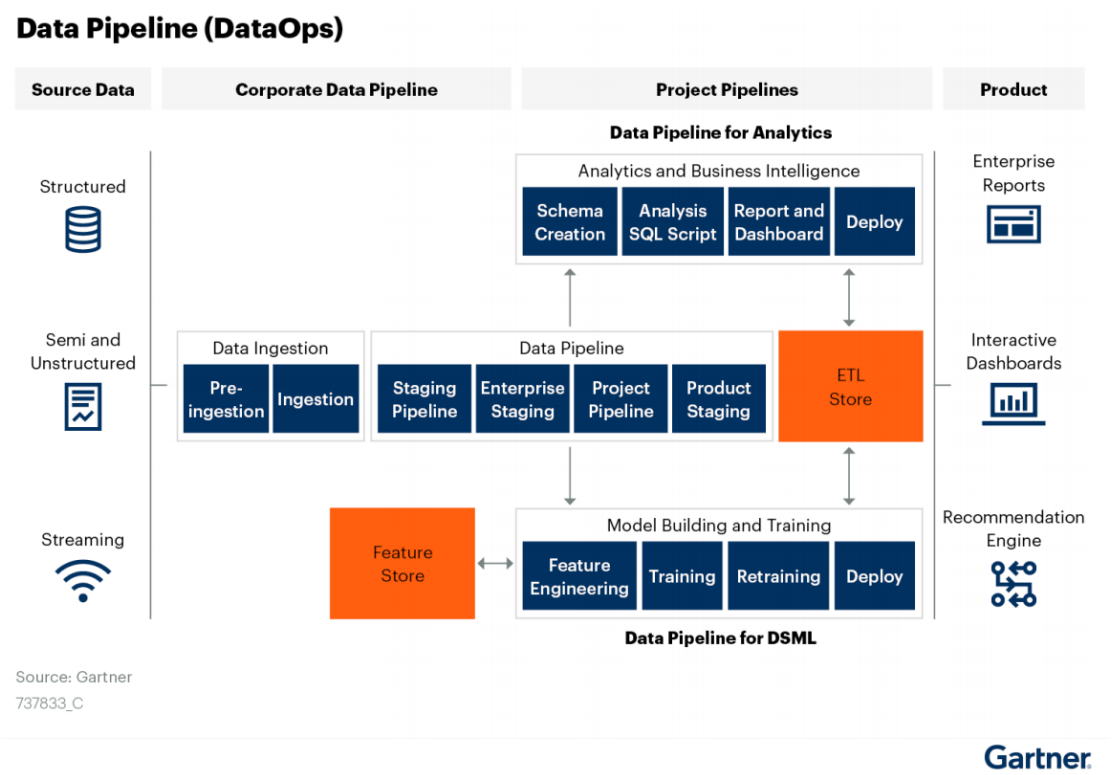

图 4 DataOps[7]

4.1.1 数据存储

4.1.1.1 定义内涵

数据存储是指将数据以某种格式记录在计算机内部或外部存储介质上。在本节中主要介绍 SecXOps 中的数据集存储方式。

4.1.1.2 技术背景

传统的数据存储是一个手动的、管理进行驱动的过程，需要专家知识管理数据存储的整个周期，不仅成本昂贵、耗费大量资源，也易受到存储资源复杂性的困扰和限制。随着数据向云上部署进程的推进，传统管理方法使得存储过程和 IT 运营更为困难。DataOps 的问世缓解了上述困境。

DataOps 的理念建立在 DevOps 之上，旨在实现敏捷数据存储、分析和管理。眼下很多公司和企业都没有完善的数据管理过程，缺乏版本控制、持续集成等环节，数据的传递流程需要人去沟通，而 DataOps 则是建立了一个完整的数据管道，极大地简化了数据的存储管理流程。

一个典型的数据管道包括数据提取、集成、转换和分析阶段。在数据存储阶段，敏捷数据流程往往从单个数据子集和子集的增量价值交付开始，需要软件开发人员、IT 运维人员和数据团队之间进行协作，并尽可能将这个流程自动化。

DataOps 以数据作为基础和支撑。然而，对安全领域而言，安全数据资产的合法获取十分困难。安全数据的来源通常是黑客论坛、博客、社交网络、应用程序论坛等，但这些渠道获取的数据存在以下问题：一是数据量往往不足以支撑项目的运行，二是数据质量无法保障，三是数据集中是否存在投毒数据、是否携带后门触发器等风险尚未可知。开发者也可以通过爬虫程序获取安全数据，但许多企业已经具备反爬虫意识，在网页代码中嵌入了反爬虫策略，同时也为数据收集者带来了法律风险。总之，安全数据的收集在获取渠道、获取方式上都存在着较高难度。

4.1.1.3 思路方案

在众多开源的自动化机器学习平台中，数据的存储方式各有差异。例如 H2O，它能够从 HDFS、S3、NoSQL、SQL 中读取数据或写入数据，能从本地和分布式文件系统接收 CSV 格式的数据，支持写代码对数据进行处理。Databricks 则与 Spark 和 Apache 的数据湖相结合，为批处理或流式提取、转换和加载提供了一个简单的接口。

而在 SecXOps 平台上，我们允许用户创建多个数据集，同时一个数据集可以拥有多个版本，在每一个版本里可以存放大量的数据。一个数据集版本内的全部数据存放在一个 Kubernetes 的 PVC 当中。

PVC 是 Kubernetes 中的一个概念，全称是持久卷申领，表达了用户对存储的需求。PV 又叫持久卷，是集群中的一块存储，可以由集群管理员预先制备，也可以通过使用存储类（storage class）来动态制备。PVC 的申领会耗费 PV 资源，SecXOps 平台上部署了 NFS 的存储服务作为一个存储类，允许 Kubernetes 在创建 PVC 时自动创建一个 PV。

当用户想要访问数据集时，平台会根据用户的选择，创建一个 ReplicaSet。ReplicaSet 是 Kubernetes 中的一种 Pod 控制器，主要作用是能够保证一定数量的 Pod 能够正常运行。ReplicaSet 会持续监听这些 Pod 的运行状态，一旦 Pod 发生故障，就会重启或重建。由于数据集的查看和在线编辑对性能需求并不会很高，因此 ReplicaSet 下创建一个 Pod 即可。Pod 内包含一个用户容器，将数据文件所在的 PVC 挂载到该容器上，供用户上传、下载或在线编辑数据文件。SecXOps 支持用户选择数据集版本的打开方式，平台上集成了 jupyterlab，允许用户能够编写脚本对数据集进行批量化的在线修改，同时还集成了 filebrowser，允许用户能够快速地批量上传和下载文件。

SecXOps 不仅支持用户从本地上传数据文件，从用户个人或企业内部配置的数据管理服务（如 LakeFS 等）中批量导入数据文件，从公有数据管理服务中批量导入数据文件，还支持用户自定义部署数据采集服务（例如一些流量采集等服务），自动化生成数据集，全面打通数据获取通道，让安全数据的来源更加广泛。

4.1.2 数据预处理

4.1.2.1 定义内涵

用户从多种渠道收集的数据可能包含噪音，或是存在不一致、不完整等问题，无法直接进行训练。为了提高数据的质量，在将数据交给模型训练之前，需要对数据预处理。数据预处理是指在数据分析之前，对数据进行的清洗、筛选、转换、特征选择和提取等一系列必要处理操作的过程。

4.1.2.2 技术背景

数据预处理的常见方法有数据清洗、数据转换、特征选择、数据降噪和数据增强。

数据清洗通常是指对收集得到的数据集通过丢弃重复数据、补全缺失数据、异常值处理

等方式，清洗掉数据集里的脏数据，完成清洗过程。数据清洗旨在提高数据质量，数据清洗的效果会直接影响到后续数据分析、模型预测的效率和准确率。

数据转换是将数据从一种形式转变为另一种形式的数据预处理方法。由于数据源的多样性、数据的异构性、数据分析的要求等各种原因，在进行数据集成和管理之前，需要对数据进行转换。数据转换根据需求对数据进行类型转换、语义转换、粒度转换、离散化、数据压缩等操作，转换方式有平滑处理、泛化处理、合计处理、规格化处理等。

特征选择旨在筛选出给定数据集中与当前机器学习任务有关的特征。复杂数据中存在大量特征，并非所有特征对模型任务都是有用的，去除不相关的特征能节省计算和存储开销，降低过拟合风险，提高模型性能。常见的特征选择算法分为过滤法、包裹法和嵌入法，不同的算法根据不同的观察变量和特征评分规则对特征进行筛选。

数据降噪对数据集中的噪声进行移除。噪声在实际数据中无法避免，噪声的存在会对模型产生误导。常使用的数据降噪方法有聚类、降维、回归等。

数据增强旨在扩充有限的数据，增加训练样本的数量和多样性，进而提高模型的泛化能力。根据在机器学习流程中适用的阶段，数据增强分为离线增强和在线增强，其中离线增强直接对数据集采取增强操作，在线增强在获得批量数据后，对批量数据进行增强。常见的数据增强方案有 AugMix[8]，CutOut[9]，MoEx[10]，MaxUp[11]，Mixup[12]，CutMix[13] 等。

4.1.2.3 思路方案

在安全领域的研究中我们发现，很多数据预处理的步骤，在不同的场景下中都可以相互借鉴，甚至可以进行直接复用。例如，对于加密流量相关的数据，当算法工程师获取到一批加密流量的 pcap 包之后，不论他们面临的任务是加密流量的精确识别、异常检测，还是恶意软件的检测，对 pcap 包最原始的数据预处理思想是大同小异的，例如采集流级特征、包级特征等并将其转化成 csv 文件。SecXOps 对在安全领域中一些常见的通用数据预处理方式进行了模块化封装，从简单的数据去重、文件合并、数据清洗，到高级的特征提取，如 pcap 文件转 csv 文件，数据降维等，支持用户通过手动设置相关参数，对当前数据进行处理。

以 pcap 文件转 csv 文件为例，传统基于机器学习的网络流量分析严重依赖人工，在实践中，获得特征、模型和参数的最优组合通常是一个迭代的过程，这个过程有一些弊端。首先，数据的合适表示和特征选择对于流量分析任务是十分重要的，但即便有专业领域知识，特征工程仍然是一个脆弱且不完善的过程，人工分析时可能会忽略不够明显的或包含复杂关系的

特征；其次，网络环境复杂多变，流量模式的变化带来特征的失效；最后，对于每一个新的流量检测或分类任务，都需要重新设计新的特征，选择合适的模型，并重新调整参数。为了避免这些问题，本节中展示一种使用 SecXOps 实现的加密流量分析方法。

对于许多分类问题，数据表示与模型选择同等重要，所以在应用机器学习方法时，如何对数据进行表示和编码是非常重要的。对于网络流量数据的编码需要满足以下几个要求：

（1）完整的表示。该方法的目标不是选择特定的特征，而是一种统一的数据编码，以避免依赖专家知识，所以需要保留包含包头在内的所有数据包信息；

（2）固定的大小。许多机器学习模型的输入总是保持相同的大小，所以每个数据包表示都必须是常量大小；

（3）固有的规范化。当特征被归一化后，机器学习模型通常会表现得更好，也能减少训练时间并增加模型的稳定性，所以如果数据的初始表示本身就是规范化的，将会非常方便；

（4）一致的表示。数据表示的每个位置都应该对应于所有数据包包头的相同部分，也就是说，即使协议和报文长度不同，特定的特征总是在数据包中具有相同的偏移量，对齐后的数据都能让模型基于这样的前提来学习特征表示。

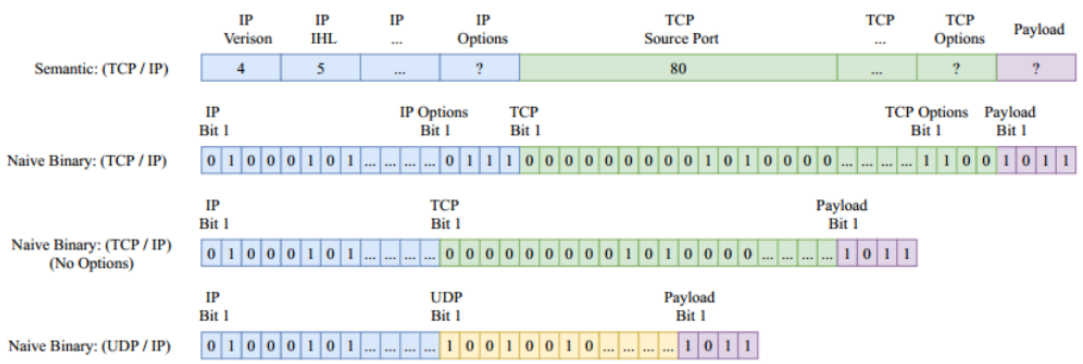

图 5　网络流量表示方式 [14]

如上图所示，网络流量表示的主要方式包括语义表示法和朴素二进制表示法。

（1）语义表示法：每个报头都有各自的语义字段，但它不保留具有区分度的可选字段的顺序，同时需要领域专业知识来解析每个协议的语义结构，另外，尽管拥有这些知识，后续还是不可避免地要进行繁琐的特征工程；

（2）朴素二进制表示法：使用数据包的原始位图表示来保持顺序，但是忽略了不同的大小和协议，导致两个数据包的特征向量对同一特征具有不同的含义，这种不对齐可能会在重要特征的地方引入噪声而降低模型性能，同时也因为无法将每一位都映射到语义上而导致结果的不可解释。

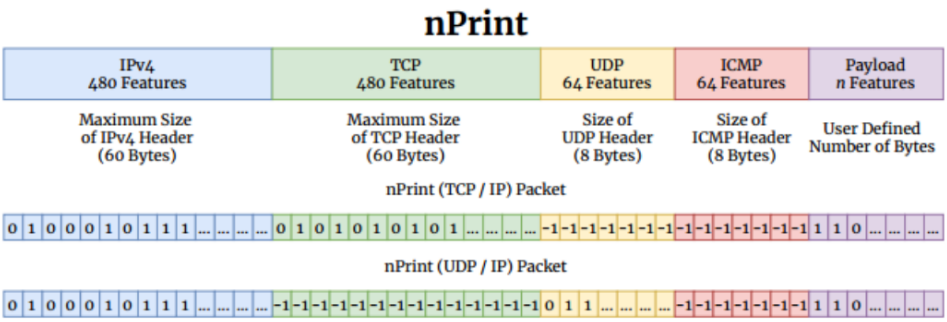

图 6　nPrint 方法 [14]

以上两种表示方法都无法满足统一化数据表示的需求，如上图所示，SecXOps 内置的数据预处理模块 nPrint，解决了单一表示方法无法统一化表示数据的问题。首先，它会保证任何数据包都可以被完整表示而不丢失任何信息；然后，使用内部填充确保每个数据包以相同数量的特征表示，并且每个特征具有相同含义，这种在位级上可解释的表示使我们能够更好地理解模型；其次，直接使用数据包的位，区分于某个位被设置为 0，将不存在的包头用 -1 填充；最后，每个数据包都用相同数量的特征表示，对于给定的网络流量分析任务，将载荷设置为可选的字节数。此外，nPrint 具有模块化和可扩展的特性，不仅可以将其他协议添加到表示中，也可以将一组数据包表示串联起来构建多包的 nPrint 指纹 [14]。

经过 nPrint 处理之后，即可将 pcap 包转成 csv 文件，随后可以进行进一步的特征处理或直接进行算法训练。

除了上述提到的几种数据处理方式之外，SecXOps 还支持用户将个人编辑的数据预处理模块进行封装并重复使用，实现定制化的数据清洗、数据增强等功能。

4.1.3　数据共享

4.1.3.1　定义内涵

数据共享是指在多个用户或多个程序之间遵循一定规则共同享用数据，并进行各种操作、运算和分析的一种技术。数据共享包括数据发布、接口、交换等内容。

4.1.3.2 技术背景

随着数字经济成为拉动全球经济增长的新引擎，大数据成为经济中重要的生产要素，大数据分析能够推动科学决策、改善用户体验、促进产品的创新。

尽管企业掌握了海量数据，但不断增长的数据需求和尚未开发的数据存量之间横亘着共享不足的数据壁垒，数据供给不足、数据共享受阻等因素阻碍着数据要素市场的构建。而大数据的共享能够进一步挖掘数据的潜在价值。数据共享是数字经济发展的重要基础，一方面，数据共享增强了数据供给能力，降低了数据采集成本；另一方面，数据共享是数据利用方式之一，实现了数据的二次开发和重复利用。出于数据共享的重要性和为企业带来的效益，数据共享逐渐成为促进数字经济发展的核心问题，在工业互联网行业的发展方兴未艾。

安全场景下，安全数据在面对相关项目需求时显得匮乏，数据开放共享更为必要。当安全数据在使用和流动时才能产生价值，在良好的、安全可控的数据共享体系的保障下，数据共享能够更大限度挖掘安全数据价值。

4.1.3.3 思路方案

SecXOps 的数据中心允许用户在个人数据集臻于完善后进行公开分享，公开后的数据集会在公开数据列表中展示出来，供用户查看详情。下图是目前 SecXOps 平台的一部分公开数据。在详情页可以看到数据的基本信息，基本格式，进入到数据集内部可以查看每一个版本及其对应的数据文件。

图 7　SecXOps 公开数据中心

在创建数据集时，SecXOps 允许用户创建标签，每一份数据集都可以创建多个标签，因此，在用户需要挑选数据集时也可以通过标签搜索的方式进行筛选。同时公开数据库中还支持名称关键字搜索、创建时间排序等常用操作，让用户快速找到合适的数据集。用户选取到合适的数据集及版本后，可以选择复制数据集及对应的版本，此时 SecXOps 会在指定的 namespace 下创建一个新的 PVC，同时创建一个 Job，将原数据版本所在的 PVC 和新数据所在的 PVC 同时挂载到该 Job 所创建的 Pod 上，将原数据从其所在的 PVC 上复制到新的 PVC 上。当 Job 运行结束后，相关的动态运行资源会被直接删除，此时新的 PVC 即可被用户访问到。用户访问数据时，平台根据用户的选择创建前几节提到的 ReplicaSet，供用户对数据进行上传、下载、编辑和删除等操作。此时用户对数据的操作不会影响到被复制的原始数据，在确保共享的前提下实现隔离。

4.2 安全分析模型自动化调优

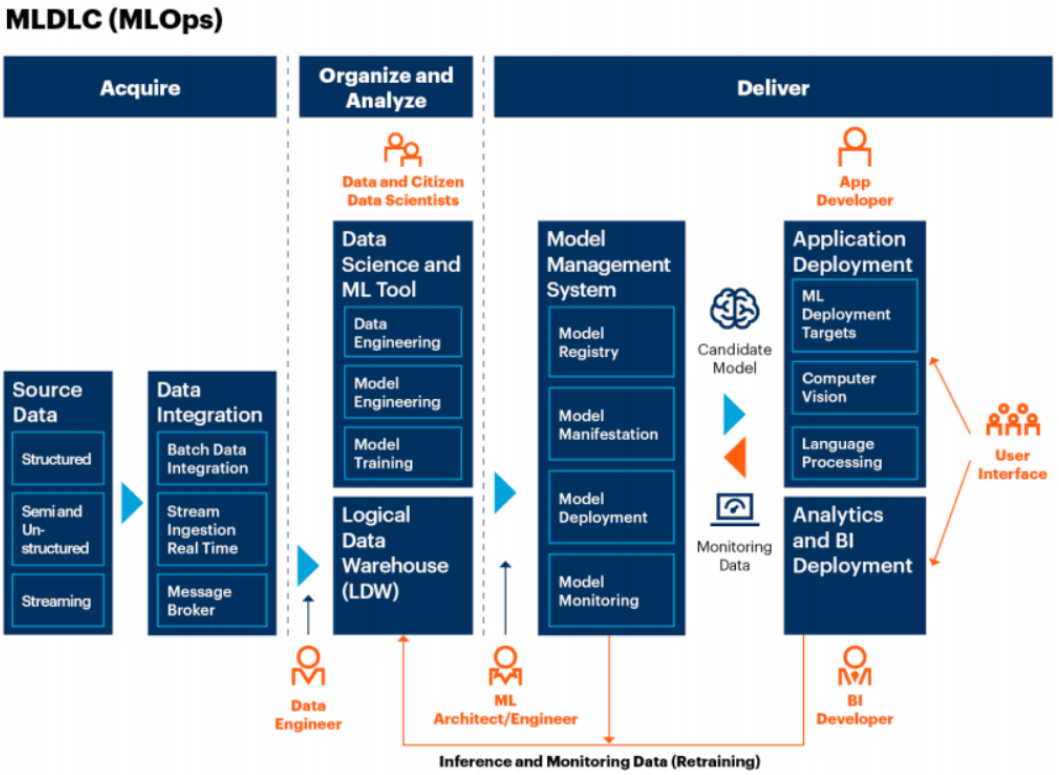

图 8 MLOps[7]

MLOps（Machine Learning Operations）是一种人工智能的工程实践，是面向机器学习项目的研发运营管理体系。旨在实现 ML 管道的操作、ML 模型的部署和管理标准化，支持 ML 模型的发布、激活、监控、性能跟踪、管理、重用、更新、维护和治理。

MLOps 位于机器学习、软件工程和数据工程的交叉点，打通了开发团队、运营团队和 AI 团队之间的屏障，建立一套标准化、自动化的模型开发、部署、运营流程，帮助企业更快、更有效地开发和完善 ML 模型。

MLOps 在机器学习项目整个生命周期的所有阶段中，通常提供数据分析、特征工程、模型调优、模型实验、模型测试、模型训练、模型的快速部署等相关功能和服务。MLOps 通过信息共享促进团队协作沟通，借助自动化、可重复的工作流以及可重用的资产提高效率，减少机器学习项目的风险。

4.2.1 超参数调优

4.2.1.1 定义内涵

在机器学习的上下文中，超参数是在开始学习过程之前设置值的参数，在神经网络训练过程中不会发生改变。通常情况下，在机器学习过程中需要对超参数进行优化，给学习器选择一组最优超参数，以提高学习的性能和效果。

4.2.1.2 技术背景

AI 模型，尤其是深度神经网络，在训练阶段的关键在于选择合适的网络架构、适当的正则化处理和超参数的调优。训练中需要调试的超参数有很多，如学习率、批量大小、动量、正则化、网络层数、隐藏层单元数等等。超参数的值不仅会影响到模型质量，对于确保模型收敛和准确率之间的平衡也至关重要。

超参数调优旨在寻找最优的超参数组合，属于非凸优化问题。主流的超参数调优算法有网格搜索、贝叶斯优化、模拟退火算法和遗传算法。

网格搜索 [15]，是一种穷举搜索手段，在指定参数范围内，按步长依次调整参数，并使用调整后的参数训练模型，循环遍历每一种可能的参数值，将在验证集上表现最好的解作为最终的超参数值。网格搜索能够找出指定参数范围内的最优参数，但超参数数量的增长会导致计算复杂度的指数级增长。

贝叶斯优化 [16] 是目前超参数优化领域中的最优框架。贝叶斯优化采用高斯过程，根据已有的参数信息不断更新先验。与网格搜索相比，贝叶斯优化调参迭代次数少，速度更快。

模拟退火算法[17]是一种启发式算法，其思想源于材料统计学，从高温环境开始，非常缓慢地降温，使粒子在从高能量状态向低能量状态转变的每个温度下都能达到热平衡，直至系统完全冷却，粒子转变为低能状态的晶体。模拟退火算法在搜索和优化超参数的过程中引入了随机因素，在搜索到超参数的局部最优解后，以一定的概率（该概率随着时间逐渐降低）接受比当前解更差的解，反复几次后，超参数的值有一定概率跳出局部最优解，达到全局最优解。

遗传算法（Genetic Algorithm，GA）[18]，也是一种启发式算法。遗传算法将超参数调优问题模拟为生物进化过程，经过选择、交叉、突变后产生下一代的解，逐步淘汰适应度函数值（评价指标）低的解，增加适应度函数值高的解，经过多代进化后，可能会出现适应度函数值很高的个体，也即超参数的最优值。

4.2.1.3 思路方案

自动机器学习是近年来火热的应用和研究方向，各种自动机器学习工具也层出不穷，它们各有优点与局限性。有的聚焦于算法，但不支持分布式训练；有的功能强大，但没有易用的用户界面，学习成本较高；有的只支持特定领域，不提供通用功能；还有的只能在云端使用。

SecXOps 集成了 NNI 自动化深度学习包，提供了基于 python 的 SDK，支持多种框架，如 pytorch、tensorflow、scikit-learn、lightGBM 等主流框架和库。同时提供了简单易用的命令行工具和友好的 web 用户界面。在超参数搜索算法上，NNI 不仅提供了 TPE、SMAC、进化算法等优秀算法，还提供了遍历、批处理、随机、Hyperband 等十多种算法。另外，还支持自动终止低效实例，加速学习过程。

```
{
    "batch_size": {"_type":"choice", "_value": [16, 32, 64, 128]},
    "hidden_size":{"_type":"choice","_value":[128, 256, 512, 1024]},
    "lr":{"_type":"choice","_value":[0.0001, 0.001, 0.01, 0.1]},
    "momentum":{"_type":"uniform","_value":[0, 1]}//动量：计算梯度的指数加权平均数，并以此来更新权重
}
```

图 9 超参搜索空间定义

启动一个超参调优任务首先需要定义搜索的超参空间，如上图所示；然后，在需要调参的网络启动之前，通过 NNI 的接口读取参数并在训练中将精确度等指标传入 NNI；最后进行实验所需的 yaml 文件编写，配置好要使用的调参算法等，即可进行超参数调优。

```yaml
searchSpaceFile: search_space.json
trialCommand: python3 mnist.py  # ubuntu:"python3",win10:"python"
trialGpuNumber: 0
trialConcurrency: 1
tuner:
  name: TPE # TPE,Random,Anneal(退火),Evolution(进化),Grid Search(网格搜索)...
  #优缺点: https://nni.readthedocs.io/zh/stable/Tuner/BuiltinTuner.html
  classArgs:
    optimize_mode: maximize # minimize
trainingService:
  platform: local
```

图 10　实验配置 yaml 文件示例

另外，NNI 的超参调优不仅能用于机器学习，对于各类系统、数据库的繁杂参数都可以根据实际场景进行有针对性的调优。使用过程和超参调优非常类似，通过 python 为系统传入不同的参数配置，然后将确定的调优指标（如读写速度，磁盘空间大小等）回调给 NNI 即可。

SecXOps 平台允许用户自由使用 NNI 提供的服务进行项目开发，同时在项目运行对应的 pod 预留了映射端口供用户访问。当超参数调优服务启动后，容器内的 8888 端口会映射到集群的某个随机端口，如下图所示。

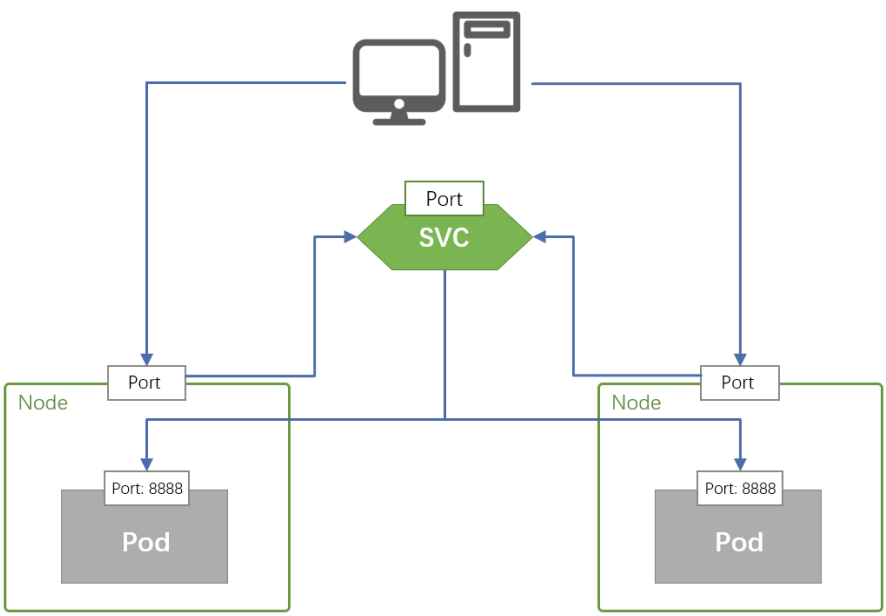

图 11　Kubernetes 端口映射

4.2.2 神经网络架构搜索

4.2.2.1 定义内涵

神经网络架构搜索是为给定数据集自动找到一个或多个架构的任务，这些架构将为给定的数据集生成具有良好结果的模型，其本质是在高维空间的最优参数搜索问题。

4.2.2.2 技术背景

深度学习模型的使用越来越大众化，在很多行业中都是不可或缺的。然而，高效神经网络的实现通常需要架构的知识和大量的时间，在不停的迭代过程中使用知识探索合理的解决方案。神经网络的形式和结构会根据具体需要而有所不同，所以针对不同的任务和不同需求需要设定特定的结构。但是以试错的方式设计这些网络是一项耗时且乏味的任务，不仅需要架构技能还需要专业领域的知识。一般的情况下，专家们会利用他们过去的经验或技术知识来创建和设计神经网络。

在大部分行业，都会关注模型效率（这里的效率不仅指执行效率，还包括了开发效率）。为了使神经网络泛化而不过度拟合训练数据集，找到最优的结构是很重要的。但是在生产力比质量更重要的时代，一些行业忽视了他们模型的效率，并且只满足于实现了他们的目标的第一个模型（能用就行）而没有进一步优化模型的性能和效率。寻找合适的网络架构不仅需要大量的时间成本和专业的架构设计技能，而且容易陷入次优解。由于缺乏时间或架构专业知识，许多行业不具备通过“足够”的模型充分利用其数据的潜力。

神经网络架构搜索（Neural Architecture Search，NAS）[19] 是一种基于优化的算法，旨在为特定数据集上的特定任务寻找神经网络的最佳结构，解决预定义的架构不完善的问题。NAS 将神经网络架构的设计从繁杂的手动方式转变为自动化方式，NAS 在图像分类、目标检测、语义分割等任务上的效果能够达到甚至优于手动设计的网络架构。

NAS 在定义的指数级搜索空间中实施特定的搜索策略，寻找在不可见数据上实现高预测性能的神经网络架构。用于探索网络结构空间的搜索策略包括随机搜索、贝叶斯优化、进化方法、强化学习和基于梯度的搜索方法。

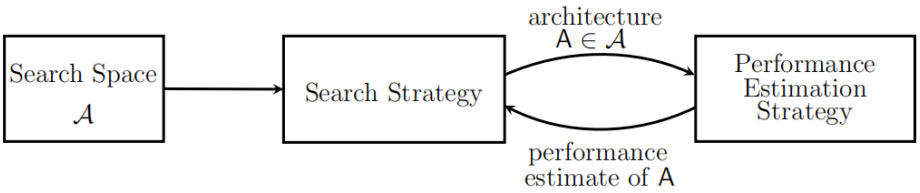

图 12 神经网络架构搜索 [19]

NAS 通过自动化探索多种潜在的网络架构，为 MLOps 的模型选择提供了更多的灵活性和更高效的解决方案。

4.2.2.3　思路方案

NNI 中实现了 ENAS、DARTS、P-DARTS 算法，并提供 one-shot 算法的接口。另外，还支持网络模态（Network Morphism）这样的经典搜索方法 [20]。

NAS 方法探索了许多具有可变复杂性的潜在解决方案，因此在计算上非常昂贵。他们的搜索空间越大，需要测试、训练、评估的架构就越多。这些方法需要大量资源和时间才能找到足够好的模型，因此在创建神经网络架构搜索任务时，SecXOps 将更倾向于将该项目所在的 Pod 分配到空闲计算资源更多的 Node 节点上。

同超参数调优类似，神经网络架构搜索也需要预先编写 yaml 文件，对实验进行配置，如图所示。

```
horName: default
erimentName: example_mnist
alConcurrency: 1
ExecDuration: 1h
TrialNum: 10
oice: local, remote, pai
iningServicePlatform: local
rchSpacePath: search.json
Annotation: False
er:
odeDir: ../../../tuners/random_nas_tur
lassFileName: random_nas_tuner.py
lassName: RandomNASTuner
al:
```

图 13　实验 yaml 配置

在训练过程中，程序会实时输出模型结果，SecXOps 同样为此类任务建立起了端口映射，允许用户能够访问并查看自己的任务完成进度，如下图所示。

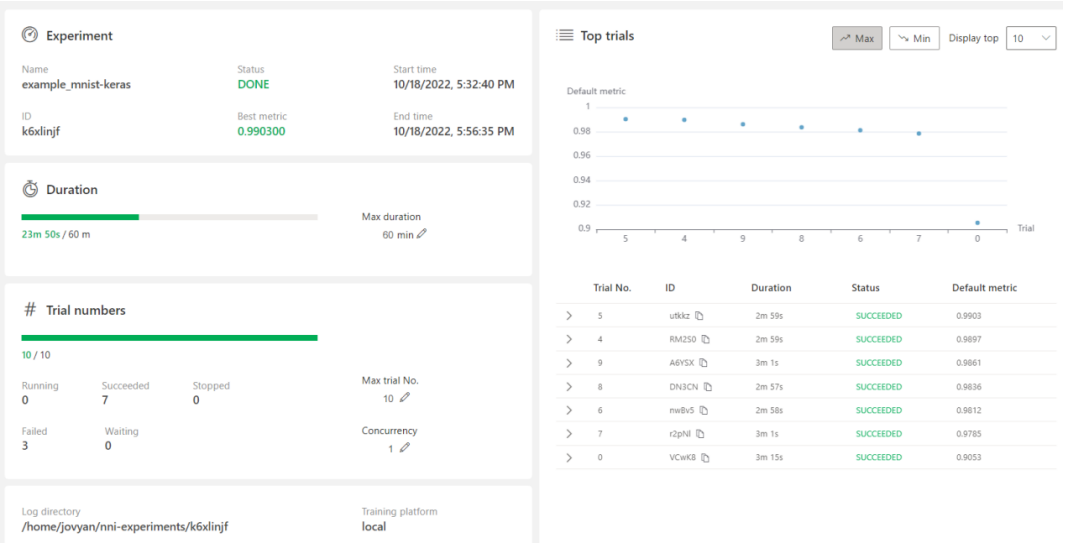

图 14　架构搜索

同时，点击对应的任务支持用户查看搜索到的神经网络架构。

4.2.3　模型自动化生成

4.2.3.1　定义内涵

模型自动化生成是对指定的经过数据预处理后的数据文件，按照预设的一定参数，自动训练出一系列机器学习模型。通过多个模型的对比，保存效果和性能较为优秀的模型，或者按照一定的集成学习策略，如 Stacking、Bagging 等，将多个简单模型进行集成，形成效果更优的模型。

4.2.3.2　技术背景

目前有很多开源的 AutoML 项目已经实现了模型的自动化生成，如 H2O、Autogluon 等。作为一个用于分布式、可扩展机器学习的内存平台，H2O 能够与 Hadoop 和 Spark 等大数据技术无缝协作。H2O 提供了许多流行算法的实现，例如广义线性模型 (GLM)、梯度提升机（包括 XGBoost）、随机森林、深度神经网络、堆叠集成、朴素贝叶斯、广义加性模型 (GAM)、Cox 比例风险、K- Means、PCA、Word2Vec，以及全自动机器学习算法。通过输入一系列超参数即可完成模型的训练，对零代码基础的用户非常友好。

Autogluon 是亚马逊推出的一个新的开源库，开发人员可以使用该库构建包含图像、文本或表格数据集的机器学习应用程序。使用 Autogluon，只需编写几行代码就可以利用深度学习的力量来构建应用程序。

大部分 Automl 框架是基于超参数搜索技术，例如基于贝叶斯搜索的 hyperopt 技术等，Autogluon 则依赖融合多个无需超参数搜索的模型。例如 Stacking 方法：Autogluon 在同一份数据上训练出多个不同类型的模型，这些模型可以是 K 均值聚类、决策树、核方法等等，这些模型的输出进入到一个线性模型里面得到最终的输出。多层 Stacking：将多个模型输出的数据，合并起来，再做一次 Stacking。在上面再训练多个模型，最后用一个线性模型做输出。为了避免后面层过多拟合数据，多层 Stacking 通常配合 K 折交叉 Bagging 使用，也就是说这里的每个模型是 K 个模型的 Bagging。它对下一层 Stacking 的输出，是指每个 Bagging 模型对应验证集上输出的合并。

K 折交叉 Bagging 则是源于 k 折交叉验证。Bagging 是训练同类别的多个模型，他们可能使用不同的初始权重或者数据块，最终将这些模型的输出做平均来降低模型的方差。相同的初始参数，训练多次，对每次的误差求平均后作为这些初始参数的最终误差，可以最大化利用数据集，同时有效避免过拟合和欠拟合现象。K 折交叉 Bagging：每一折对应不同的初始参数，训练出多个模型，对结果求平均。

4.2.3.3 思路方案

SecXOps 对 Autogluon 在表格数据上进行操作的流程进行了封装，并在前端进行了可视化展示。用户只需要选择训练集、验证集、测试集、标签列的名称索引以及最终模型保存策略（如最优性能、最优质量、性能与质量平衡等）即可进行一键式的模型生成，可用于后续的模型服务中去。

4.3 安全分析模型核心服务部署

ModelOps 对所有的人工智能模型（图形模型、语言模型、基于规则的模型）以及决策模型的整个生命周期进行管理，确保对生产中的所有模型进行独立验证和问责，其核心功能涵盖了模型存储、模型测试、模型回滚和跨生产环境的 CI/CD 集成。

尽管在核心功能上 ModelOps 与 MLOps 类似，但 ModelOps 能够为专家提供自主权来解释结果和验证生产中人工智能模型的 KPI。而在不完全依赖数据科学家和 ML 工程师的情况下，ModelOps 使模型推理变得可行。

ModelOps 在生产过程中管理 AI 模型的部署、治理和监控，协助企业在实际生产环境中扩大人工智能模型的应用规模，为企业的所有生产模型提供全面的运营能力和治理能力。

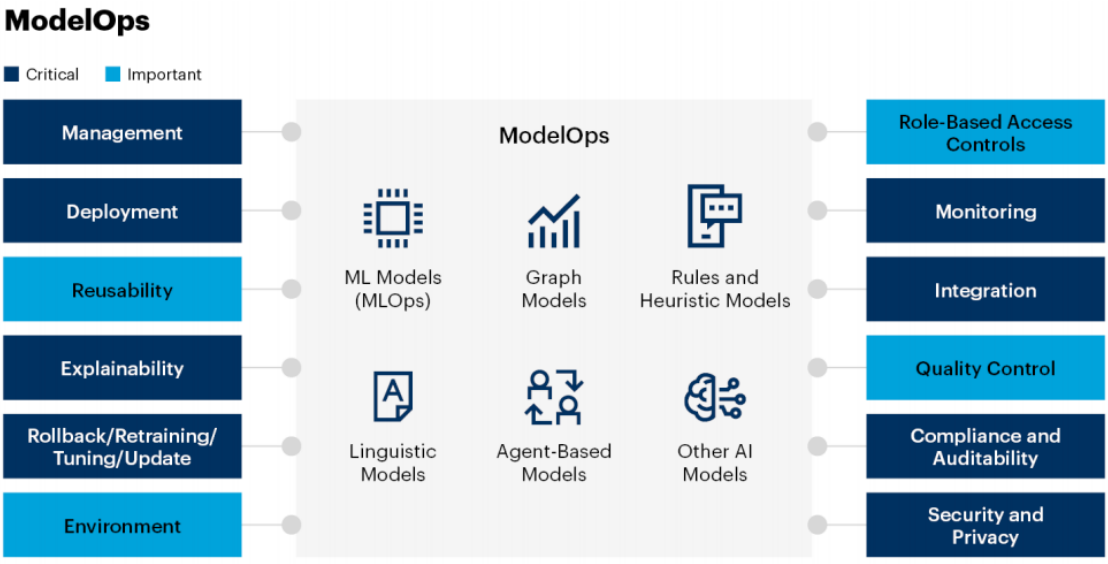

图 15　ModelOps[7]

4.3.1　模型推理

4.3.1.1　定义内涵

模型推理是指使用训练完成的 AI 模型，通过输入数据，获取经过模型处理后产生结果的过程。

4.3.1.2　技术背景

在经过模型训练、模型封装、模型部署上线过程后，ModelOps 能够将模型部署为模型推理服务。

推理服务的常见部署方式有三种 [21]：

1. 在物理机上部署。使用 Flask、Tensorflow Serving 等技术将训练好的模型文件和对应的推理代码封装为服务包，拷贝到物理机上实施部署，最后将服务接口和调用方法提供给模型推理服务。

2. 在虚拟机上部署。将物理机划分成多个虚拟机，使用 Flask、Tensorflow Serving 等技术将训练好的模型文件和对应的推理代码封装为服务包，部署在虚拟机上（为避免资

源冲突，一般一个虚拟机上部署一个模型服务）。考虑到大规模并发请求可以引入负载均衡进行流量分发。最后将对外的服务接口和调用方法提供给模型推理服务。

3. 基于容器技术部署。通过 Docker 封装、打包模型推理任务，并隔离不同模型服务的运行环境，构建 Kubernetes 集群对推理服务进行编排，对基础资源进行合理分配和调度，保障服务的分布式容灾和资源的弹性伸缩功能。

另外，针对不同的推理任务，用户需要在 ModelOps 平台提供的多种模型之间进行选择，ModelOps 平台通过提供模型筛选建议或自动化筛选策略，协助用户选择最优模型并简化后续模型更新和测试服务。

Tensorflow Serving 是针对机器学习模型灵活且高性能的服务系统，专为生产环境而生。Tensorflow Serving 基于 Docker 快速部署模型，方便地进行版本迭代和容器管理，此外 Tensorflow Serving 基于 C++ 构建，摒弃了一些模型训练之中的冗余参数和设计，让模型可以高性能地运行在 CPU/GPU 服务器上。处理机器学习的推理方面，Tensorflow Serving 在训练和管理生命周期之后进行建模，并通过高性能的参考计数查找表为客户提供版本化的访问。Tensorflow Serving 提供与 Tensorflow 模型的现成集成，但可以轻松扩展，以便服务于其他类型的模型和数据。

Tensorflow 的 Serving 功能有很多特点：

- 可以同时服务多个模型或相同模型的多个版本
- 允许部署新模型版本，而无需更改任何客户端代码
- 支持金丝雀发布和 A/B 测试
- 高性能、低消耗、最小化推理时间
- 内置调度程序，能分批处理各个推理请求，以便在 GPU 上联合执行，并具有可配置的延迟控制

尽管 Tensorflow 的官方 Serving 服务有着极高的性能，却无法对未经向量化的数据进行运算。因此在使用 Tensorflow 的 Serving 功能时需要配置额外的数据预处理和数据后处理服务。

4.3.1.3 思路方案

SecXOps 支持用户在平台上部署 Tensorflow 的 Serving 功能，当用户有其它模型推理需

求时也支持用户封装自定义模块进行推理。通过预设 Dockerfile 能够实现模块的封装，与模型进行一对一绑定后，即可部署一个完整的模型推理服务。

用户创建模型推理服务时可以指定模型，SecXOps 将自动找到对应的模型推理模块，按照该模块内包含的 Dockerfile 创建一个 Pod，该 Pod 被包含在一个 Job 中。Kubernetes 中的 Job 和上文中提到的 ReplicaSet 相同，是一类 Pod 控制器，它会创建一个或者多个 Pod，并将继续重试 Pod 的执行，直到指定数量的 Pod 成功终止。随着 Pod 成功结束，Job 跟踪记录成功完成的 Pod 个数。当数量达到指定的成功个数阈值时，任务才结束。删除 Job 的操作会清除所创建的全部 Pod。

模型推理任务完成后，SecXOps 平台会自动采集 Job 中的 Pod 日志，将最终的输出或错误信息打印给用户。

4.3.2 模型更新

4.3.2.1 定义内涵

本节的模型更新是指在模型训练完成并正式上线后，由运维人员采集并提供新的数据对原有模型进行再训练、更新参数的过程。

4.3.2.2 技术背景

随着时间的推移，由于周期性事件、突变等状况的发生，当下的数据集和之前用于训练模型的数据集分布可能不再一致，这种现象被称为数据分布漂移。由于数据分布漂移现象的存在，基于过往数据训练的模型在推理上的准确性受到了影响，不再适用于当下数据上的任务，因此需要对模型进行更新。

在监控到生产数据分布发生变化后，ModelOps 提供两种主流模型更新服务：

- 全量更新。在所有数据上对模型进行重新训练，这种更新方式保障模型学习到更全面的分布，最大限度提高模型准确率。
- 增量更新 / 微调。在新数据上对模型进行微调，这种更新方式对计算资源的需求更小，模型的更新速度也更快，但在实际应用场景中，微调无法像全量更新那样提供更好的模型性能 [22]。

4.3.2.3 思路方案

和模型推理类似，SecXOps 支持封装模型更新模块，通过模块对应的镜像创建容器，将

代码、数据和模型挂载到容器中，进行模型的更新操作。

SecXOps 为模型更新服务开发了一个 python 的 SDK，允许用户通过调用 SDK 中的函数，将训练过程的评价指标输出，在前端进行可视化的展示，如下图所示。

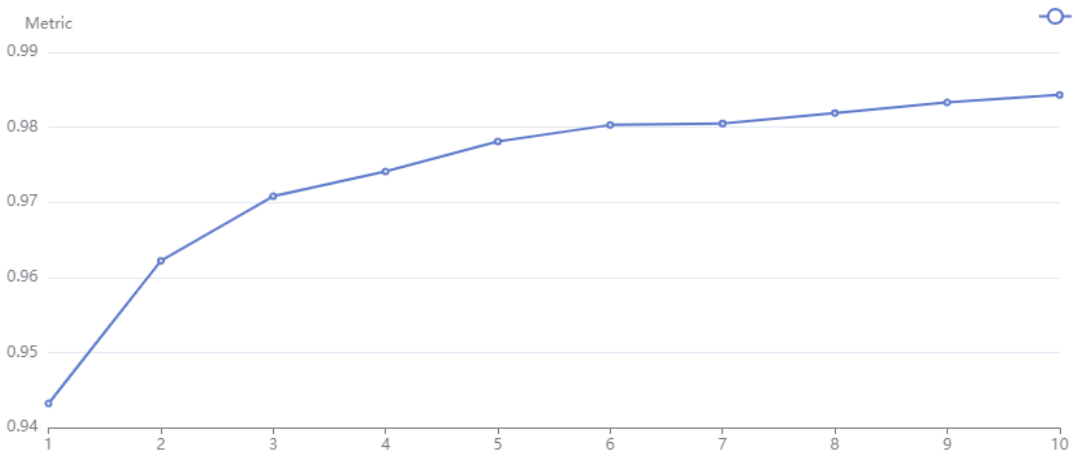

图 16　模型更新折线图

图上的横坐标表示用户选择输出的轮次序号，纵坐标表示输出的指标，这里的指标提供多种选择，如精确度、召回率等等。通过折线图的对比，用户可以实时观测到更新结果，并选择在合适的位置停止。用户可以通过模块中封装的相关参数，选择保存模型的方式，保存下来的模型最终会存储在原有模型的仓库中，作为一个新的版本。

4.3.3　模型测试

4.3.3.1　定义内涵

模型测试是指针对某个或某几个版本的模型，通过输入指定数据集对模型的输出结果进行评估，查看精确度、召回率等各项指标的过程。

4.3.3.2　技术背景

无论是在机器学习模型的离线开发训练阶段还是在线提供服务阶段，都需要对模型质量进行评估测试，以保障模型在性能和功能上满足业务需求。

离线阶段的模型测试的指标主要是精确度、召回率、F1。在混淆矩阵中，统计了模型对样本的预测值和样本真实值之间不同状况的样本数，将真阳性记作 TP，真阴性记作 TN，假阳性记作 FP，假阴性记作 FN：

精确度（Precision），又称查准率，表示的是模型预测为正的样本在实际中也属于正类的概率：

$$精确度 = \frac{TP}{TP + FP}$$

召回率（Recall），又称查全率，表示的是在实际中为正的样本中被模型预测为正的概率，评估了分类结果的完整性：

$$召回率 = \frac{TP}{TP + FN}$$

F1 是对精确度和召回率的调和平均，综合评估了精确率和召回率：

$$F1 = \frac{2 * 精确率 * 召回率}{精确率 + 召回率}$$

而在线阶段的模型测试类型主要通过在线流量对多个模型进行 A/B 测试。

A/B 测试是一种随机测试，对 A 组（对照组）和 B 组（实验组）进行假设检验，通过测试环节的表现，决定最终采用哪一组方案。对新模型和处理生产流量的旧模型进行 A/B 测试，通过向各版本模型分配流量或调用各版本模型进行测试 [23]，比较哪个版本的模型更符合安全性和业务需求，这往往是模型测试的最后一个环节。在 A/B 测试中，需要测试模型变体，并比较各个变体的相对性能。如果新版本带来的性能优于或等同于原有版本，即可使用新模型替换原有模型。

4.3.3.3 思路方案

在模型测试方面，SecXOps 会将用户选择的数据集、模型及其对应模块的镜像进行一一组合，例如对同一项目中按照不同优化函数或训练了不同轮次的三个模型进行测试，测试数据挑选了两个不同来源的数据，SecXOps 即会创建六个不同的运行 job。这六个 job 会按照一定策略选取资源较为充足的 node 节点进行运行，最终的命令行输出会被采集，展示给用户。同时支持调用 SecXOps 内部的 SDK，将测试结果进行图表化的展示。

4.4 安全分析工作流定制

工作流（Workflow）是对工作流程及其各操作步骤之间业务规则的抽象、概括描述。工作流系统是以规格化的流程描述作为输入的软件组件，它维护流程的运行状态，并在人和应用之间分派活动。

为实现特定业务目标，工作流利用计算机在多个参与者之间按某种预定规则自动传递文档、信息或者任务。

市面上已经涌现了大量的开源工作流管理工具，包括 Airflow、DVC、Elyra、Flyte、Kale、Metaflow、Ploomber 等。其中，Jupyter lab 的插件 Elyra 提供交互式搭建工作流的功能，适合基于 Kubernetes 进行部署。

4.4.1 工作流设计

4.4.1.1 定义内涵

一个工作流模块是指在工作流场景下的一个可以执行的流程，其内部可能包含了数据、代码、模型、脚本以及代码运行环境等。通过将一个个工作流模块组合起来，并将不同模块之间的输入和输出进行对接，可以形成一个完整的工作流模板，进行一键式地运行。

4.4.1.2 技术背景

为了能够使模块成功运行，需要确保模块内拥有代码、代码运行时所用的环境、启动代码的命令等，这些信息可以被配置在一个 yaml 文件中。

Yaml 文件是一个可读性高，易于理解，用来表达数据序列化的格式。它的语法和其他高级语言类似，并且可以简单表达清单（数组）、散列表、标量等数据形态。它使用空白符号缩进和大量依赖外观的特色，特别适合用来表达或编辑数据结构、各种配置文件等。

Kubernetes 中的大量操作都可以通过 yaml 文件来进行。

4.4.1.3 思路方案

Jupyterlab 的一个工作流模块是容器内的一个代码文件，通过可视化地将一个个文件组合起来即可实现工作流的运行。而这种方式无法跨容器运行，因此 SecXOps 实现了一个从项目到工作流模块的快速封装功能，支持用户从项目中挑选代码或脚本进行模块化。在模块化过程中，SecXOps 会自动保存容器内指定的代码文件和脚本，同时为了确保代码能供正常运行，需要检测容器内是否存在 Dockerfile，确保了用户环境能够在工作流运行时进行加载。Dockerfile 准备完成后，程序就拥有了运行环境，启动命令则是在 yaml 文件中进行存储。SecXOps 允许用户在封装模块时填写启动模块所需要的指令，如“python3 train.py --input data.txt”、“sh start.sh”等等。在封装模块时，所有预设的参数都需要以 $ 符号附带参数名称进行替换，确保使用模块的用户能够自由指定参数。这些启动命令将被存储在 yaml 文件的启动配置项字段，工作流运行时会调用这些命令，让用户能够正常运行模块。

4.4.2 工作流运行

4.4.2.1 定义内涵

工作流运行是工作流模板的依次执行，在工作流运行时，用户可以随时取消或查看正在运行的任务。由于工作流运行的模板的不同，运行过程中可能会产生不同的新资源，如数据处理类型的工作流会产生新的数据集，AI 算法类的工作流会产生新的模型，甚至超参调优类型的工作流会产生新的模型训练代码，这些新生成的资源都可以在本模块中进行保存，来完善本平台的数据集库、模型库和算法库。

4.4.2.2 技术背景

在项目的 jupyterlab 中，用户可以自由安装 Elyra 搭建工作流。在使用 Elyra 搭建可视化工作流之前，用户首先需要基于容器和 Kubernetes 构建 Kubeflow。Kubeflow 提供面向机器学习业务的敏捷部署、开发、训练、发布和管理平台，通过集成开源生态中越来越多的项目，如 Jupyter， Knative， Seldon Core 等，搭建了机器学习应用从开发到上线到迭代的生命周期内的工具链，解决企业应用机器学习中遇到的和 DevOps 类似的工具链的困扰，因此，Kubeflow 也成为了当前云原生场景下主流的 MLOps 平台。

Elyra 利用 Kubeflow Pipelines 进行工作流的构建。使用 Elyra 中的 Pipeline Editor，用户可以将若干个 ipynb 文件组装成工作流，并在任何 Kubernetes 环境中的 Kubeflow 管道上运行。如本小节上述内容，将机器学习工作流程拆分为多个阶段后，每个阶段中的具体任务对应的代码被封装为镜像，作为 Elyra 中的 ipynb 文件。Elyra 根据流程之间的顺序或并行特性，连接各个 ipynb 文件，确保在工作流中，执行完上游节点的任务后，处理下游节点的任务。

4.4.2.3 思路方案

在 SecXOps 中，用户可以自由扩展 jupyterlab 的功能，因此可以轻易实现容器内工作流编排。然而当在用户需要实现跨项目跨容器的工作流运行时，SecXOps 支持将用户创建的模块以 DAG 图的形式进行编排，将一个个工作流模块编排起来即是将一个个容器编排起来，随后按照一定的资源动态调度策略在集群上运行，获取每一个模块的运行结果。

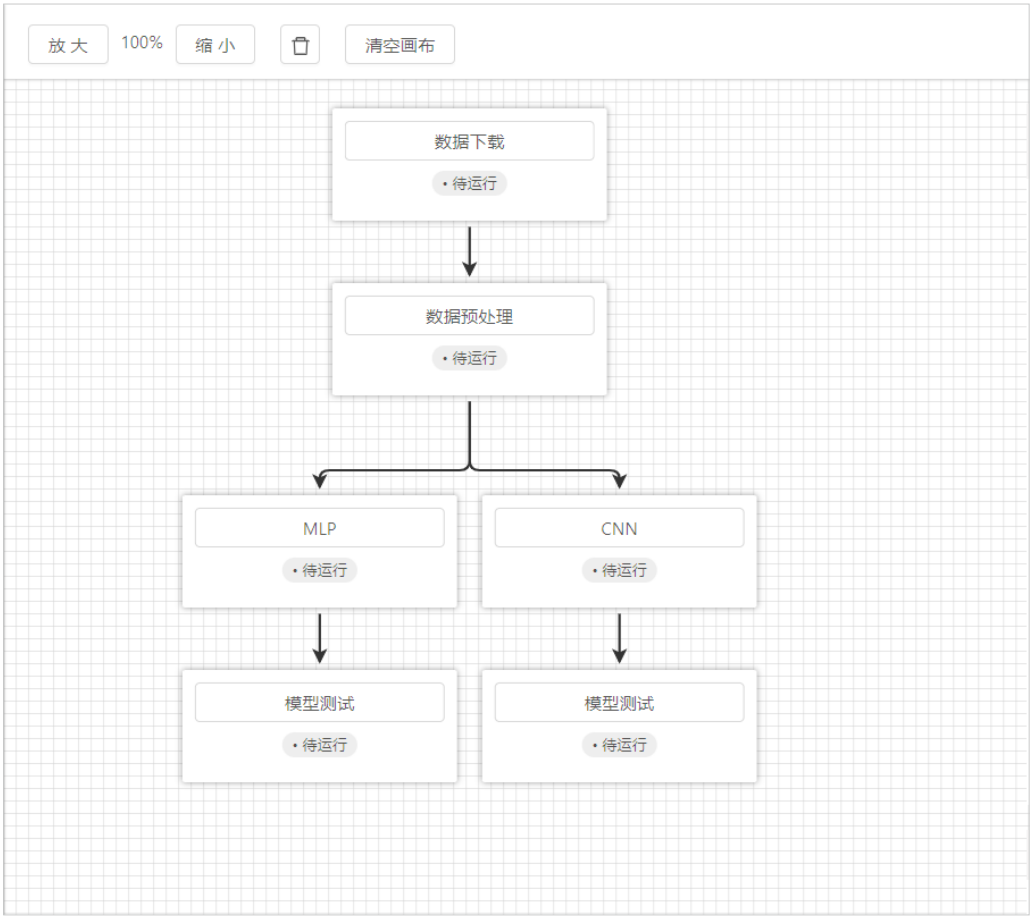

图 17　工作流模块编排

如上图所示，每个方块代表一个模块，模块的运行会按照其内部关联的 Dockerfile 创建一个容器，容器运行在以 Job 创建的 Pod 中。当一个模块运行结束后，该模块的后继节点会判断其全部前驱节点是否完成运行，若完成则开始启动该模块的运行，反之则继续等待。当一个模块运行失败时，该模块所在路径会被阻塞，用户需要重新修改参数或调整模块才能继续运行。

4.5　安全分析开发环境一键部署

Gartner 报告中指出 [7]，Platform Ops for AI 是一种编排和扩展运营人工智能的方法，利用跨数据、机器学习、人工智能和应用程序开发管道的多种最佳实践，为基于人工智能的系统创建一个高效的交付模型。Platform Ops for AI 支持将交付模块化、编排、发现和自治作为业务框架的组成部分。

当跨多个环境部署应用程序时，数据模型开发和运营、孤立的 DataOps 和 MLOps 之间的不匹配会给数据和概念漂移带来巨大挑战。DataOps 被用于管理数据管道，MLOps 和 ModelOps 被用于构建健壮的机器学习和 CI/CD 管道。Platform Ops for AI 是从数据到交付的平台级别的技术框架，是一种运维能力的整合，利用 DataOps、MLOps、ModelOps 以及 DevOps 的实践来确保端到端人工智能平台的安全性和治理，对齐数据、人工智能、机器学习模型部署管道，保障其一致性。

Platform Ops for AI 旨在为 AI 系统在搭建的所有阶段提供持续集成和持续交付，支持和保障系统的可重用性、结果可再现性、代码回滚、数据沿袭和安全环境。

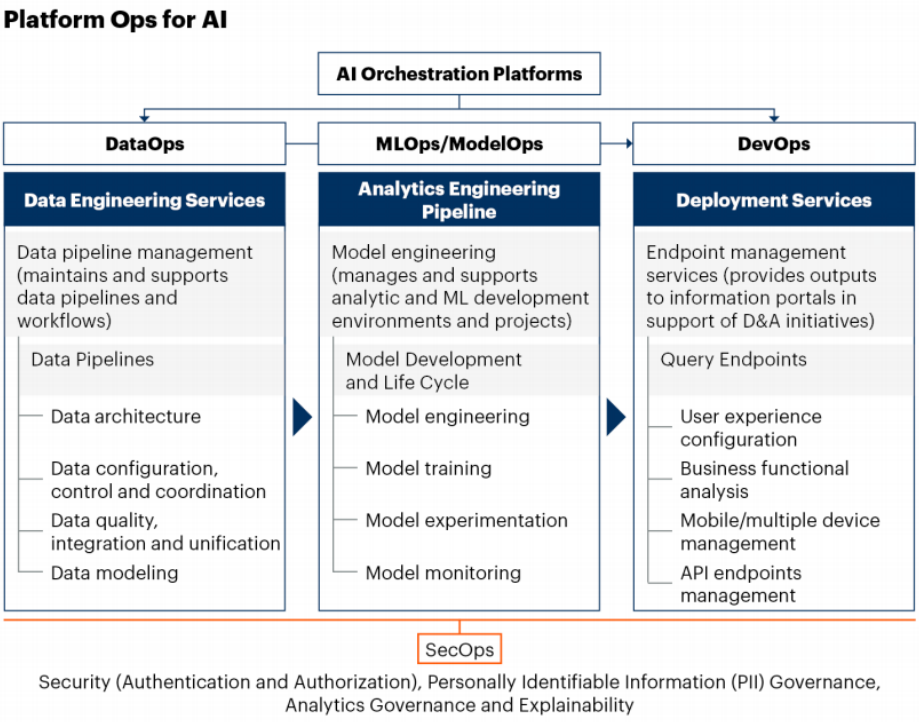

图 18　Platform Ops for AI[7]

4.5.1　环境容器化

4.5.1.1　定义内涵

容器作为一种轻量级虚拟化技术，实现了进程级别的封装隔离。容器化将应用程序和所有相关组件（库、框架和其他依赖项）打包隔离，确保了容器内的应用在任何系统环境和架构上得以无缝移动和运行，并在多种平台和基础架构上保持运行结果的一致性。

相比于传统的虚拟机技术，容器体积更小、启动速度更快，且在主机上运行的所有容器共享操作系统内核和资源，有着轻量级和可移植的特性，容器的出现使得应用的迁移变得更加简易。

4.5.1.2 技术背景

为了实现 DevOps 理念，容器化的引入至关重要，因此容器化也是搭建 Platform Ops for AI 的基础。以 Docker 为代表的容器对于开发人员和运营人员的重要性主要在两方面：一是持续集成和持续部署，轻量级的容器允许开发人员在本地建立容器堆来复制生产环境，并在容器堆上搭建和运行应用程序；二是利用容器的可移植性，实现应用程序的便携测试、部署。

为了管理容器的整个生命周期，容器编排的概念应用而生。容器编排自动执行容器的部署、管理、扩展和负载均衡等。常见的容器编排平台是 Kubernetes（k8s），是一个跨主机的容器管理平台，支持大规模的分布式、容器化管理。

4.5.1.3 思路方案

SecXOps 的一切功能都是基于 Kubernetes 和 Docker 实现的，如下图所示。

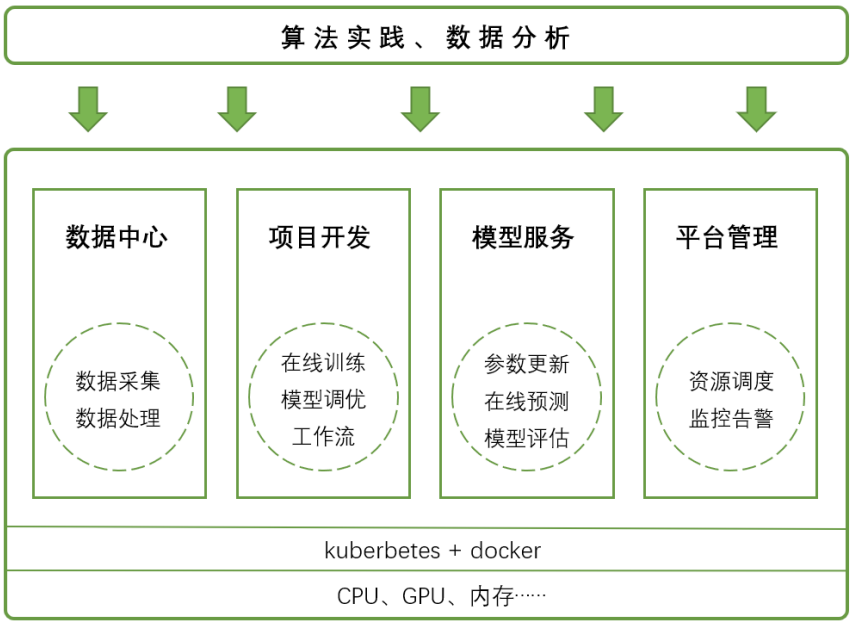

图 19 SecXOps 架构图

为了能够支撑用户自由配置环境并进行在线开发，SecXOps 平台在集群的每一个 node 上都预先下载好了一些常用的镜像，如 tensorflow2.9.1-gpu 等。在这些镜像中都内置了 jupyterlab 开发环境，还包含丰富的插件和工具，如 elyra 工作流相关插件、tensorboard 可视化工具等等，尽最大可能满足算法开发人员在代码开发过程中的种种需求。另外在用户需要扩展容器的功能时，也可以直接对开发环境进行修改，不会出现环境冲突的问题。

下图是 jupyterlab 的在线编程页面，其包含了 jupyter notebook 的所有功能，支持 python、JAVA、R 等多种编程语言以及 markdown、latex 等写作语言及公式输入，可以集编程与写文档于一身，同时也升级增加了包括多窗口编程在内的多种功能。

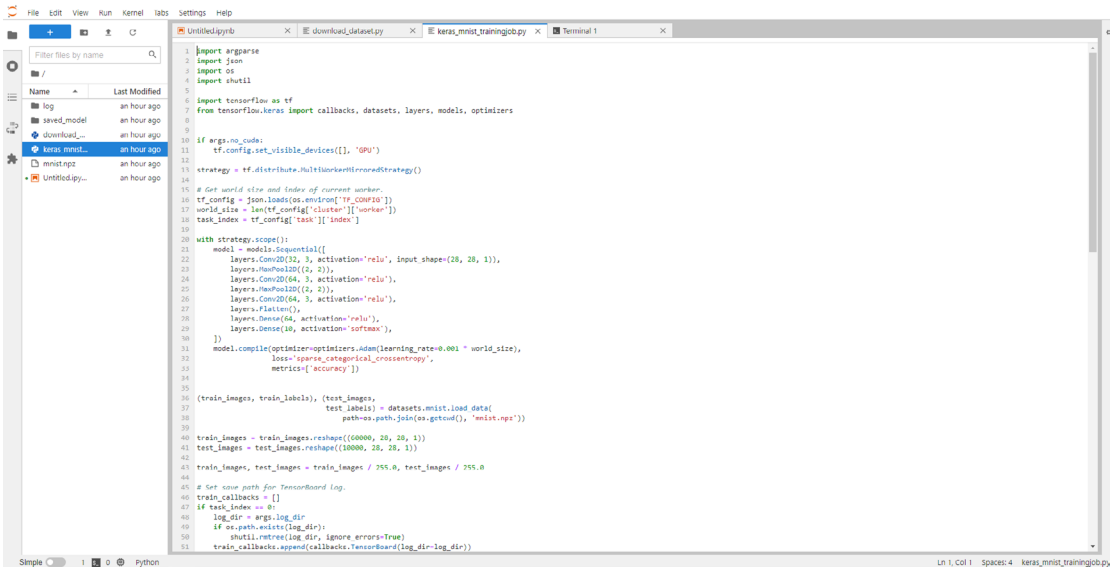

图 20　Jupyterlab 开发环境

4.5.2　环境迁移

4.5.2.1　定义内涵

Platform Ops for AI 作为整合了 DataOps、MLOps、ModelOps 的复杂技术平台，在项目开发时仅使用一套系统无法支撑平台的稳定搭建，往往需要开发系统、集成测试系统、正式环境系统在项目生命周期中协作配合。将表、索引、并发程序、配置内容等资源从源环境迁移到另一个目标环境的过程就叫做环境迁移。

4.5.2.2 技术背景

发生迁移的环境一般包括开发环境、测试环境、生产环境[24]。

开发环境：产品的配置可以在开发环境中进行变更。对于迁移过程，开发环境一般只作为源环境。

测试环境：测试环境包含了生产环境中所有业务的应用程序和配置变更。将开发环境中的资源迁移到测试环境中测试配置变更执行。测试环境既可以作为源环境也可以作为目标环境。

生产环境：测试应用程序满足需求后，将应用程序和配置内容迁移到生产环境中。生产环境一般只作为目标环境。

在环境迁移过程中，首先需要对源环境的资源进行拷贝，因此需要结合容器的镜像存储功能。Docker registry 提供后端镜像存储服务，在迁移之前为了确认迁移后没有发生镜像丢失的现象，先获取一份 Docker registry 中的镜像列表，接着采用复制镜像或是复制 registry 存储目录等方法，完成从源环境到目标资源的迁移工作。

4.5.2.3 思路方案

和数据集的共享类似，SecXOps 也支持项目代码的共享。团队开发过程中可以随时从 gitlab 上下载或同步代码，用户正在编辑的代码也会存储在 PVC 中，挂载到一个正在运行的供用户在线编程的 pod 上。

容器的环境迁移需要通过封装镜像来实现。Docker 封装镜像有两种方式，docker commit 和 docker build。使用 docker commit 可以很方便地制作镜像，然而这种方式对于镜像的使用者是完全未知的，使用者无法获取封装时对镜像做了哪些修改。同时这种修改是永久性的，如果新的使用者使用该镜像创建容器修改之后继续使用 docker commit，将会导致整个镜像越来越臃肿。而通常的 Docker 镜像是多层存储的，每一层都是在前一层的基础上进行的修改，容器以镜像为基础，同样是多层存储，因此，如果可以把每一层操作的命令都写入一个脚本，用这个脚本来制作镜像，上述问题将不再存在。这种方式即是使用 Dockerfile 进行 docker build。Dockerfile 是一个用来构建镜像的文本文件，文本内容包含了构建镜像所需的指令和说明。一个简单的 Dockerfile 示例如下图所示，通过类似的文件进行 build 即可将环境进行复制。

```
from ubuntu
RUN apt update
RUN apt upgrade
RUN pip install tensorflow
RUN pip install nltk
```

图 21　Dockerfile 示例

用户在 SecXOps 平台上所开发的算法按照用户预先设定好的类型标签可以分为数据预处理、特征提取、模型训练、超参调优等多种类型。当某个项目所使用到的算法性能优越时，用户可以选择将其公开展示，此时 SecXOps 会检测对应的项目版本内是否有 Dockerfile，如果有，则会在当前 Node 上根据该 Dockerfile 创建一个镜像，镜像内只包含用户的开发环境，数据和代码会分别创建一个 Job，复制出一个新的 PVC。镜像创建成功后，SecXOps 会观测当前项目所在的 Node 的资源是否充足，当 CPU、内存资源充足时，优先选择在该 Node 节点上按照封装好的镜像创建一个容器，供用户进行开发，当 CPU、内存资源不充足或需要使用 GPU 资源时，SecXOps 会将该镜像 push 到专为 SecXOps 平台搭建的 Harbor 服务中，随后按照一定的策略挑选一个 Node 节点，将镜像从 Harbor 中拉取下来并创建 Pod。该 Pod 在启动时挂载数据和代码对应的 PVC，实现一键式的数据、代码以及开发环境的环境迁移。整体流程如下图所示。

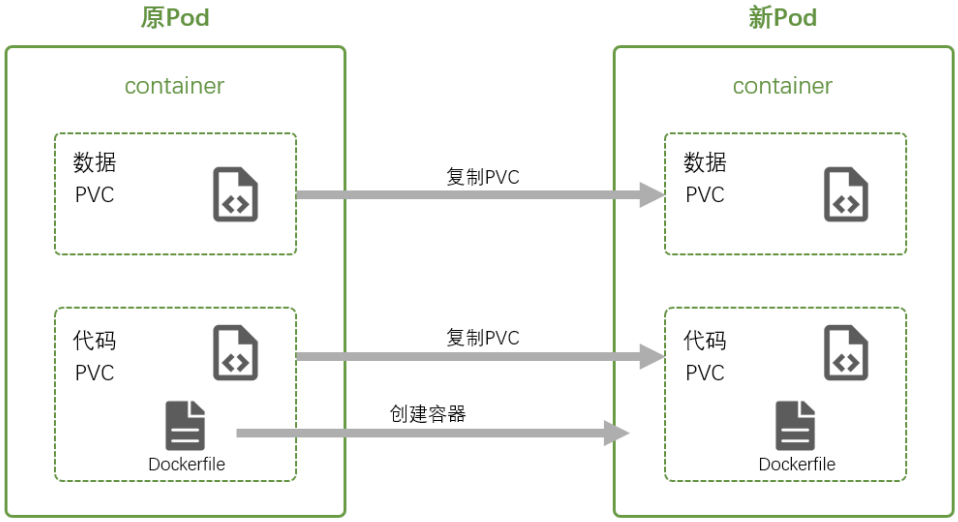

图 22　环境迁移

Harbor 是一个开源的 docker registry，通过指定策略和基于角色的访问控制来保护相关组件，确保镜像没有漏洞且签名可信。Harbor 是一个 CNCF 毕业项目，它提供了合规性、性能和互操作性，能够在 Kubernetes 和 Docker 等云原生计算平台上一致、安全地管理工件。另外，Harbor 拥有用户交互页面，允许管理员在 UI 页面直接对镜像进行查看、删除等操作。SecXOps 部分镜像列表如下图所示。

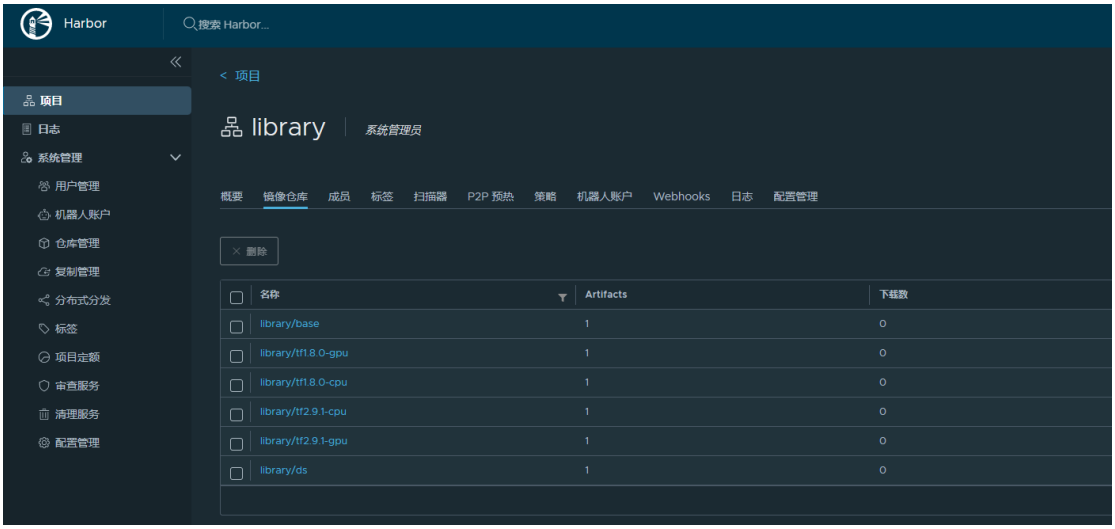

图 23　镜像列表

4.6　模型资源动态调度管理

资源动态调度（Dynamic Resource Scheduler，DRS）旨在根据智能负载均衡算法（专家系统、神经网络、遗传算法等）周期性检查集群内主机的硬件负载（包括 CPU、内存等）的情况，在不同主机间实现虚拟机的迁移，达到集群内主机间负载均衡目的，维护系统稳定性。与静态调度不同，动态资源调度在运行前不设定固定的调度映射规则，而是在运行中依据在线状态动态求解可行的任务分发方案，具备高灵活性、高效性和实时性，更适用于在拥有海量数据、众多计算任务的业务中快速响应用户请求。

4.6.1　资源分配

4.6.1.1　定义内涵

计算资源分配是指为进程分配 CPU、内存、GPU 等资源，确保程序正常运行的同时，尽可能为后续可能创建的新进程预留足够的资源。

4.6.1.2 技术背景

Kubernetes 作为最泛用的容器编排调度系统，提供默认的调度和驱逐机制。

Kubernetes 中的 Pod 将一个或多个容器作为一个单元进行管理，在应用中往往不单独部署容器，而是对 Pod 进行部署和操作。Pod 是 Kubernetes 的最小调度单位，选定需要调度的 Pod 后，根据 Pod 内的容器所需的资源和节点的可用资源，将 Pod 调度到合适的节点上。

Kubernetes 管理的集群中，每一个计算节点都包含了一定的资源。客户端提交 Pod 对象创建请求后，调度器 kube-scheduler 从集群中选择最佳的节点创建和运行 Pod，该过程细分为节点预选、节点优选和节点选定，逐步进行节点的检查过滤、节点打分和最终的节点选择。Kube-scheduler 使用 Predicate 预选策略进行节点的过滤，使用 Priority 调度算法为节点进行打分。

在进行资源分配时，如果出现资源紧张的状况，先隔离资源不足的节点，接着依照优先级，回收优先级低的 Pod 所占用的资源。此外，Kubernetes 提供优先级和抢占机制来处理 Pod 调度失败的情况。

Kubernetes 的容器动态调度策略基于扩散思想，首先依据 Node 节点的衡量指标，度量 Node 节点性能情况，建立高负载队列和低负载队列，根据系统的实施运行情况，不断调整不同 Node 节点之间的容器对象，将高负载队列的 Pod 迁移到低负载队列中，实现资源的均衡利用。

4.6.1.3 思路方案

SecXOps 的计算资源分配主要应用在开发时创建的 ReplicaSet 和部署服务时创建的一个个 Job 上。对于开发时创建的 ReplicaSet，用户可以设置明确的资源上限，如内存上限限制为 1Gi 等，同时，和部署服务时创建的 Job 一样，SecXOps 平台会根据任务类型将 Pod 分配到配有不同硬件的 Node 节点上，如将 GPU 密集型任务分配到 GPU 资源充沛的节点上等。

另外，SecXOps 配备了 Kubernetes 的 dashboard 工具，如下图所示。管理员可以通过可视化图表查看集群状态，单个 Pod 的 CPU 和内存的使用率等资源利用信息，对故障的 Pod 进行快速定位和处理，保障集群内任务的正常运行。

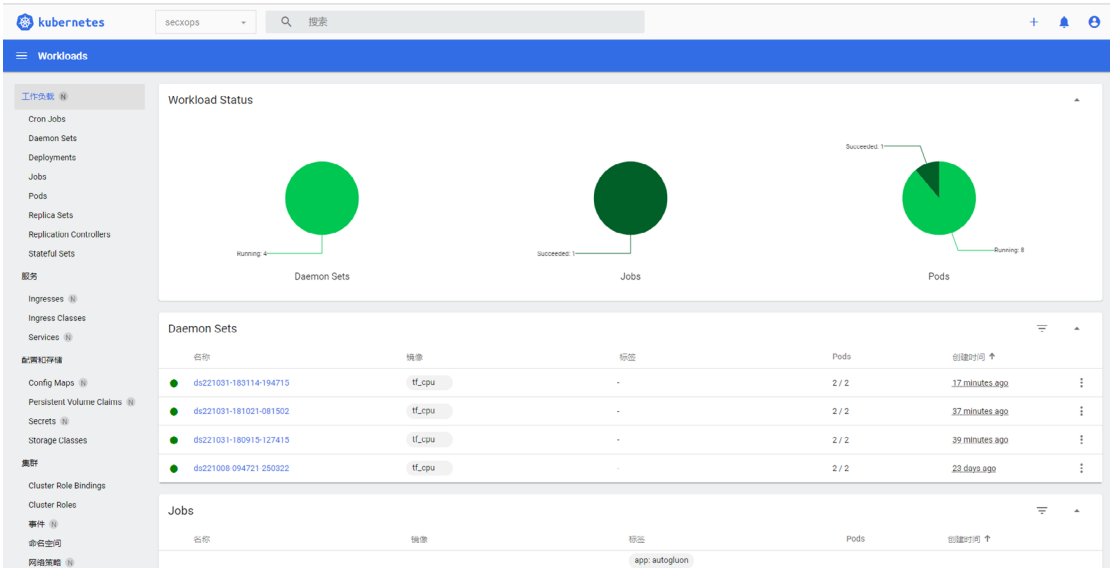

图 24　dashboard

4.6.2　负载均衡

4.6.2.1　定义内涵

负载均衡就是将任务分摊到多个操作单元上进行执行，例如 Web 服务器、FTP 服务器等，使多台设备协调配合，共同完成工作任务。

4.6.2.2　技术背景

负载均衡从概念层面上理解，是一种数据转发技术，它为一组服务器绑定到一个虚拟 IP 地址，并把这个 IP 地址和后端的真实服务器产生关联，当用户访问某个虚拟 IP 地址时，负载均衡器通过一定策略将请求转发给真实服务器。

负载均衡的技术实现方式多种多样，从服务提供方来看，网络中的数据包的结构和内容是基本一致的，通过改写网络数据包源目标地址能够将请求转发到多个上游服务器中，实现负载均衡。具体来说，可以从网络协议的角度上来分，有多种不同的实现方案，例如可以基于网络层协议实现负载均衡，也可以基于传输层协议、链路层协议以及应用层协议实现。另外，负载均衡组件根据不同的目的及不同的应用场景，有不同的部署模式，如集中式负载均衡、进程内负载均衡、独立进程负载均衡等。负载均衡策略也非常丰富，如随机策略、轮询、最小连接法、一致性哈希策略、基于权重的一些策略等等。

除了分散并发请求和提供应用横向扩展之外，负载均衡的另一个重要作用是保障系统的高可用，在 SecXOps 中，用户创建的每一个项目、每一个任务都会启动一个 Pod，显然不能将其简单地部署在某一台物理机上。尤其是对于分布式系统而言，应用假设系统必然出现故障，为了确保整体的高可用性，我们需要构建一个当局部失效时不会影响整体功能的系统，负载均衡技术就提供了这种能力。因此，冗余部署和故障审查也是负载均衡所要考虑的。

主流的负载均衡软件如 LVS、Nginx、HAproxy 在功能性上有不同的侧重点，在 SecXOps 中，我们直接使用了 Kubernetes 的负载均衡策略。

4.6.2.3 思路方案

Kubernetes 的负载均衡配置非常简单，只需要在创建服务时定义一个 LoadBalancer 类型的 Service 即可。以创建项目为例，用户填写完成表单之后，SecXOps 会为该项目创建一个 ReplicaSet，用于管理对应的 Pod，同时创建一个 LoadBalancer 类型的 Service，用于将服务发布出来。

针对不同的基础架构云平台，Kubernetes Cloud Manager 包含了能够支持多种供应商 API 的 Service Controller，Service Controller 相当于 Kubernetes 集群与外部云平台之间的接口控制器，其主要职责在于监控 Service 和 Endpoint，如果捕获到 Service 创建的时间，则调用负载均衡 API 完成负载均衡调度、虚拟 IP 分配、负载均衡器配置（包括 LoadBalancer VIP 创建、LoadBalancer Pool 创建及 LoadBalancer Member 挂载）[25]。以创建项目为例，用户填写完成表单之后，SecXOps 会为该项目创建一个 ReplicaSet，用于管理对应的 Pod，同时创建一个 LoadBalancer 类型的 Service，用于将服务发布出来。

当用户访问该服务的虚拟 IP 地址时，请求将会被路由到负载均衡器，负载均衡器查询有效的上游地址后将请求转发到一台 Node 节点上与该服务对应的 nodeport 中，再经由节点的 iptables/ipvs 规则转发到目标 Pod 中，此时用户就可以成功访问到服务内容。具体的负载均衡过程在此不做详细展开。

05

SecXOps
典型应用场景

5.1 加密流量检测的数据闭环

5.1.1 需求场景

随着人们网络安全意识的普遍提高和加密技术的不断发展，在安全与隐私保护需求的驱动下，网络中的加密流量呈现井喷式增长。特别是伴随着加密协议的不断演进，网络通信流量的加密化已经成为一个不可阻挡的趋势，引领我们走向全加密时代。

然而，伴随着加密流量的使用越来越广泛，加密在保护用户隐私的同时也给网络安全带来新的隐患。攻击者可以通过加密技术隐藏自己的攻击行为，传统的流量分析技术难以对其进行识别。从个人与企业的角度来看，网络全面化加密的滥用给个人与企业的财产安全造成了危害。勒索软件、钓鱼攻击、数据泄露等威胁可以采用加密来规避检测，比如特斯拉、波音、SpaceX 供应商拒付赎金遭机密泄露，富士康被索要巨额赎金，微软泄露 2.5 亿条记录。从社会的角度来看，网络全面化加密的滥用给网络安全造成了威胁，影响社会稳定。比如暗网中可能存在买卖公民身份信息、数字货币的非法交易，各个社交平台上网络谣言的散布、网络诈骗给人民群众带来的财产损失等。

加密流量检测的研究一直是学术界和工业界都非常关注的技术方向，分析的对象即识别过程中的输入形式，包括数据包级、数据流级、主机级、会话级等，流量识别的目标也是多样的，包括加密与非加密流量识别、加密协议识别、服务识别、加密应用软件识别，还有更细粒度的加密流量内容参数识别、网站指纹识别、异常加密流量识别等。以破密方法应对加密流量检测的思路已经越来越难以为继，当前的研究思路以非破密手段为主，在不解密的前提下实现加密流量的检测与识别，且部分方法仍然使用了加密流量中尚存的明文部分，比如加密协议握手过程中传输的明文信息。

然而加密流量检测在实际应用中困难重重，在数据的获取与分析层面尤为突出。具体表现在：

1. 特征信息不足。流量全加密时代的来临导致明文信息不足，载荷不能作为识别加密流量的特征，包长序列、包到达时间等特征也不足以区分不同的加密流量，可用特征维度显著减少，高辨别力特征更加稀有，所以维持和提升加密流量识别性能的瓶颈是分类特征的信息量不足，而非识别算法。我们需要挖掘隐藏的特征属性、增加分类特征，进而给识别任务带来增量信息。

2. 概念漂移问题。随着网络安全攻防对抗的愈演愈烈，识别目标会不断迭代、优化、升级、甚至发生改变，加密流量的特征也会随之发生变化，这些概念漂移问题使得之前训练

好的模型的准确率等性能可能会逐渐下降。可能的解决思路是对模型的结构进行调整以适应概念漂移，比如加深层、加宽层、根据数据分布变化复合新旧模型等。

3. 标注样本缺乏。传统的机器学习方法依赖于大量标注好的样本，不仅需要大量人力导致标注成本极高，也可能有侵犯用户隐私的风险，而且新的识别目标在出现早期都是小样本或者零样本的，不再适应这种新场景下的机器学习要求。我们需要研究如何减少对标注数据的需求，可以考虑小样本学习、主动学习、半监督学习、无监督学习等方法。

随着 TLS 加密技术在互联网上的普及，越来越多的恶意软件也使用 TLS 加密方式传输数据。如何通过不解密的方式直接从大体量的加密流量中检测出恶意流量，是学术界和工业界一直非常关注的问题，且已经取得了一些研究成果，但大多都是使用单一模型或多个弱监督模型集成学习的方法。与常规的单分类器检测方法不同，本节使用一种使用多模型共同决策的方法，能够在加密恶意流量的检测问题上表现出优异的性能，总体思路是利用不同异构特征训练多个不同的分类器，然后使用其检测结果进行投票从而产生最终的判定结果 [26]。

5.1.2 应用方案

在数据获取方面，通过 SecXOps 平台上现有的公开数据库中挑选合适的数据集，在 4.1 节中，我们可以看到 SecXOps 平台上已经拥有了大量公开数据集。这些数据集包含平台用户从公开数据集网站上获取到的数据、在现实环境中采集到的数据、还包含按照一定策略预处理后的或新生成的数据。针对本场景，如下图所示。

图 25　加密流量数据

从平台数据集上获取到 2022 加密流量数据，由于本地存在一些新的数据文件，因此在将数据集复制到个人空间后，可以点击数据集选择 filebrowser 进入容器中，如下图所示。

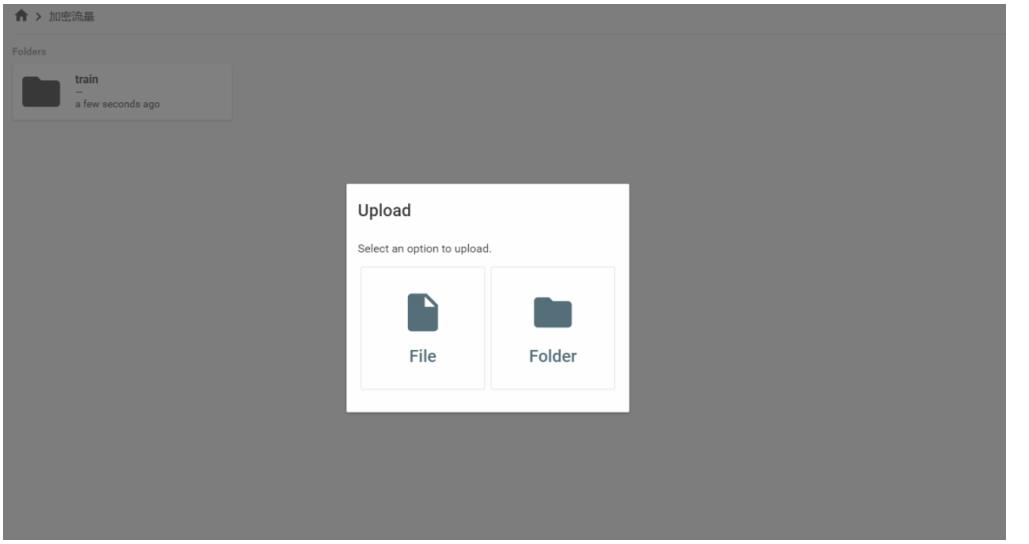

图 26　Filebrowser

Filebrowser 中可以选择上传文件或文件夹，将数据文件从本地上传到数据集对应位置中之后，即可退出数据集，进行下一步数据预处理。若想要一些个性化的数据处理，可以选择使用 jupyterlab 打开，在其内可以编写代码或脚本，对数据进行操作。

在数据预处理部分，本案例选择从数据包级、流级和主机级三个不同层次分别提取行为特征构建多个模型来提升对黑白样本的识别能力，一部分模型使用多维特征进行综合分析，还有一部分模型使用黑白样本区分度较大且置信度较高的单维特征缓解多维特征中潜在的过拟合和误报问题。

从网络流量中抽取各层次的特征信息的最终目的是为了获取网络行为的本质特征，从而提高攻击行为检测的性能。通常每一个尺度的识别过程需要提取不同层次的特征，但是流量行为记录内的特征来源广、并具有异质性和层次性，直接用于攻击行为识别会影响到模型的性能。此外，同时行为识别要拥有自适应的学习能力，即在海量数据的背景下，既能降低原始数据集的维度，又能保留数据的关键重要特征。因此，目标是寻找一种能够提供可靠的无监督特征学习的技术，将大量高维、非线性的无标签原始数据进行特征融合并降维，改进现有技术的性能和准确性。单一尺度下的特征融合就是解决在同一粒度下不同层次特征的融合问题。事实上，加密流量攻击行为识别是一个典型的分类问题，它的任务是时刻关注网络行

为通过模型判断是否向网络系统管理员发出报警信息。通过跨尺度特征融合策略进一步提取和集成这些信息，可以更有效地进行攻击行为识别任务。

针对不同尺度下的目标可以训练不同的分类器，单分类器在识别时会出现偏向性的问题，需要一种自适应多分类器融合方法进行反向链接行为识别 [27]。

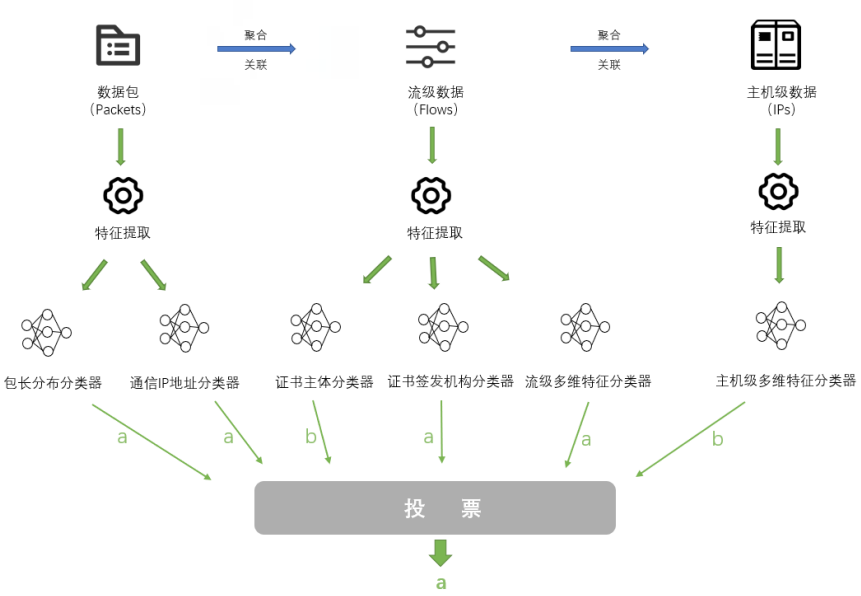

图 27　加密流量分析流程图

如上图所示，在 SecXOps 中，通过挑选合适的数据处理方法，将原始数据包转化成包级特征，通过关联整合形成流级数据、主机级数据，随后再进行一定的特征提取形成多维度的特征数据文件。

由于提取的多种特征结构是异构的，且具有不同的组织特点，所以并不适合将所有特征统一编码输入到单个模型中进行学习。该方法针对各类特征的特点分别构建合适的分类器，最后根据每个分类器的结果进行投票，以少数服从多数的原则即可产生最终的检测结果。对于模型训练后的集成工作，可以选择在 SecXOps 的模型服务处进行模型的集成判断，也可以在后续的工作流服务中进行，在本章节不进行深入介绍。

5.1.3　应用小结

将机器学习应用于网络流量分析任务的性能，除了取决于模型本身之外，数据的适当表示和特征的选择同样重要。SecXOps 通过封装一种自动网络流量分析的方法，将数据包进行

统一表示，并将其转化为适合表示学习和模型训练的格式，能够快速完成初步的数据预处理，而且这种方法不仅适用于常见的网络流量分析任务，而且表现出比其他方法搭建的模型更好的性能。

5.2 Web 安全分析模型的持续优化

基于 Web 环境的互联网应用越来越广泛，形形色色的 Web 应用也越来越多地嵌入人们的日常生活，不法分子为牟取非法利益，通过各种手段对 Web 服务器进行攻击，试图获取个人信息、窃取企业重要数据或篡改内容等。Web 攻击是恶意攻击者针对用户上网行为或网站服务器进行攻击的行为，常见 Web 攻击包含 SQL 注入攻击、跨站脚本攻击、本地文件包含、DDoS 攻击、CC 攻击等，多种网络攻击严重侵害个人用户利益，严重危害网站安全运营。据 Gartner 调查，信息安全攻击有 75% 都是发生在 Web 应用层，2/3 的 Web 站点都相当脆弱，易受到攻击。为应对 Web 攻击造成的危害，许多企事业单位部署 WAF 等安全防护设备来保障 Web 业务应用的正常运营。

传统 WAF 防护设备通过配置规则，来应对攻击者层出不穷的攻击手段，面对业务场景及攻击方式的变化，需要不断调整规则来应对攻击者的挑战。然而仅靠规则引擎及其他语义引擎难以完全抵御 Web 攻击。随着人工智能技术的不断发展，结合人工智能技术提升 Web 应用防火墙的防御能力，成为各企业的共识。多家安全厂商将 AI 安全分析引擎与规则引擎、语义引擎等结合在一起构筑抵御 Web 攻击的防御系统 [28]。

图 28　多分析引擎联合防御

5.2.1 需求场景

为保障企事业单位网站的业务正常运行，加持 AI 能力的 WAF 等安全防护设备广泛应用在运营商、高校、医疗、金融、政府、能源、交通等领域，为 Web 应用安全保驾护航。据统计每隔三到五年，网络流量数据成分就会发生显著变化，同时，随着生产环境中新业务的不断上线，业务场景的不断变化，使得基于当下数据建立的分析引擎，随着时间演变将产生性能下降的问题。部署后的 WAF 设备不断需要运营人员根据防护站点情况及 WAF 设备检测分析结果，对其内置规则引擎的规则进行调整，减少误报及漏报情况，提高防护精度，避免对防护站点正常业务产生影响。正如规则引擎需要不断调整规则一样，Web 应用防火墙中的 AI 安全分析引擎在面对新的业务场景、防护业务发生变化、数据产生漂移时，检测能力会下降，需要 AI 工程师对 AI 模型进行优化。

传统的 AI 安全分析引擎运营流程如图所示。AI 分析引擎部署之后，安全运营人员对 AI 分析引擎的告警数据进行分析，当告警数据中出现大量正常业务数据，影响 Web 站点正常运行时，安全运营人员将收集误报数据并反馈给安全专家，由安全专家对数据进行分析，并对数据打标签后，将数据交由 AI 分析引擎开发人员，由开发人员对 AI 分析引擎的误报漏报情况进行优化，然后发布优化后的 AI 分析引擎，对设备中的分析引擎进行迭代更新。可以看到如同传统规则库的维护一样，AI 安全分析引擎的运营维护流程同样存在繁琐耗时的问题。

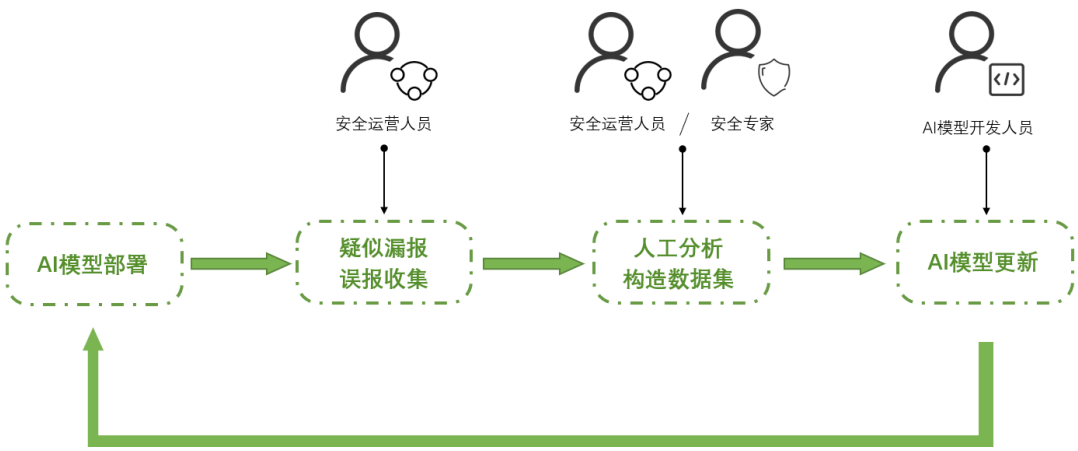

图 29　传统 AI 安全分析引擎运营流程

5.2.2 应用方案

不同于规则引擎，AI 模型本身具备自主学习和自我更新的能力，AI 模型能够根据当前数据的变化，不断学习调整模型参数，实现自我更新进化。

为了减少 AI 安全分析引擎运营过程中的繁琐流程，实现分析引擎的自动化运营，提高运营效率，SecXOps 平台打通 AI 安全分析引擎的开发、测试、部署及后续运营流程，尤其对于 AI 安全分析引擎的运营维护，SecXOps 平台基于 AI 模型的自主学习特性，通过采用 AI 模型增量学习及重新训练等措施，实行 AI 模型自动迭代升级，形成 AI 安全分析引擎的运营闭环。既解决现有 AI 模型性能退化问题，也减少 AI 模型优化升级过程中繁琐的人力沟通过程，降低人力沟通成本，提升 AI 安全分析引擎的运营效率。AI 安全分析引擎运营的具体应用方案如下图所示。

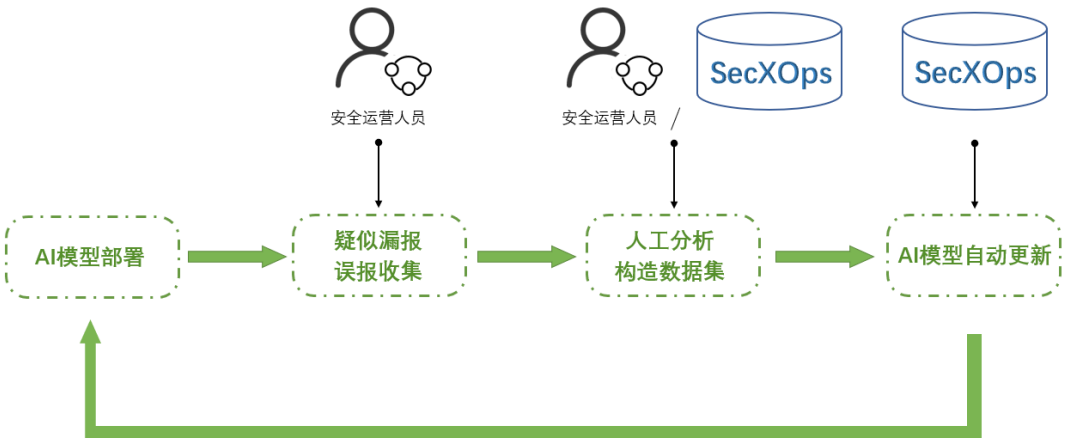

图 30　SecXOps AI 安全分析引擎运营流程

1. AI 安全分析引擎开发

SecXOps 平台提供 AI 安全分析引擎的开发环境，AI 安全分析引擎的开发遵循一般应用中 AI 模型的开发流程。针对常见的 SQL 注入、XSS 攻击、远程命令执行等 Web 攻击的检测，属于监督学习任务，AI 安全分析引擎开发人员针对分析引擎的实际业务场景，收集需要检测的 Web 攻击数据以及业务场景中的正常业务数据，构建丰富的数据集，完成数据集清洗、数据集打标签等前期工作。依据选择 AI 模型的不同，进行适配的数据预处理进行特征提取，如原始数据统计特征提取、利用神经网络进行数据高维特征提取、利用注意力机制提取数据注

意力分布等。基于提取的数据特征，构建初始 AI 模型，并进行 AI 模型训练，对于收敛的 AI 模型，开发人员采集测试数据，利用 SecXOps 提供的测试功能进行 AI 模型的性能测试，生成鲁棒的 AI 模型。针对未知类型 Web 攻击的检测，常采用自编码器、聚类等算法，对于聚类算法，前期不需要对 Web 攻击数据集打标签，利用 SecXOps 平台训练聚类模型并进行模型测试，生成鲁棒的聚类模型，实现未知类别 Web 攻击的检测。

2. AI 安全分析引擎部署

SecXOps 平台提供模型发布功能，模型开发者将训练好的成熟 AI 模型及其数据预处理模块、模型推理服务模块共同打包发布，WAF 等设备的开发人员将发布的 AI 安全分析引擎嵌入到 WAF 等 Web 安全防护设备中，与其他规则引擎、语义引擎融合，形成联合防御系统，降低 WAF 等设备的误报及漏报情况，提升防护设备抵御各种类型 Web 攻击的能力。

目前的 AI 安全分析引擎与规则引擎、语义引擎的融合有多种方式，目前常用的融合方式为串接融合。串接融合将 AI 安全分析引擎、规则引擎、语义引擎以串联的方式进行部署，每个分析引擎的部署位置由各分析引擎的防护重点及实际业务场景的特点决定，后接的分析引擎对前一个分析引擎的告警结果进行分析，在尽可能低耗时的情况下，保证整个联合防御系统对于 Web 攻击的防御能力。除了串接融合，分析引擎还可采用并接融合的方式，流量数据将分别经过三个分析引擎进行检测，经过多分析引擎的共同研判，确定该条数据为正常业务数据还是 Web 攻击数据，并接融合可以提高 WAF 设备检测准确率，但会增加时间开销。

3. AI 安全分析引擎运营

对于嵌入 WAF 等安全防护设备的 AI 安全分析引擎，需要安全运营人员对其进行持续的运营维护。SecXOps 平台提供 AI 模型参数自动更新功能，当 AI 安全分析引擎在生产环境中出现大量正常业务误报，影响 Web 站点正常使用，或者出现大量 Web 攻击数据漏报情况，检测能力下降时，安全运营人员通过 AI 模型参数自动更新功能进行 AI 安全分析引擎的优化升级。在生产环境中安全运营人员采集疑似 Web 攻击误报漏报的数据，利用 SecXOps 内置的聚类模型进行数据的初步打标签，并通过专家知识的辅助对聚类结果进行分析验证、清洗数据、生成数据的准确标签，以及构建数据集。基于构建的数据集与 AI 模型，在 SecXOps 平台创建 AI 模型参数更新任务，配置数据集、模型及运行参数，即可进行 AI 模型的参数更新。对于训练表现好的模型，需要通过模型测试才能进行模型发布。模型测试目前有两种方式：第一种，在生产环境中采集数据，利用 SecXOps 平台的模型测试功能进行测试；第二种，对于训练好的模型，在生产环境中进行旁路部署，在不影响业务环境的情况下进行模型检测

能力的测试。对于通过模型测试的 AI 模型进行模型发布，并对设备中的原 AI 模型进行替换，完成 AI 安全分析引擎的迭代升级。

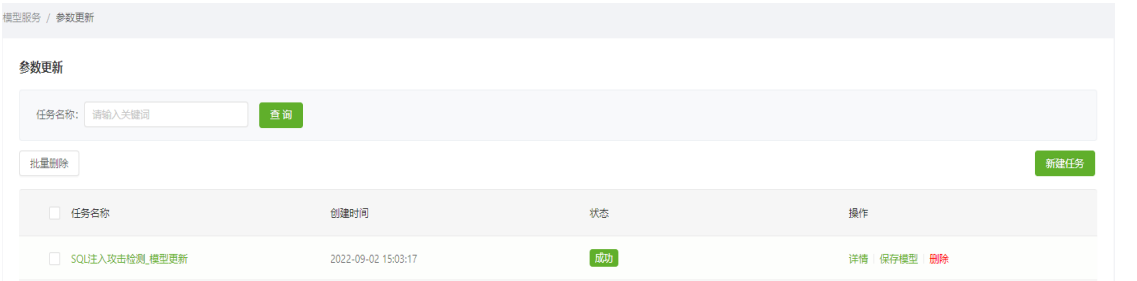

图 31　创建超参数搜索任务

4. AI 安全分析引擎超参数调优

当生产环境新业务上线，数据分布发生较大变化时，安全运营人员仅通过 AI 模型参数更新功能可能无法解决 AI 安全分析引擎检测能力下降的问题，此时，安全运营人员可以使用 SecXOps 平台的超参数调优功能进行 AI 模型更新。安全运营人员构建生产环境的真实数据集，在 SecXOps 平台创建模型超参数调优任务，基于该数据集，寻找模型的最佳参数，生成在该数据集上表现最好的模型，该 AI 模型通过后续的模型测试后，将达到检测性能要求的 AI 模型进行发布，并部署到设备中对原 AI 安全分析引擎进行迭代升级。

5.2.3　应用小结

AI 安全分析引擎部署后，运营过程中面对一系列繁琐的流程，SecXOps 平台通过内置 AI 模型自动参数更新以及模型超参数调优功能，可以解决当 AI 模型性能下降时，需要进行模型更新的问题，并在整个 AI 模型迭代更新过程中尽可能的实现自动化运营，减少人力沟通过程，提高 AI 安全分析引擎运营效率。同时，部分安全场景的数据可以通过 AI 标注功能，实现数据的初步打标签工作，辅助安全运营人员构建 AI 模型更新需要的数据集，进一步减少 AI 安全分析引擎运营过程中的人力工作，使安全运营工作进一步朝着自动化、智能化的方向发展。

5.3　工控协议识别算法自动化调参

随着工业 4.0 的到来，工业控制系统越来越广泛的应用于诸如电力、石油、化工、天然气、交通运输、医疗、物联网等领域，在生产工作中发挥了重要作用，促进工业向自动化、智能化方向发展，在关乎国计民生的领域发挥着越来越重要的作用。然而随着数字化、网络化、智能化的工业控制系统逐渐接入互联网，针对工业控制系统的各种网络攻击事件日益增多，

病毒、木马、蠕虫、僵尸网络等也威胁到工业控制系统的安全，勒索病毒的出现，在企业损失大量数据的情形下，也对企业造成不可估量的经济损失。尤其针对重要工业基础设施的攻击，给工业生产造成重大影响。

5.3.1 需求场景

安全防护设备对工控网络进行安全防护，需要对其中的工业控制协议进行深度解析，分析出数据包中的操作指令及收发双方，从而将异常报文及异常业务逻辑操作交由工控安全业务处理模块进行安全规则校验，以达到入侵检测及防护工控设备的目的。

通过对工控协议的深度解析可以从指令级和语义级对潜在的网络攻击进行检测、感知和防护。因此为了监测和保护工业设备，识别它们之间的工控协议成为重要前提。协议识别的准确率、覆盖率和处理效率直接决定了网络安全设备功能的完备性及处理效率。

目前工控协议的识别主要有以下几种方法：

1. 基于端口号的工控协议识别，该方法根据应用层协议公开在 IANA 中注册的端口号来识别协议，该方法可以识别大部分的已知协议，但无法识别使用动态端口及其他层的协议 [29]。

2. 基于指纹的工控协议识别，利用协议的指纹特征库对网络数据包的内容进行特征匹配，可以将数据包载荷的前 N 个字节或者数据包的完整载荷或者自定义特征与协议特征库中的特征进行匹配，以进行协议识别。

3. 基于协议逆向的工控协议识别，该方法分为两大类：基于执行轨迹的逆向识别和基于网络轨迹的逆向识别。通过对工控协议进行字节流拆分、关键词提取、关系分析、特征提取以及对报文结构的分析来进行工控协议的识别。

4. 基于深度流及行为特征检测的工控协议识别 [30]，不同的协议用于数据传输时，其数据流的特征也会根据协议行为的不同而有所差异，同时，其会话连接状态也会有所不同。该方法对数据流特征及会话行为特征进行归类统计，并根据统计特征的不同利用分类器模型来进行工控协议的识别。对于未知的工控协议，也可采用聚类模型来进行发现 [31]。

基于传统的工控协议识别方法，存在识别种类少、识别准确率较低的问题，而且对于协议逆向需要大量的安全背景知识。随着人工智能技术的不断发展，将 AI 技术应用于工控安全领域，助力于工控协议的识别，保障工控安全，越来越成为各安全厂商的共识。

5.3.2 应用方案

SecXOps 平台通过嵌入 NNI 及 Autogluon 框架，提供 AI 模型超参数调优、神经网络架构搜索、自动模型生成功能，本节以 AI 模型对工控协议的识别为例，来介绍 SecXOps 的上述功能如何帮助 AI 模型开发，助力于工控协议的识别。

1. 超参数调优

对于工控流量数据，首先利用 SecXOps 平台提供的数据处理模块进行数据清洗、数据解析，然后提取数据包级特征，并将数据包聚合为数据流，提取数据流级特征，构造用于工控协议识别的特征集。

图 32 创建超参数搜索任务

开发者进行超参数调优任务，首先在 SecXOps 平台创建超参数搜索任务，然后编写工程代码，以 CNN 模型为例，开发者搭建基本卷积神经网络结构，设定网络参数初始化方式，实现从数据输入到模型训练的流程。然后配置参数搜索空间的 json 文件，需要设定搜索的模型超参数，如图所示，设定了卷积核尺寸、动量、dropout、批处理大小、学习率四个超参数，并设定每个超参数的搜索空间。

```
{
    "kernel_size": {"_type":"choice", "_value":[3,, 5, 7],
    "momentum": {"_type":"uniform", "_value":[0, 1]},
    "dropout_rate": {"_type":"uniform", "_value":[0.5, 0.7, 0.9]},
    "batch_size": {"_type":"choice", "_value":[16, 32, 64, 128, 256]},
    "learning_rate": {"_type":"choice", "_value":[0.0001, 0.001, 0.01, 0.1]}
}
```

图 33 超参数搜索空间

然后进行模型实验的配置，编写相应的 yaml 配置文件。启动一个超参数搜索任务，模型实验结束后，将返回每种超参数组合下的训练指标，开发者可根据实验结果得到在当前网络架构下的最佳超参数。

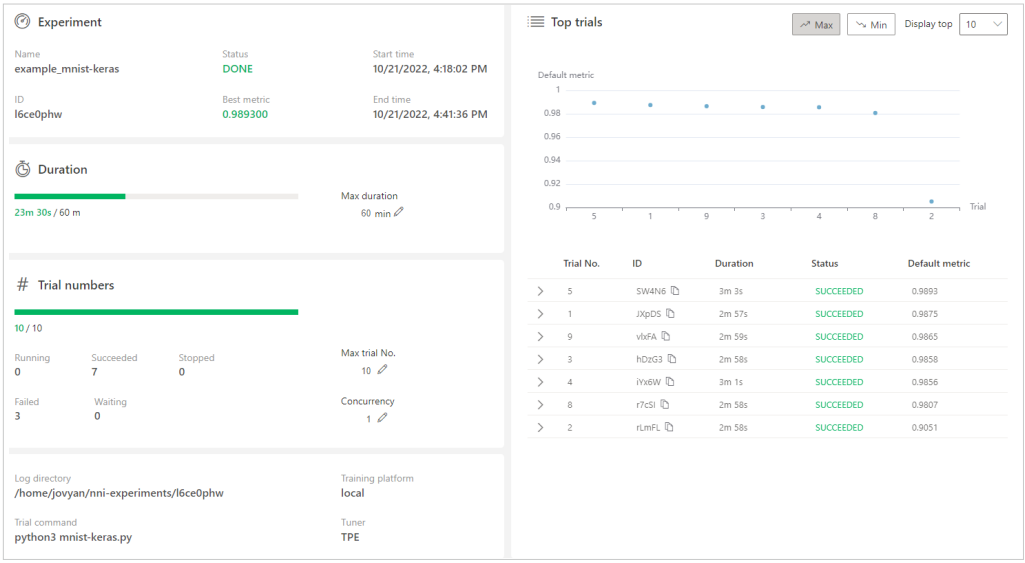

图 34　超参数搜索结果

2. 神经网络架构搜索

基于上面构建的工控流量数据的特征集，进行工控协议识别的模型架构搜索任务。同样先在 SecXOps 平台创建网络架构搜索任务。基于任务场景，编写工控协议识别的工程代码，编写神经网络的基本架构，完成数据输入到模型训练的完整流程。首次进行网络架构搜索时，通过运行命令“nnictl ss_gen”生成搜索空间 json 文件，并配置具体的网络架构搜索范围。然后设置实验相关的 yaml 文件，对整个网络架构搜索实验进行配置，配置成功后，即可运行实验。

```
{
    "LayerChoice_1": {
        "_type": "Layer_choice",
        "_value": [
            "0",
            "1",
            "2"
        ]
    },
    "LayerChoice_2": {
        "_type": "Layer_choice",
        "_value": [...]
    },
    "LayerChoice_3": {
        "_type": "Layer_choice",
        "_value": [...]
    },
}
```

图 35　网络架构搜索空间

实验运行成功后，可以在页面查看实验结果，选择得分最高的一组实验，即可得到在该搜索空间下的最佳神经网络架构。

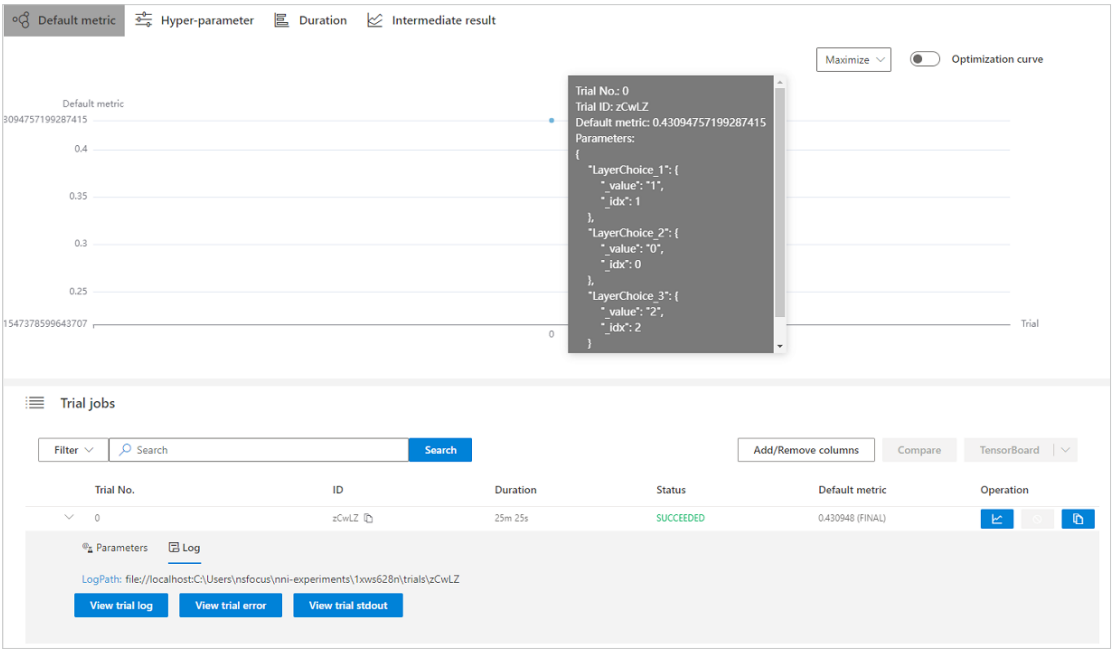

图 36　神经网络架构搜索实验结果

3. 自动模型生成

基于上面构建的工控流量数据的特征集，进行自动模型生成的任务。自动模型生成功能利用 SecXOps 平台内置的 Autogluon 框架。不同于上面两个任务，基于 Autogluon 的自动模型生成，无需预先设定模型，也无需预设模型的超参数，也不需要配置搜索空间及实验的 yaml 文件，仅需设定数据集及实验相关参数即可。

首先构建工控协议识别的工程代码，将实验数据转化为 Autogluon 中所需的格式，设定训练集、验证集、测试集，并定义训练数据的标签，设定任务类型以及训练评估指标，如准确率、召回率、log_loss 等，配置参数 presets 为“best_quality”，该参数将允许训练时间足够长，以产生更加鲁棒的模型，通过设置该参数为“good_quality”、“optimize_for_deployment”，将产生轻量的模型，减少模型的推理时间。设定模型测试模式，保存最优模型，并配置 leaderboard，用于查看实验中各模型的训练指标。出于时间成本的考虑，可以设置实验提前终止的条件。配置完实验相关参数后即可进行模型训练，得到最优模型后，可以将表现较好的模型进行 Bagging 得到集成模型。自动模型生成的实验结果如下图所示。

```
Fitting model: NeuralNetFastAI ...
    0.82     = Validation score   (accuracy)
    4.25s    = Training   runtime
    0.02s    = Validation runtime
Fitting model: XGBoost ...
    0.87     = Validation score   (accuracy)
    0.45s    = Training   runtime
    0.01s    = Validation runtime
Fitting model: NeuralNetTorch ...
    0.85     = Validation score   (accuracy)
    2.18s    = Training   runtime
    0.01s    = Validation runtime
Fitting model: LightGBMLarge ...
    0.83     = Validation score   (accuracy)
    0.49s    = Training   runtime
    0.01s    = Validation runtime
Fitting model: WeightedEnsemble_L2 ...
    0.87     = Validation score   (accuracy)
    0.3s     = Training   runtime
    0.0s     = Validation runtime
AutoGluon training complete, total runtime = 12.17s ... Best model: "WeightedEnsemble_L2"
```

图 37　自动模型生成实验结果

通过 leaderboard 可以更清晰的看到模型的各项具体指标，通过下图可以看到测试最优模型为 XGBoost 及 WeightedEnsemble_L2。

In [4]: `predictor.leaderboard(test_data, silent=True)`

Out[4]:

	model	score_test	score_val	pred_time_test	pred_time_val	fit_time	pred_time_test_marginal	pred_time_val_marginal	fit_time_
0	XGBoost	0.974	0.87	0.010971	0.007979	0.381979	0.010971	0.007979	
1	WeightedEnsemble_L2	0.974	0.87	0.013963	0.007979	0.655247	0.002993	0.000000	
2	RandomForestGini	0.968	0.84	0.126632	0.106736	0.531581	0.126632	0.106736	
3	RandomForestEntr	0.966	0.83	0.134669	0.123664	0.419883	0.134669	0.123664	
4	ExtraTreesGini	0.964	0.82	0.127691	0.108711	0.413922	0.127691	0.108711	
5	ExtraTreesEntr	0.962	0.81	0.128654	0.110676	0.406938	0.128654	0.110676	
6	LightGBMLarge	0.952	0.83	0.015957	0.006982	0.462761	0.015957	0.006982	
7	NeuralNetTorch	0.952	0.85	0.042885	0.009973	2.298853	0.042885	0.009973	
8	KNeighborsDist	0.930	0.65	0.110705	0.116686	0.004985	0.110705	0.116686	
9	LightGBM	0.914	0.85	0.013962	0.008975	0.210438	0.013962	0.008975	
10	NeuralNetFastAI	0.850	0.82	0.036901	0.020944	2.043537	0.036901	0.020944	
11	LightGBMXT	0.844	0.83	0.010971	0.007979	0.988357	0.010971	0.007979	
12	KNeighborsUnif	0.766	0.73	0.111670	0.110704	0.006981	0.111670	0.110704	

图 38 leaderboard 自动模型生成实验结果

5.3.3　应用小结

SecXOps 平台通过内置 NNI 及 Autogluon 对外提供模型超参数搜索、神经网络架构搜索、自动模型生成功能，对于模型开发者提供了大量帮助，有效缩短了模型开发时间，并更易于

找到最佳模型。对于欠缺相关 AI 知识的开发者，也能利用自动模型生成功能完成模型的初步开发。

5.4 Webshell 安全检测的增量开发

5.4.1 需求场景

Webshell 是黑客经常使用的一种恶意脚本，目的是获得服务器的执行操作权限，例如执行系统命令、窃取用户数据、删除 web 页面、修改主页等，其危害不言而喻。黑客通常利用常见的漏洞，如 SQL 注入、远程文件包含 (RFI)、FTP，甚至使用跨站点脚本攻击 (XSS) 等方式作为社会工程攻击的一部分，最终达到控制网站服务器的目的。

Webshell 攻击检测技术主要有以下几种：

1. 基于流量的 webshell 检测。该方法信息采集相对较为方便，通过流量数据直接分析原始信息。基于对流量数据中携带的 payload 行为进行分析，不仅能够对已知的 webshell 行为进行检测，还能够识别出未知的伪装性更强的 webshell 攻击行为。
2. 基于日志的 webshell 检测。web 日志是 web 服务器记录用户访问行为产生的文件，标准的 web 日志是纯文本形式，每行一条记录，对应客户端浏览器对服务器资源的一次访问。通过分析日志数据，对网站的访问行为进行建模，不仅能够检测到可疑的漏洞攻击行为，还可以提取到特定时间段特定 IP 对应的访问行为，回溯整个攻击过程。

5.4.2 应用方案

本节示例中的 webshell 安全检测模型使用基于流量的 webshell 检测方法，通过构建神经网络来对 payload 进行行为分析。然而，该程序在运行时对 python 的版本、相关语料库、开源框架的版本等的依赖性较强。例如代码使用到的 tensorflow2.9.1-gpu 版本，需要配置 cuda11 相关环境，下载时间长、安装步骤多；同时使用了 nltk 中的 english 停用词语料库，由于是在国外网站下载，经常遇到下载失败的情况。种种原因导致在多人合作开发算法时，或团队中有新成员加入时，环境的配置较为繁琐，费时较多，致使团队工作效率下降。

为了解决开发环境配置困难的问题，本平台采用了 4.2 节中所提到的一系列技术，实现了老代码开发环境一键化复制。

用户在本平台上开发代码时，可以选择从零开始写代码，也可以选择从平台的公开项目库中挑选合适的代码进行增量开发。对于前者，平台上封装了多种基础镜像供用户选择，包含了 tensorflow 和 pytorch 等的常见版本。

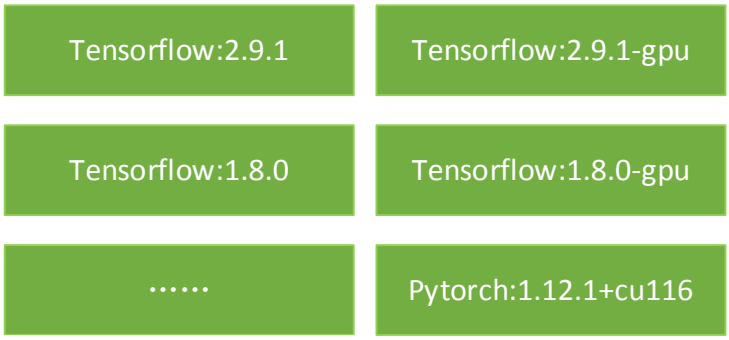

图 39　基础镜像

这些镜像中还预先配置了算法开发过程中可能用到的库，在用户创建时减少用户的下载时间。当用户将个人项目的环境配置完成并确认能够稳定使用后，通过在容器中编写 Dockerfile 文件，可以将整个项目的环境配置过程保存下来，随即可以选择将项目公开，存储在公开项目库中供其他用户使用。当用户在公开项目库中选取了心仪的项目后，只需点击拷贝即可将项目代码、数据文件和配套环境同时复制出来，使用户可以直接运行项目查看结果。

在 webshell 安全检测场景中，用户 A 可以选择 tensorflow2.9.1+gpu 镜像，创建代码编辑环境。在使用 nltk 库时，停用词语料库会自动下载到用户目录下，由于该语料库大小并不大，可以选择连同代码一起存放在 PVC 中。用户目录下的文件结构如下图所示。

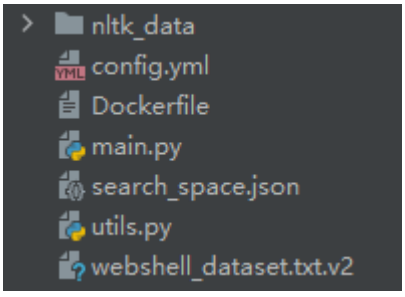

图 40　文件结构

当环境配置完成后，用户 A 需要编写 Dockerfile 文件，将配置过程保存下来。随后点击公开按钮，平台会检测项目中是否存在 Dockerfile 文件，若缺少会给用户提示。随后用户 B 即可在公开项目库中看到本项目，点击项目对应的拷贝按钮，平台会自动按照项目中的 Dockerfile 文件构建一个镜像，随后创建一个容器，将原项目所映射的 volume 作为模板，复制一份新的 volume 映射到该容器上，完成数据、代码、环境的全部复制。此时在用户 B 进入到 jupyterlab 页面后即可看到所有配置已经复制完整，可以直接进行开发或运行。

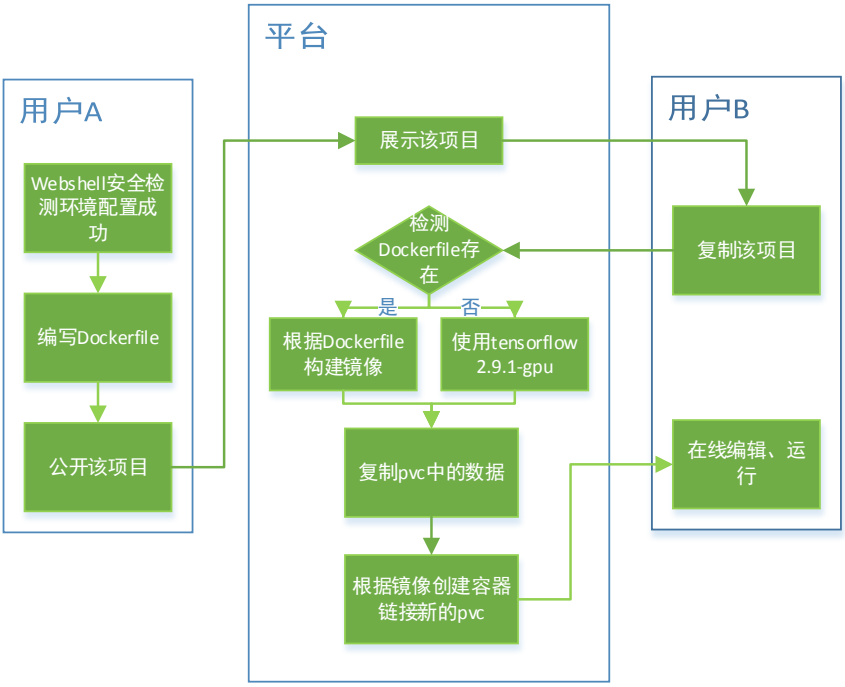

图 41　开发环境一键化复制流程

5.4.3　应用小结

在本场景中所使用到的配置相对较为简单，所使用的数据是已经经过初步处理的文本数据，因此在环境复制时只需要复制一些和数据预处理、特征提取以及算法相关的库即可。然而在一些安全场景下，数据的采集过程也是需要相对应的工具才能进行的，例如安装相关工具来采集流量等。相对于手动安装环境来说，配置时间可以减少 80% 以上，且不易出错，更快更可靠。

5.5　安全告警日志的工作流服务

随着攻防对抗的升级，在有限的资源投入下，当前安全运营中心信息过载，关键指示器数据无法得到有效的分析和处置，进而进入持续积压的恶性循环，造成告警疲劳现象。目前在告警运维中，需要应对告警疲劳，威胁失焦等问题。现阶段，基于大规模、多源数据汇聚的集中式 SOC 运营平台，已成为安全运营团队的基础工具及流程推进工具。因此，需要针对告警日志需要进行有效的分析，以提升安全运营的效率。

5.5.1 需求场景

在网络安全运营中，由于攻击产生的数据包往往会与正常活动产生的数据包较为相似，如果攻击检测规则较为敏感，会产生大量告警误报。在业务场景多样化的背景下，安全运营人员在实际生产环境的运维中需要对各种防护系统的告警进行研判，发现攻击行为，进行应急响应，保护网络安全。面向网络攻击检测/防护系统产生的原始告警日志，安全运营中心（SOC）会通过多种攻击行为聚合规则、无用告警过滤基线以及各种机器学习算法模型等方案对告警进行处理，然后基于高置信度运营规则等方案进行告警运营，但是基于“高置信规则”的方法只是将“攻防经验”这一专家知识融入到告警分析中，但实际上，影响告警最终研判结果的因素往往除了“攻防经验”外，还与客户业务、资产、网络拓扑等因素相关，单一的研判方式导致了大量的误报和漏报。在复杂的工业场景中告警数量通常也会超出运营人员的排查能力，安全运营中心需要从海量安全数据中分析出能对企业安全造成风险的事件。在这个环节中，海量数据会引发“告警疲劳”问题，现有基于高置信度规则等运营方案有存在两个问题：1、残余风险不可控、未知威胁难发现 ；2、基于高置信度规则列表的方法依然需要处理大量噪声。

5.5.2 应用方案

为了应对上述问题，企业和组织提出安全编排自动化与响应技术来应对传统方案无法应对的威胁数据增多、设备孤立技术整合度低等问题，通过工作流的编排来降低不同技术间的转换成本。相关技术已经被应用到安全告警日志分析中，针对海量告警数据进行智能分析，通过工作流的编排技术对海量告警实现分阶段、多种维度的过滤、分类、聚合、排序的操作，将人工成本降低，实现在有限人力成本下，精确率和召回率的双重提升。

告警的智能分析目标在于，对待分析的网络安全告警数据做智能分析，根据分析结果，提取攻击数据，并且根据启发式策略做事件归并，形成可运营的按照风险排序的事件风险列表。因此，为了实现智能分析需要包括多个关键技术，即降噪过滤、告警归并、威胁推荐等，将多个技术来对原始数据进行分析处理，在告警分析的工作流的指导下，利用人机结合的方法，基于面向的运营场景实现智能分析过程的定义、排序和驱动标准化的告警日志分析，具有定制化、灵活化、联动化的特点，相关分析流程如下图所示。

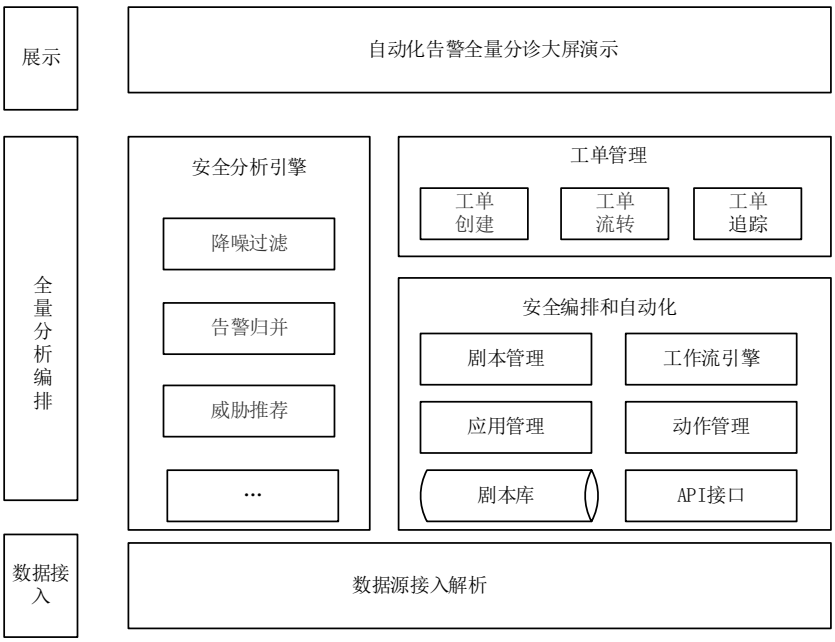

图 42　告警智能分析安全编排技术架构

为实现安全告警日志智能分析的目标，在基于安全编排的告警分析工作流运行时，安全运营人员可以随时取消或查看正在运行的任务。在基于安全编排的告警分析工作流运行过程中可能会产生不同的新资源，如告警数据处理类型的工作流会产生新的数据集，其中涉及告警数据算法模块的工作流会产生新的模型，这些新生成的资源都可以在告警日志分析工作流中进行保存，以不断完善告警数据集库、模型库和算法库。在基于安全编排的告警分析工作流中，首先将原始数据进行解析作为输入。基于告警智能分析的关键技术，对告警进行分析处理，包括：降噪过滤模块、告警归并模块、威胁推荐排序模块。其中，降噪过滤模块会通过多个模块过滤掉尽可能多的误报，例如基于 AI 的告警降噪，基于业务的告警降噪，基于载荷匹配的告警降噪；告警归并模块将剩余告警进行聚类等归并操作，旨在达成“研判少量事件便可研判整个类”的运营效果；威胁推荐模块运用推荐算法，将专家知识融入流程，对告警进行推荐。通过告警智能分析的各个模块，对告警日志进行分析，完成对告警的分诊处理，得到告警的可信度和优先级，将低质量的告警日志变成高质量有价值的告警，提升告警的质量。在 SecXOps 中，安全运营人员可以通过平台上的工作流服务，实现容器内的告警日志分析工作流的编排，如下图所示，SecXOps 支持将安全运营人员创建的上述告警日志分析模块以 DAG 图的形式进行编排，按照一定的资源动态调度策略在集群上运行，最终得到告警日志分析每一个模块的运行结果。

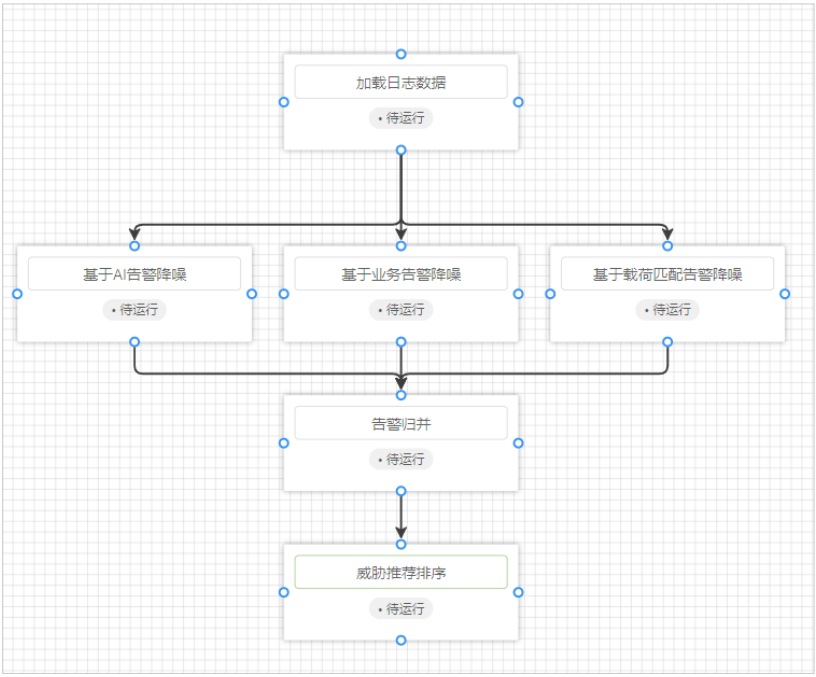

图 43　基于安全编排的告警分析工作流

综上所述，安全编排与自动化部分的系统能力是告警智能分析安全编排技术架构的核心技术，将告警分析过程中涉及的不同模块组件的安全功能通过可编程 API 接口进行封装，以完成告警智能分析的过程，使得整个过程可以通过剧本的方式进行表述和保存，以提升告警日志分析的效率。

5.5.3　应用小结

尽管大规模安全事件、情报等日志带来威胁分析的全新机遇，但大规模事件、告警分析任务，向运营人员的技能提出更高要求，不止需要较高的安全事件分析能力与攻击技战术研判能力，还需要对客户业务环境、IT 系统环境足够熟悉，才能达到较高的研判效率与精度。然而，安全经验复杂多样、企业环境动态多变，大规模告警、数据中存在错误、误报、噪声等各类问题，给运营专家的研判响应分析工作带来巨大挑战。在这个背景下，如何充分发挥数据科学，包括基础数据分析、机器学习、模式挖掘等方法的优势，加速大规模告警的自动化研判，辅助运营专家的主动狩猎，成为运营能力发挥的关键组成。基于工作流编排的安全告警智能分析技术，从开发、封装、构建一系列具备告警分析、研判的自动化组件，为运营专家和团队提供数据、知识驱动的数据科学能力支撑，以实现自动化智能化的告警数据分析，加速关键安全告警日志分析能力与流程的自动化进阶。

06

SecXOps 技术发展趋势

可以预见，安全智能分析技术的发展，将全面提升网络安全关键应用场景下威胁检测的效果，推动安全分析从基础级、领先级，向卓越级演进。根据 Gartner 2021 年十大数据和分析技术趋势，XOps 的目标是利用 DevOps 最佳实践实现效率和规模经济，在保证可靠性、可重用性和可重复性的同时，减少技术和流程的重复并实现自动化。同时在 2022 年重要战略技术趋势报告中预测，到 2025 年，10% 的企业会落实 AI 工程化最佳实践，相比于另外 90% 未采用类似实践的企业，前者的 AI 工作可以创造至少高三倍的价值。这要求企业建立并完善 AI 工程化实践，将 DataOps、ModelOps 和 DevOps 的最佳实践纳入其中。SecXOps 相关技术仍处于蓬勃发展阶段，技术演进仍存在诸多问题需要解决。在此，我们从 SecXOps 技术涉及到的几个核心 Ops（DataOps、ModelOps、DevOps 以及 PlatFormOps）展望安全智能分析发展的关键趋势。

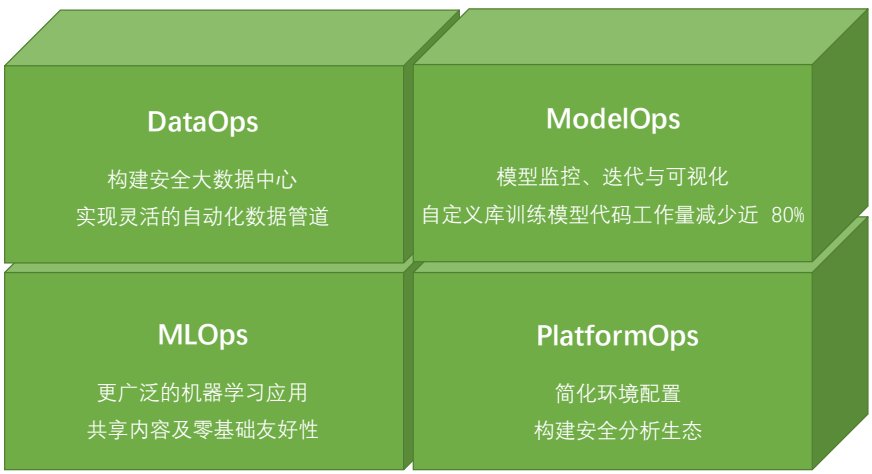

图 44　SecXOps 技术发展趋势

- DataOps 层面：构建安全大数据中心，实现灵活的自动化数据管道

人工智能算法在安全领域应用种类多样。用 AI 做应用安全防护，比如 WAF 或入侵检测；用 AI 做攻击应用，比如自动化漏洞挖掘或攻击验证码；从 AI 自身安全来说，可以用 AI 来攻击 AI……纷繁复杂的 AI 应用需要大量且多种多样的数据集。因此，构建一个庞大的安全大数据中心是整个 SecXOps 的基础，数据驱动着后续的开发和模型服务。另外，需要确保数据管道畅通，上传、下载和导入到项目等功能更加快捷，提升数据的使用效率。增加更多自动化数据预处理、特征提取步骤，将安全数据分析的历史经验应用起来，让后续的数据分析、模型构建站在巨人的肩膀上。

- ModelOps 层面：模型监控、迭代与可视化，自定义库训练模型代码工作量减少

由于攻防技术更新迭代速度之快，发生在安全数据上的概念漂移现象非常明显。一个好的安全产品需要不断更新其内使用的 AI 模型，保证产品的时效性和可用性。SecXOps 期望将模型发布后的性能监控、漂移检测和模型再训练一键化、可视化，帮助用户更好地管理和规划基础架构成本，记录和追踪模型迭代过程中产生的超参数信息，让数据科学团队、机器学习工程师和软件开发团队更专注于构建和维护系统，业务人员和维护人员可以轻松获取模型各项指标并持续进行监控。

- MLOps 层面：更广泛的机器学习应用，共享内容及零基础友好性

MLOps 平台有很多软标准，例如是否涵盖广泛的 ML 任务类型，是否拥有良好的社区共享内容，以及使用平台是否需要很高的专业水平等。SecXOps 期望为绝大多数机器学习任务提供一个良好的实验及部署平台，从简单的机器学习、到深度学习模型甚至强化学习算法，都可以借助 SecXOps 大放异彩。另外，SecXOps 提供完整的数据、代码和模型共享，期望用户能够按照个人需求，快速挑选出合适的资源加以利用。同时，SecXOps 也将继续致力于完善 AutoML 功能模块，让机器学习模型的构建对初学者更加友好。

- PlatFormOps 层面：简化环境配置，构建安全分析生态

在 AI 模型的开发过程中，经常需要借鉴以往项目的经验，从 GitHub 上或从其它途径获取的代码时常因为环境配置问题而导致运行困难重重。SecXOps 期望将复杂的环境配置简单化、流水线化，通过容器化部署、结合低代码、拖拽创作方式等来快速构建研究环境，充分沉淀与吸收前人经验作为知识载体进行共享复用，形成开放共创的安全分析生态，满足不断变化的运维、安全需求。

07

总结

安全智能分析技术已进入以大数据和人工智能技术为核心能力的新阶段，面对大规模、多源、高维运营数据的涌入与融合，构建有效的安全智能分析技术体系，支撑网络安全防御迈向高度自动化、智能化，解放安全分析的生产力，逐渐成为网络安全防御的核心动能之一。

本技术白皮书描述了安全智能分析背景和趋势，以及面临的挑战，提出了 SecXOps 概念内涵、技术优势以及核心能力，从安全分析的实践出发，重点总结了 SecXOps 关键技术在五个安全分析场景中的应用实践。安全智能分析技术通过 SecXOps 将更好的应用到网络安全的实际场景中，有效提升网络安全各个方面的威胁检测、态势感知等能力，推动安全智能分析从基础级迈向卓越级。当然，SecXOps 技术的研究和攻防场景应用实践仍然具备较大的上升空间，在理论方法、标准制定和模型运营等方面需要进一步的研究与探索，期望白皮书能够促进 SecXOps 技术体系的成熟，以构建智能分析的良好技术生态。

参考文献

[1] 方滨兴 , 时金桥 , 王忠儒 , 余伟强 , 人工智能赋能网络攻击的安全威胁及应对策略 , Publisher, City, 2021.

[2] 《AISecOps 智能安全运营技术白皮书》.

[3] https://baijiahao.baidu.com/s?id=1729511007848802003&wfr=spider&for=pc.

[4] D. Arp, et al, Dos and don’ ts of machine learning in computer security., in: Proc. of the USENIX Security Symposium, 2022.

[5] 杨晨 , 马瑞成 , 王雨石 , 翟岩龙 , 祝烈煌 , 深度学习与工业互联网安全 : 应用与挑战 , Publisher, City, 2021.

[6] Gartner Top 10 Data and Analytics Trends for 2021, in, Gartner, 2021.

[7] Gartner., Demystifying XOps: DataOps, MLOps, ModelOps, AIOps and Platform Ops for AI, in, 2021.

[8] D. Hendrycks, N. Mu, E.D. Cubuk, B. Zoph, J. Gilmer, B. Lakshminarayanan, Augmix: A simple data processing method to improve robustness and uncertainty, Publisher, City, 2019.

[9] T. DeVries, G.W. Taylor, Improved regularization of convolutional neural networks with cutout, Publisher, City, 2017.

[10] B. Li, F. Wu, S.-N. Lim, S. Belongie, K.Q. Weinberger, On feature normalization and data augmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12383-12392.

[11] C. Gong, T. Ren, M. Ye, Q. Liu, Maxup: A simple way to improve generalization of neural network training, Publisher, City, 2020.

[12] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, Publisher, City, 2017.

[13] S. Yun, D. Han, S.J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: Regularization strategy to train strong classifiers with localizable features, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6023-6032.

[14] J. Holland, P. Schmitt, N. Feamster, P. Mittal, New directions in automated traffic analysis, in: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 3366-3383.

[15] P. Liashchynskyi, P. Liashchynskyi, Grid search, random search, genetic algorithm: a big comparison for NAS, Publisher, City, 2019.

[16] J. Snoek, H. Larochelle, R.P. Adams, Practical bayesian optimization of machine learning algorithms, Publisher, City, 2012.

[17] P.J. Van Laarhoven, E.H. Aarts, Simulated annealing, in: Simulated annealing: Theory and applications, Springer, 1987, pp. 7-15.

[18] D. Whitley, A genetic algorithm tutorial, Publisher, City, 1994.

[19] T. Elsken, J.H. Metzen, F. Hutter, Neural architecture search: A survey, Publisher, City, 2019.

[20] https://nni.readthedocs.io/en/stable/.

[21] https://cloud.tencent.com/developer/news/704682.

[22] https://bbs.huaweicloud.com/blogs/361213.

[23] https://aws.amazon.com/cn/blogs/china/a-b-testing-ml-models-in-production-using-amazon-sagemaker/.

[24] https://www.ibm.com/docs/zh/cdfsp/7.6.1.1?topic=manager-migration-environments.

[25] 孟凡杰、苏菲、谢文利、李建强 , 《Kubernetes 生产化实践之路》, 电子工业出版社 , 2020-12.

[26] 绿盟科技 , 流量全密化趋势下的检测困境和思考 , in, 2022.

[27] http://datacon.qianxin.com/blog/archives/122.

[28] Gartner, Magic Quadrant for Cloud Web Application and API Protection, in, 30 August 2022.

[29] Service Name And Transport Protocol Port Number Registry[EB/OL], in, 2020-11-01.

[30] A.G. Nguyen T, A survey of techniques for internet traffic classification using machine learning[J], Publisher, City, 2008, 10.

[31] Z.Y. Yu H, Xiong G, et al. , POSTER: mining elephant applications in unknown traffic by service clustering[C]. , in: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2014, pp. 1532-1534.

天枢实验室

天枢实验室立足数据智能安全前沿研究，一方面运用大数据与人工智能技术提升攻击检测和防护能力，另一方面致力于解决大数据和人工智能发展过程中的安全问题，提升以攻防实战为核心的智能安全能力。

NSFOCUS

THE EXPERT BEHIND GIANTS

巨人背后的专家

扫描绿盟科技官微二维码
可在手机端直接观看报告电子书