arXiv:2301.00484v1 [cs.DC] 1 Jan 2023

面向远程工业 4.0 应用的联邦雾计算

Razin Farhan Hussain

提交给研究生院教员的学位论文，部分满足哲学博士学位的要求

路易斯安那大学拉斐特分校 2022年秋季

批准：

Mohsen Amini Salehi，主席 高级计算机研究中心

Sheng Chen 高级计算机研究中心

Xu Yuan 高级计算机研究中心

Li Chen 高级计算机研究中心

Mary Farmer-Kaiser 研究生院院长

© Razin Farhan Hussain

2022

版权所有

摘要

工业4.0基于物联网设备、传感器和执行器运行，改变了各个领域对计算资源和软件解决方案的使用方式。各种工业4.0延迟敏感型应用程序基于机器学习运行，并利用生成的传感器数据实现自动化及其他工业活动。将传感器数据发送到云系统耗时较长，不利于满足应用程序的延迟约束。在这种情况下，可以利用雾计算支持延迟敏感型应用程序。在异构雾系统中执行这些应用程序时，会呈现随机执行时间行为，进而影响任务完成时间。因此，我们研究并建模了各种工业4.0基于机器学习的应用程序的随机执行，并介绍了工业4.0应用程序在现实场景中的执行时间轨迹。

石油和天然气等偏远海上产业容易遭遇灾害，需要协调开展各种延迟敏感型活动。因此，灾害期间计算需求激增，可能导致其采购的雾计算资源超额订阅。为此，本学位论文提出将邻近的雾计算系统联邦化并组建雾联邦，以增强远程工业4.0站点抵御计算需求激增的能力。我们提出了一种针对延迟敏感型任务、适用于雾联邦的统计资源分配方法。

许多现代工业4.0应用程序基于微服务工作流运行，这些微服务仅在单个工业站点内使用。因此，工业4.0解决方案需要了解应用程序的架构，尤其是单体式与

微服务架构之间的区别。因此，我们提出一种基于概率的资源分配方法，可在雾联邦中划分微服务工作流，以满足其延迟约束。工业4.0的另一个关注点是联邦雾计算的数据隐私。因此，我们提出一种基于联邦学习的解决方案，可在联邦雾系统中训练工业机器学习应用，同时不损害数据机密性。

献给我的妻子 *Rezwana Mahjabeen*、我的女儿 *Ruzainah Shehzeen Hussain*、我的父母 *H.K.M. Altaf Hussain* 和 *Fariha Akhter*、我的姐姐 *Sabrina Shahreen Sarna*、我的兄弟 *Shoumin Rafsun Hussain*，以及我所有的朋友和挚爱。

致谢

我衷心感谢我的导师 Mohsen Amini Salehi 博士一直以来的鼓励，以及他对计算机科学的热情；尤其感谢他给予我的指导、支持与合作。感谢我的论文委员会成员 Xu Yuan 博士、Li Chen 博士和 Sheng Chen 博士。感谢 Sm Zobaed、Ali Mokhtari、Davood Ghatreh Samani 和 Chavit Denninart 为本学位论文的工作提供帮助。最后，感谢路易斯安那大学拉斐特分校高级计算机研究中心和研究生院给予的支持与指导。

目录

摘要 .iii

献词 .v

致谢 .vi

表目录 .xii**图目录** .xiv**第1章：引言** .1

1.1 研究问题与目标 .7**1.2 贡献** .8**1.3 论文组织结构** .9

第2章：背景与文献研究 .13**2.1 计算：工业4.0的重要组成部分** .13**2.2 工业4.0中的分布式计算系统** .14**2.2.1** 云计算 .14**2.2.2** 面向远程工业***4.0***的边缘计算与雾计算 . . . . 16**2.2.3** 边缘到云连续体 .18**2.2.4** 智能油气中的边缘到云连续体用例 .19**2.2.5** 油气行业的计算全景 .21**2.3 智能油气：数据与软件方面** .22**2.3.1** 油气行业中的大数据 .22**2.3.2** 机器学习作为油气行业中的数据驱动应用程序 .23**2.3.3** 数字孪生：油气行业中的另一种数据驱动应用程序 .25**2.4 面向人工智能及智能油气中其他数据驱动应用程序的边缘到云** .26**2.5 远程工业4.0中的联邦雾计算及其挑战** . . . . 27**2.5.1** 工业***4.0***实时服务 .28**2.5.2** 远程工业***4.0***中的异构雾系统 .29**2.5.3** 雾系统中的任务完成不确定性 . . . . . 30**2.5.4** 工业***4.0***应用程序的软件架构 .31

2.6 工业4.0中雾联邦的范围 .33**2.7 联邦雾计算系统的数据隐私方面** .34**2.7.1** 雾联邦中联邦学习的主要挑战 . . . . . 35**2.8 工业4.0中智能解决方案的不足** . 36**2.9 本学位论文的总结与定位** .38

第3章：云与雾系统中基于深度神经网络的应用性能分析 .40**3.1 概述** .40**3.2 工业4.0中的基于深度神经网络的应用** .41**3.2.1** 火灾检测 .41**3.2.2** 人体活动识别 . 42**3.2.3** 溢油检测 .43**3.2.4** 声阻抗估计 .44**工业 4.0 计算平台** .44**3.3.1** 亚马逊云 . 44**3.3.2** 作为雾计算系统的 ***Chameleon*** .46**性能建模的环境设置** .46**3.5 以应用为中心的推理时间分析** . . . . . 47**3.5.1** 概述 .47**3.5.2** 推理执行时间的统计分布 . 48**3.5.3** 集中趋势与离散程度度量分析 .52**3.6 以资源为中心的推理时间分析** . 53**3.6.1** 使用刀切法估计置信区间 .55**3.6.2** 使用 ***Bootstrap*** 方法估计置信区间 . 56**3.7 总结与讨论** .58

第4章：联邦雾计算管理单体式
偏远工业现场中的工作负载 .60
4.1 **概述** w 60
4.2 **联邦雾系统中的端到端时延** .61
4.2.1 估算通信延迟 .61
4.2.2 估计计算延迟 .63
4.2.3 估计端到端延迟 .64

4.3 联邦雾计算系统中的鲁棒资源分配 .65**4.4 联邦雾计算的性能评估** .
69**4.4.1** 负载均衡器的基准任务分配启发式方法 .70**4.4.2** 实验结果 .71**4.5 总结** .
74**第5章：将远程工业4.0智能微服务应用适配至联邦雾计算系统** .76**5.1 概述** .
76**5.1.1** 面向工业***4.0***的智能微服务应用 .77**5.1.2** 面向工业***4.0***微服务应用的联邦雾系统 .78**5.2 微服务应用工作流的分区方法** .81**5.3 雾联邦中分区微服务应用的资源分配方法** .85**5.4 软件架构感知型联邦雾系统的性能评估** .89**5.4.1** 微服务工作流划分方法的比较 .89**5.4.2** 资源分配方法的比较 . . . . . 91**5.4.3** 雾联邦扩展的影响 .
93**5.5 总结** .96**第6章：联邦雾计算系统中的数据安全与隐私方面** .98**6.1 概述** .
98**6.2 联邦学习问题表述** . . . . . 100**6.3 通过联邦学习缓解类别不平衡** . . . . . .
102**实验设置** .104**性能评估** .105**6.5.1** 损失函数调优 .106**6.5.2** 使用独立同分布数据分布的影响 . . . . . 107**6.5.3** 使用非独立同分布数据分布的影响 . . . . 108**6.5.4** 使用非独立同分布且不平衡数据分布的影响 .109**6.5.5** 类别不平衡强度的影响 . . . .
. . . . 111

6.5.6 工作节点数量对全局模型的影响 .1126.5.7 验证聚合方案对全局模型的影响 .1136.6 总结 .115

第7章：工业智能解决方案的威胁与副作用
4.0 .1187.1 概述 .1187.2 智能油气行业中的网络威胁与副作用分类体系 .1207.3 信息技术与运营技术交互导致的漏洞 . . . . . 1217.4 智能油气行业中的网络威胁 .1247.4.1 敏感数据的漏洞 .1257.4.2 智能系统的脆弱性 .1267.4.3 恶意软件与信息技术（*IT*）的漏洞 .127不兼容的物联网设备 .1317.5.1 硬件层面不兼容 .1327.5.2 软件级不兼容 .1337.6 利用区块链应对智能油气领域的网络威胁 . . . . . 1347.6.1 基于区块链的控制系统（*SCADA*） . . . . . 1347.6.2 利用区块链在工业物联网中建立信任 . . . 1377.7 工业物联网中智能解决方案的风险 .1387.7.1 人机交互问题 .1387.7.2 机器对机器交互问题 .1407.8 智能工业中的偏见 .1427.8.1 人工智能（*AI*）解决方案导致的偏见 .1427.8.2 工业*4.0*智能解决方案中的自动化偏见 .1447.8.3 油气行业中的性别偏见 .1457.8.4 智能油气解决方案中的认知偏差 . . . . . 1457.9 总结 .146

C第8章：结论与未来研究方向 .1488.1 讨论 .1508.2 未来研究方向 .155

8.2.1 工业*4.0*应用中跨联邦雾系统的基于强化学习的资源分配 .155
8.2.2 跨联邦雾系统的数据局部性感知资源分配 .155
8.2.3 面向工业*4.0*操作的动态容错联邦雾系统 .156
8.2.4 面向智能工业*4.0*解决方案的人机交互认知方面 158
8.2.5 工业领域人机交互的雾计算与高级分析 ... 160

参考文献 .161

个人简介 .178

表目录

表3.1. 油气工业4.0中使用的基于深度神经网络的应用程序及其网络模型、输入数据类型、使用范围和代码库。 42

表 3.2。 Amazon EC2 中为基于深度神经网络的应用进行性能建模而考虑的异构机器类型和虚拟机配置。在此表中，机器学习优化型表示 AWS 提供的 Inferentia 机器类型.45

表3.3。 Chameleon 云中的各种虚拟机规格经过配置，用于表示一致的异构系统. 46

表3.4。 使用夏皮罗-威尔克检验得出的 AWS 云中机器上基于深度神经网络的应用的执行时间分布。 .48

表3.5。 使用夏皮罗-威尔克检验得出的 Chameleon 云中深度神经网络应用的执行时间分布。 .49

表3.6. 使用柯尔莫哥洛夫-斯米尔诺夫检验分析 AWS 云端机器上基于深度神经网络的应用程序的推理时间分布。 .50

表3.7. 使用 K-S 检验分析 Chameleon 机器上基于深度神经网络的应用程序的推理时间分布。 .50

表3.8. AWS 中观测推理时间的集中趋势指标（μ）和数据离散度指标（σ）的测量结果。 .52

表3.9. Chameleon 云中推理时间的集中趋势指标（μ）和数据离散度指标（σ）。 .52

表3.10. AWS 中异构机器针对每个基于深度神经网络的应用程序的 MIPS 值。 .54

表3.11. Chameleon 云上异构机器针对每种基于深度神经网络的应用程序的每秒百万条指令数值。 .54

表3.12。通过刀切重采样法得出的 AWS 机器上基于深度神经网络的应用程序的 MIPS 值置信区间。.55

表3.13。通过刀切重采样法得出的 Chameleon 机器上不同基于深度神经网络的应用程序的 MIPS 值置信区间。.55

表 6.1. 溢油检测数据集中各类别的像素分布.102

图目录

图 1.1. 面向实时延迟敏感任务的各类智慧行业（例如，石油和天然气、医疗保健、交通运输）中的先进计算系统.2

图1.2. 一个偏远的海上智能油田由多座石油钻井平台（石油开采场）组成。在这种场景下，石油钻井平台、钻井船，甚至救援船都配备了以移动数据中心形式构建的雾计算系统，以满足石油开采计算需求，并应对任何不可预见的紧急情况（例如，溢油检测、有毒气体检测）.5

图1.3. 边缘-雾-云范式中典型微服务应用“消防安全”的执行场景.6

图2.1. 智能石油和天然气行业可利用各种云服务（例如，仿真、分析、可视化、计算、机器学习、报告），存储、处理和分析传感器生成的数据，并控制工业设备。. 15

图 2.2。 以石油和天然气行业为例的工业4.0边缘到云连续体。该连续体主要分为四个层级，即终端设备、边缘、雾和云数据中心。三角形底部是能源受限的终端设备，而向顶部延伸时，则会发现能耗更高的系统.18

图2.3。 基于无人机的检查场景：无人机采集图像，并可在边缘计算资源中执行实时分析，而长期分析则在远程云计算设施中进行.20

图 2.4. 工业4.0 智能应用的各种软件架构。地震分析采用单体应用，而消防安全应用则采用微服务架构.31

图 2.5. 典型的联邦学习场景，由联邦学习工作节点和一个持有全局模型的中央服务器组成。在训练开始时，全局模型会广播给参与工作节点，以便它们使用各自对应的本地训练数据进行训练。在联邦学习工作节点经过一段时间的训练后，更新后的模型会发送回服务器，以便与全局模型进行整合。. 35

图 3.1。 FCN-8 模型以框图形式呈现，由 5 个全卷积网络模块和 2 个上采样模块组成。模型接收 SAR 图像作为输入，并执行逐像素分类，输出一幅标注图像。 .43

图3.2。 时间卷积网络（TCN）模型的示意图，该模型由六个时间模块、输入数据和以预测的人工智能形式呈现的输出组成。 .44

图3.3。 在 AWS 上的异构虚拟机上运行时，溢油检测应用程序推理执行时间的随机性。对于每个虚拟机实例，溢油检测应用程序执行 30 次，并将这些执行按横轴上的尝试次数绘制。纵轴表示每次尝试的推理时间。 .47

图3.4。 AWS 和 Chameleon 机器针对各种基于深度神经网络的应用程序的 MIPS 值比较分析。为便于展示，MIPS 值归一化至 $[0,1]$。 .57

图4.1。 配备负载均衡器模块的雾系统，该模块可促进雾联邦。传感器生成的任务请求由负载均衡器模块接收，并分配给最有可能成功完成任务的雾系统. 66

图4.2。 使用负载均衡器中的不同任务分配启发式算法时，超额订阅级别（到达任务数量）增加对截止期限错失率的影响.71

图4.3。 不同启发式算法在雾联邦中对每个任务引入的平均通信延迟开销.73

图4.4. 使用各种任务分配启发式算法时任务的平均完工时间（秒）.74

图5.1. 基于微服务的工作流结构以框图形式呈现。要完成消防安全应用，必须处理每个微服务.77

图5.2. 海上油气行业拥有可支持智能微服务应用的雾联邦基础设施.79

图5.3. 工作流分区方法的流程图。已分区工作流被发送至资源分配模块，该模块在此流程图中以终止框表示.82

图 5.4. 采用所提出的概率分区技术时，各分区技术在工作流截止期限达成率方面的比较。横轴表示到达的工作流执行请求数量的增加，纵轴表示工作流截止期限达成率.90

图5.5. 采用所提出的工作流划分技术时，基于微服务的工作流应用的资源分配技术比较.92

图5.6. 单体应用的资源分配技术比较。所提出的资源分配技术 MR在每次应用到达试验中均优于其他基线方法.93

图5.7. 在考虑微服务应用的情况下，随着超额订阅情形增加，扩展雾联邦对所提出的分区和资源分配技术的影响。度数表示每个雾系统为执行工业4.0应用所拥有的邻居数量.94

图 5.8. 扩展雾联邦对单体应用所采用的资源分配技术的影响。度数表示每个雾系统为执行工业4.0应用所拥有的邻居数量.95

图6.1。 雾联邦中的联邦学习设置。多家公司共享其雾系统，以训练溢油检测深度神经网络模型，同时通过联邦学习保障数据安全。 99

图 6.2。 考虑类别不平衡和全局收敛的联邦学习训练。训练时使用 Tversky 损失来应对类别不平衡。每轮训练结束后，使用动态阈值检查平均交并比（ mIoU），以判断是否达到全局收敛.103

图6.3. 使用 Tversky 损失函数比较 FedAvg 和 FedBal 的训练损失。Tversky 指数的 alpha 参数从 0.6 变为 0.8（从左到右），并记录了 FedAvg 和 FedBal 算法每轮的损失.107

图 6.4. 在独立同分布设置下，比较 FedBal 与 FedAvg、FedSGD 和 FedProx 方法的全局模型性能.107

图6.5. 使用 FedAvg 和 FedBal 方法，按平均交并比比较全局模型的性能。数据分布为非独立同分布，工作节点数量为 6，每轮联邦训练均进行了 50 轮训练. 109

图 6.6. 在非独立同分布且数据分布不平衡的情况下，比较 FedBal 与 FedAvg、FedProx 和 FedSGD 方法的全局模型性能.110

图 6.7. 在非独立同分布数据分布下，从高强度（每个工作节点 1 个类别）到低强度（每个工作节点 3 个类别），比较 FedAvg 与 FedBal 方法的全局模型性能。图中以条形图形式绘制了 3 种情形（1 个类别、2 个类别和 3 个类别）下 FedBal 与 FedAvg 的平均交并比差值.112

图 6.8. 通过将联邦工作节点数量从 6 个增加到 25 个，测量联邦工作节点对 FedBal 和 FedAvg 全局模型性能（平均交并比）的影响。对于工作节点池的每种情况，FedBal 和 FedAvg 方法均进行 20 个联邦轮次的训练，并将两种方法各自的最大平均交并比绘制为条形图.113

图 6.9. 工作节点权重（在每个联邦轮次上取平均）对全局模型平均交并比的影响. 114

图7.1. 采用方框流程图形式整理的分类体系，反映了使用先进技术实现的智能解决方案的缺点。三个主要层级分别以橙色、蓝色和黄色表示。白色方框代表其父方框的不同类型（示例）.120

图7.2. 智能石油和天然气公司通过不同的网络运行，以管理智能油气行业的整体运营。信息技术（IT）平台与业务应用和油气行业的财务方面密切相关，而运营技术（OT）平台则直接涉及石油或天然气开采及生产运营。IT 和 OT 平台在某些环节相互连接，从而形成网络攻击者渗透整个系统的突破口.122

图7.3. 勒索软件从开始到结束的结构。勒索软件客户端通过恶意电子邮件或其他外部媒介进入 IT 平台。随后，客户端与黑客的命令与控制服务器通信，以下载加密密钥。勒索软件客户端会加密用户数据，最后发送勒索通知.129

图 7.5. 基于区块链的石油和天然气公司端到端SCADA系统内的数据传输。区块链可在传输数据以供处理时进行加密，即使数据在传输过程中遭到劫持，也能提高数据安全性。. 136

图 7.5. 从感知到控制操作的人机交互工作流程.139

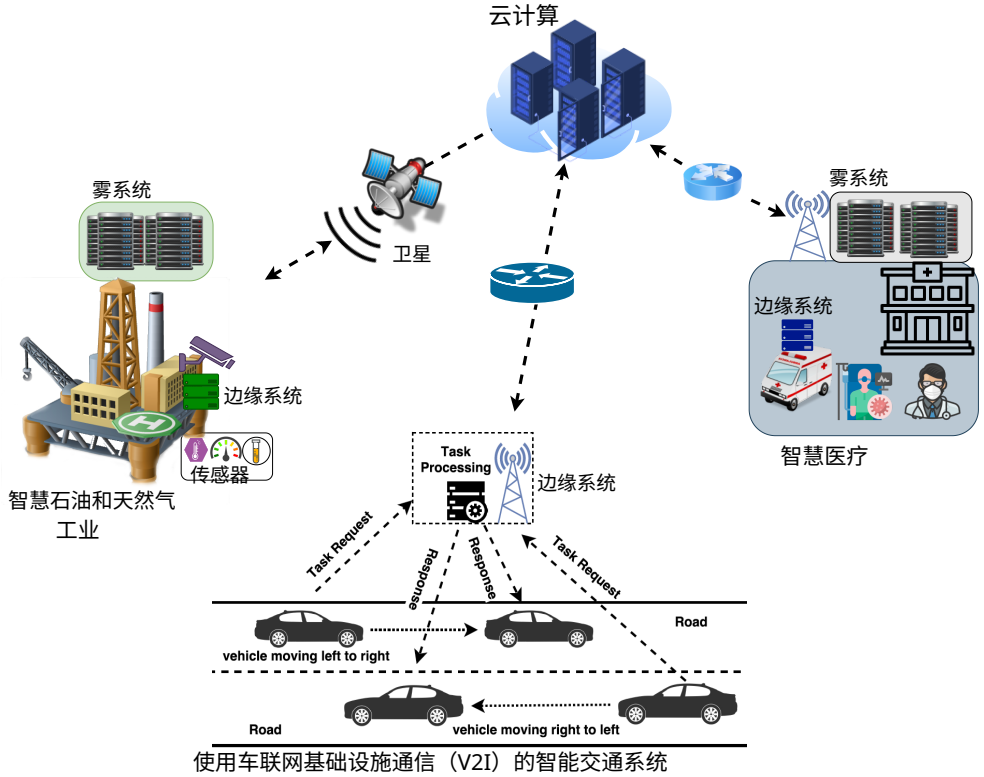
图7.6。 压缩机组所在的封闭石油生产区发生了一起小型火灾事故。火灾报警器发出警报，洒水喷头开始喷水，导致发电机发生电力故障，电动门因此锁闭。遗憾的是，工作节点当时正在进行管道维护，因门关闭而被困在设施内。在此，机器间交互引发了现场工作人员的安全问题.141

第1章：引言

工业4.0正在彻底改变各行业对计算资源的利用方式 [1]。随着物联网（IoT）和现代计算系统（例如，边缘计算、雾计算和无服务器计算）的兴起，各行业借助智能传感器和执行器变得更加智能，这些设备每天都会生成大量数据 [2]。然而，存储和分析传感器生成的数据所需的计算资源成本高昂，在偏远地区尤其稀缺 [3]。在工业领域，各类应用程序（例如，机器学习（ML）、报告、警报生成器和监控）利用传感器生成的数据来实现自动化或执行复杂操作。有时，这些数据需要实时反馈，以便开展容错性要求低且对延迟敏感的活动（例如，石油钻井平台上的钻井作业、工人安全作业、产品制造）。另一方面，某些任务需要强大的计算能力，且对延迟的容忍度较高，因此需要云数据中心的支持。例如，“消防安全”应用程序 [4]使用一种深度神经网络（DNN）模型，该模型需要在配置完善的云数据中心进行大量训练。同样，石油工业广泛使用“油藏模拟”来预测不同生产策略下油田的表现，这需要大量地震数据和高性能计算系统[5]。

在工业领域的偏远或遥远地区（例如，海上石油开采场、太阳能场区），将传感器生成的数据传输到云数据

图 1.1. 用于实时延迟敏感任务的各智慧行业先进计算系统（例如，石油和天然气、医疗保健、交通运输）



中心成本高昂且时延大，影响了在数据源附近进行计算的使用 [6]。此外，实时应用要求更快的响应时间，而这通常无法通过云计算资源实现。因此，将计算资源部署到终端客户附近的数据源处，是偏远地区工业的一项基本要求。

靠近数据源进行计算的一种解决方案是边缘计算 [7]，如图 1.1 所示。它将计算资源部署在更靠近终端设备和数据生成源的位置。因此，边缘计算可定义为一种分布式计算平台，将工业应用部署在更靠近数据源的位置，例如物联网设备或本地计算机服务器。与数据源的这种接近性

可以带来显著的业务效益，例如更快地获取洞见、更快的响应速度以及更充足的带宽。尽管边缘计算支持实时延迟敏感型应用，但边缘设备的资源有限，因此需要高效的资源分配 [8] 机制。因此，靠近终端用户的计算平台还有另一种解决方案，即雾计算系统 [9] 。雾计算系统通过提供更多计算资源以及更全面的中间件来高效管理工作负载，从而补充边缘计算。

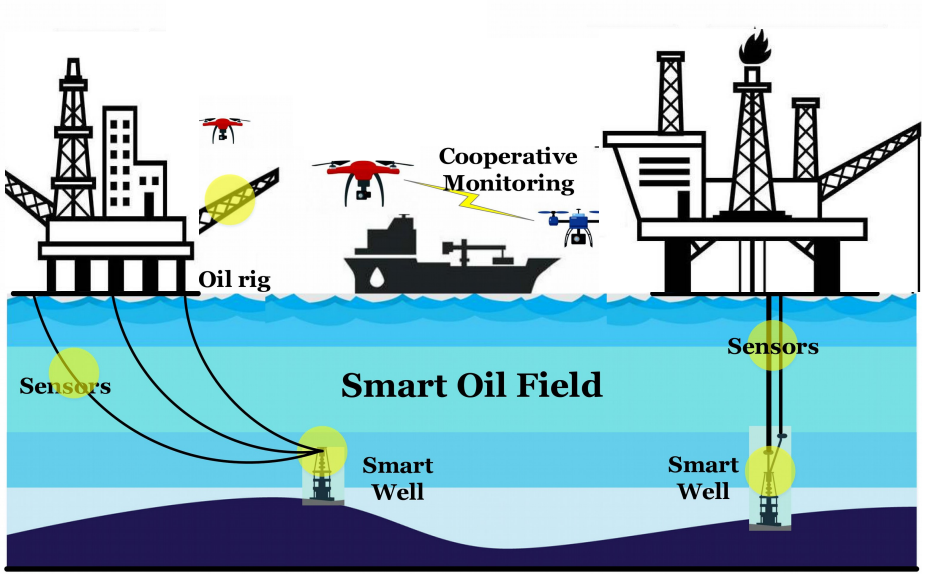
工业4.0革命的主要驱动力是机器学习（ML）或深度神经网络（DNN）应用程序 [10, 11, 12] ，它们可确保高效的工业运营和工作场所安全。因此，基于机器学习或深度神经网络的应用程序涵盖训练阶段和推理阶段 [13]。由于时间和计算资源受限，训练阶段通常在线下进行，而推理执行则可通过通用计算系统完成。基于深度神经网络的应用程序主要在配置了高端硬件（例如 图形处理器、TPU、FPGA）的云数据中心或计算服务器上训练。相比之下，推理操作则在靠近终端用户的雾计算系统上执行。因此，系统工程师或系统管理员必须了解这些基于深度神经网络的应用程序在雾计算系统中的性能 [14]。尤其对于容错型延迟敏感关键深度神经网络应用，预测其在计算资源上的推理执行时间至关重要，有时甚至能够在灾害情境中挽救生命。因此，我们对推理

时间进行了统计分析，研究对象是云与雾系统上各种工业4.0应用的推理执行时间。由此，我们提供了一种执行时间工作负载轨迹，可帮助系统架构师制定负载均衡方案，使其能够应对工业4.0智能应用随机变化的执行时间。因此，采用这些稳健的方案能够确保效率、生产力和工业安全。

在偏远海上工业中，紧急情况（例如 灾害、事故）发生时，边缘或雾计算系统中的任务处理需求可能会大幅增加，导致一些任务因未能满足其延迟约束（又称截止期限）而被丢弃。因此，我们建议整合附近私有的计算资源，组建雾系统联盟，以应对紧急情况期间激增的任务处理请求。例如，如图1.2所示，在偏远海上智能油田中，各家公司可以分别建设多个石油开采场，这些开采场通常配有私有雾计算系统，以满足日常计算需求。发生灾害或爆炸等其他紧急情况时，为协调多项恢复程序，计算需求会激增。例如，发生火灾时，无人机巡检、火灾检测以及生成带有精确火灾位置的警报等各项活动，都需要实时协调，以消除紧急情况。在这种情况下，一些延迟敏感型任务可以卸载 [15] 到其他计算资源更多或负载较低的雾系统。因此，联邦雾系统的韧性取决于能否通过在联邦内高效分配资源来满足激增的计算需求。因此，我们

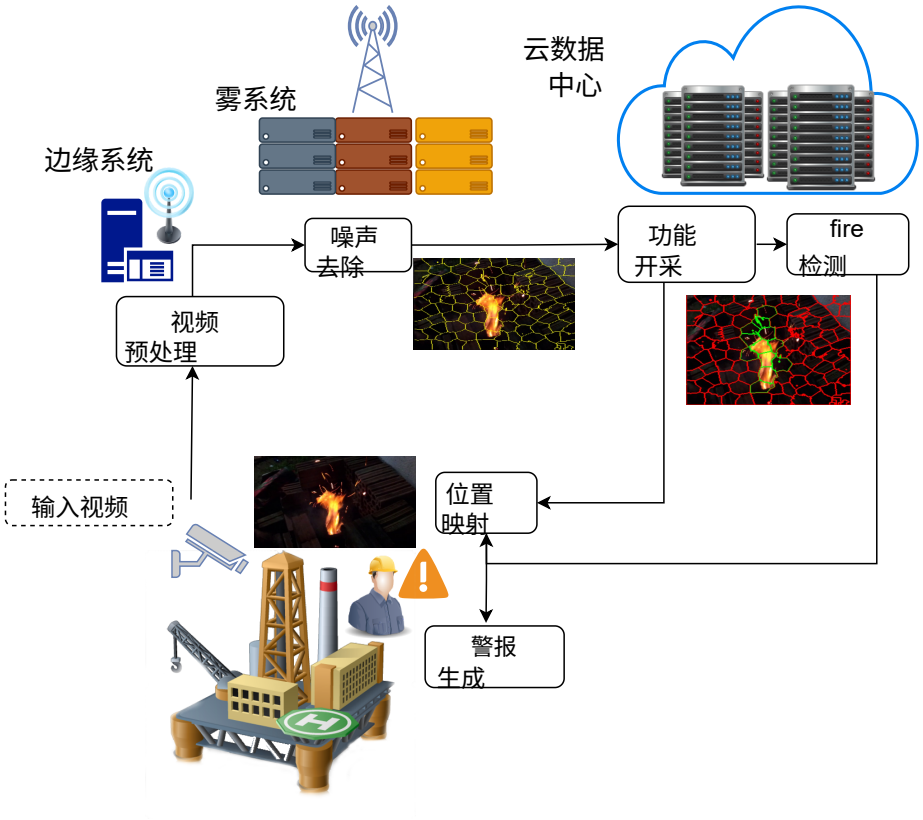
建议采用一种面向时延敏感型单体任务的雾联邦概率资源分配方法，以满足紧急情况下的计算需求。

图1.2。 远程海上智能油田由多个石油钻井平台（石油开采地点）组成。在此场景中，石油钻井平台、钻井船，甚至救援船都配备了以移动数据中心形式存在的雾计算系统，用于支持石油开采的计算需求以及应对任何不可预测的紧急情况（例如：溢油检测、有毒气体检测）



工业4.0中的智能应用可以采用各种软件架构（例如，单体式、微服务 [16]），以满足工业运营的不同需求。因此，微服务架构是广泛使用的软件架构之一，它由各种微级服务组成，在开发与部署方面具有显著优势 [17]。例如，如图 1.3 所示，基于“消防安全”微服务的应用程序包括视频预处理、噪声去除、特征提取、火灾检测、位置映射和警报生成等微服务，它们分别执行不同的任务。在典型的工业场景中，这些微服务由各种执行平台提供支持，而这些平台可能具有

图 1.3。 边缘-雾-云范式中的典型微服务应用“消防安全”执行场景。



随机执行延迟。相比之下，单体架构是构建软件应用程序的传统统一范式。单体软件旨在独立运行，其组件或服务紧密耦合，而非像微服务架构那样松散耦合。尽管工业革命推动了微服务应用的使用，但各行业仍有许多单体遗留应用正在运行，需要现有执行平台提供支持。在紧急情况或灾难发生时，灾害区域附近会产生各种具有不同延迟要求的应用请求，需要不同的计算支持。因此，

随着按时执行大量应用程序的需求激增，附近的执行平台会出现超额订阅，从而可能导致执行平台的性能下降。在这种情况下，为了利用所提出的联邦雾系统满足高计算需求，需要一种高效的资源分配方法，并能感知接收应用程序的内部结构、计算需求和通信延迟。因此，超额订阅情形下执行平台的可靠性取决于其能否满足各种计算需求，从而确保工业安全。

在偏远工业区域联邦化计算资源，会给联邦中的每个参与雾系统带来安全方面的顾虑，因为这些雾系统由私营公司所有。此外，各个雾系统可能包含敏感数据，这会给拥有这些计算系统的公司带来隐私问题。因此，考虑到跨雾联邦进行机器学习应用训练会受到数据稀缺的影响，而这又会妨碍构建准确的机器学习模型，因此应采用安全且保护隐私的分布式机器学习训练方法，以构建准确的机器学习模型；这类模型在工业4.0的紧急情况下可能至关重要。

1.1 研究问题与目标

本研究的根本目的是在现代边缘计算与雾计算系统中，识别、评估并管理偏远地区（例如，海上油田）的工业4.0应用程序的稳健执行。因此，我们开发了能够在紧急和超额订阅情况下利用雾系统、满足偏远工业部门计算需求的解决方案。本学位论文

探讨以下研究挑战，以构建稳健且服务质量高效的工业4.0应用联邦雾系统：

1. 在考虑工业革命和工业4.0的背景下，如何利用现代分布式计算系统服务于远程智慧行业（例如，石油和天然气、能源）？

2. 工业4.0应用程序在雾系统中的统计执行行为是什么？

3. 如何构建强健的联邦雾计算系统，以便在工作负载激增期间高效获取计算需求？

4. 如何利用现代微服务架构以及单体遗留应用支持工业4.0应用程序，并维护雾联邦的服务质量（QoS）？

5. 如何安全地利用联邦雾来提升工业4.0应用程序的性能？

1.2 贡献

在研究工业领域联邦雾计算系统的诸多方面时，我们发现了各种障碍。因此，除了应对重大挑战之外，我们还提出了最先进的解决方案，并对推荐的方法进行了全面的性能评估。根据前一节概述的研究主题，本学位论文的主要贡献如下：

• 厘清工业4.0与现代分布式计算系统（例如，边缘计算、雾计算）之间的联系，并探讨在远程智能油田背景下运用高级分析（例如，人工智能、机器学习、深度学习）的范围。

• 雾计算与云计算环境下基于机器学习的工业4.0应用性能分析?

• 提出针对四种不同基于机器学习的工业4.0应用的推理执行时间真实工作负载基准测试。

• 通过提出基于贝叶斯概率的资源分配方法，利用雾系统为超额订阅且试图从灾难中恢复的系统中的延迟敏感型任务提供支持，从而实现联邦雾的理念。

• 提出一种适用于联邦雾系统的统计资源分配方案，该方案能够感知工业4.0微服务工作流应用的内部软件架构和随机时延要求。

• 提出一种保护数据隐私的基于机器学习的工业4.0应用训练方案，适用于偏远工业现场的联邦雾系统。

1.3 论文组织结构

• 第2章探讨了相关研究工作，并介绍了边缘与雾计算、雾计算联邦系统、负载均衡与任务分配技术，以及联邦雾系统的数据隐私方面的背景。

因此，本文通过各种用例场景阐述了在远程智能油田中利用现代分布式系统的范围。

– **Razin Farhan Hussain**、Ali Mokhtari、Mohsen Amini Salehi 和 Ali Ghalambor面向石油和天然气行业智能运营的物联网由爱思唯尔以图书形式出版（ISBN：9780323998444）。

• 第3章研究了基于机器学习的工业4.0应用在异构云计算资源上的性能。对基于机器学习的应用进行统计分析，有助于生成真实世界的工业4.0应用推理执行时间工作负载，这对系统架构师开发稳健的负载均衡解决方案大有裨益。

– **Razin Farhan Hussain**、Alireza Pakravan 和 Mohsen Amini Salehi云计算系统中智能工业*4.0*应用的性能分析 发表于第22届 IEEE 国际高性能计算与通信会议论文集（HPCC-2020）

• 第4章探讨了在灾难发生时，于偏远的海上智能油田建立雾联邦的潜在优势与可行性。利用雾联邦中的概率负载均衡启发式算法进行资源分配，可以有效确保系统的弹性。此外，与基准方法相比，采用概率方法的优势得到了在 EdgeCloudSim 中创建的合成工作负载仿真的支持[18]。

– **Razin Farhan Hussain**、Mohsen Amini Salehi、Anna Kovalenko 和 Omid Semiari 面向远程智能油田灾害管理的联邦边缘计算 发表于第21届IEEE高性能计算与通信国际会议（HPCC-2019）论文集

– **Razin Farhan Hussain**、Omid Semiari 和 Mohsen Amini Salehi基于边缘计算的车联网基础设施通信（*V2I*）网络鲁棒资源分配，发表于第3届IEEE雾计算与边缘计算国际会议论文集（ICFEC'19）

– **Razin Farhan Hussain**、Mohsen Amini Salehi 和 Omid Semiari面向绿色石油和天然气产业的无服务器边缘计算，发表于IEEE绿色技术会议（GreenTech）论文集，2019年

• 第5章探讨了面向工业4.0应用的先进微服务软件架构，以增强远程联邦雾系统的鲁棒性。因此，负载均衡器应了解接收应用程序的软件架构以及执行平台的不确定性。由此，在雾联邦中分布接收应用程序，可提高应用程序按时完成的可能性。

– **Razin Farhan Hussain**, Mohsen Amini Salehi 使远程工业 *4.0* 应用适配联邦雾计算系统，为提交至 Future Generation Computing System 期刊而准备，2022年

• 第6章探讨了工业4.0联邦雾计算系统中基于机器学习的应用训练的数据隐私方面。

• 第7章探讨了工业4.0智能解决方案的弊端和副作用。本章指出并提出了针对不同工业领域安全问题的各种尖端解决方案。

> – **Razin Farhan Hussain**、Ali Mokhtari、Mohsen Amini Salehi 和 Ali Ghalambor 的著作面向石油和天然气行业智能运营的物联网由爱思唯尔出版（ISBN:9780323998444）。

• 第8章通过讨论我们的主要发现以及工业4.0领域高效利用雾计算平台的未来研究方向，为学位论文作结。

第2章：背景与文献研究

2.1 计算：工业4.0的重要方面

工业系统正迅速从由人控制的流程转向闭环控制服务，借助广泛的传感器与计算基础设施自主支持其运营。这一变革对于支持日益增长的数据密集型和时间敏感型工业4.0应用程序至关重要，尤其是在海上石油与天然气（O&G）油田等偏远地区，这些地方的计算机基础设施受限，人力资源有限。要实现这些系统，需要开展跨学科研究，探索远程工业中的工业4.0、现代计算基础设施（如边缘和云）以及高级分析（例如，机器学习、深度神经网络）之间的交汇领域。

因此，本章旨在阐述基于工业4.0范式基本原理，建立智能稳健的远程工业所面临的挑战、前景与解决方案。通过本研究，研究人员和从业者能够更有效地提升远程工业的安全性、可持续性、环保性和自动化水平，进而降低成本。本章探讨了多种支持偏远地区工业计算需求的计算机技术。此外，本章还阐释了如何利用物联网（IoT）、机器学习方法和分布式计算平台等前沿计算解决方案的协同作用，改善工业流程。作为偏远海上工业的典型例子，我们考察了石油和天然气行业；该行业随着工业

革命和工业4.0的发展，经历了重大转型。另一方面，远程海上油气行业一直面临各种灾害和灾难事件，引发了人们对生产效率和安全措施的担忧。例如，深水地平线（2010年）[19],Usumacinta自升式平台灾难（2007年） [20], 孟买高北灾难（2005年） [21],以及Ocean Ranger灾难（1982年）等事故，都与工业现场的安全故障密切相关。因此，这些事故促使我们改善偏远地区工业的计算支持，以确保安全和生产力。本章因此探讨工业4.0的各种分布式计算技术、联邦友好型执行平台、软件架构和安全方面，并重点关注油气行业。

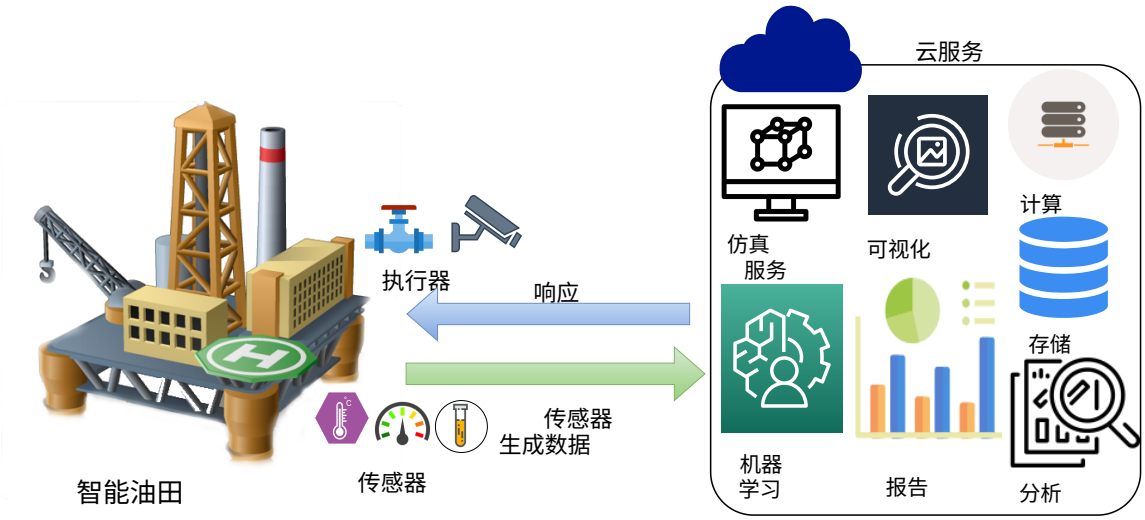
2.2 工业4.0中的分布式计算系统

2.2.1云计算

云计算是一种概念，使资源（例如，计算、存储、服务）能够按需以服务形式提供，且可配置、可共享 [22]。现代云系统提供不同层级的多种服务，例如基础设施即服务（IaaS）、平台即服务（PaaS）、软件即服务（SaaS）和函数即服务（FaaS）。

如图 2.1 所示，智能油气行业日益依赖托管在远程互联网服务器（又称云数据中心）上的基于云的服务。这些数据中心用于存储和处理其数据。根据图 2.1，各种传感器生成的数据会发送至云服务提供商，以使用不同类型的云服务。在这些服务中，有些会将有价值的决策

图 2.1. 智能石油和天然气行业可采用各种云服务（例如，仿真、分析、可视化、计算、机器学习、报告），以存储、处理和分析传感器生成的数据，并控制工业设备。



发送给执行器，从而在智能油田中闭合自动化闭环。云技术使油气公司能够使用各种数据相关服务和计算服务（例如，机器学习和可视化），而无需维护任何计算基础设施。然而，数据隐私和安全问题一直令这些公司担忧，阻碍它们全面采用云服务。这些安全顾虑使云服务的采用略有停滞，尤其是在该行业的主要参与者中。另一种更安全的替代方案是将数据存储在本地计算设施中，这种设施称为私有云 （近来称为雾计算）。

从积极的一面来看，云系统的性能和易用性对油气行业颇具吸引力。例如，数据驱动的云服务的主要用户之一是北美页岩行业，该行业每年都会钻探数千口井

每年 [23]。云服务的可扩展性有助于高效利用这些井产生的不断增长的数据量，从而推动该行业显著扩张。因此，各种现代化的基于云的数据分析服务应运而生，帮助油气公司改进运营工作流程并做出有利可图的决策。

2.2.2远程工业*4.0*的边缘计算与雾计算

由于延迟敏感型应用的重要性日益凸显，以及偏远海上产业中贴近终端用户的实时运行需求不断增加，人们对边缘计算概念的兴趣也随之升温。此外，工业领域中物联网（IoT）设备和智能传感器的广泛普及，产生了海量需要在本地处理的数据。在典型场景中，数据会被传输到云数据中心 [24], ，再通过互联网将响应或结果传回客户端，这两步都会耗费时间和金钱。因此，人们提出了一种分布式计算范式，将其部署在靠近最终客户端的位置，并在网络边缘处理客户端数据 [25]。研究人员称这种计算方式为“边缘计算”，因为它运行在网络外围。

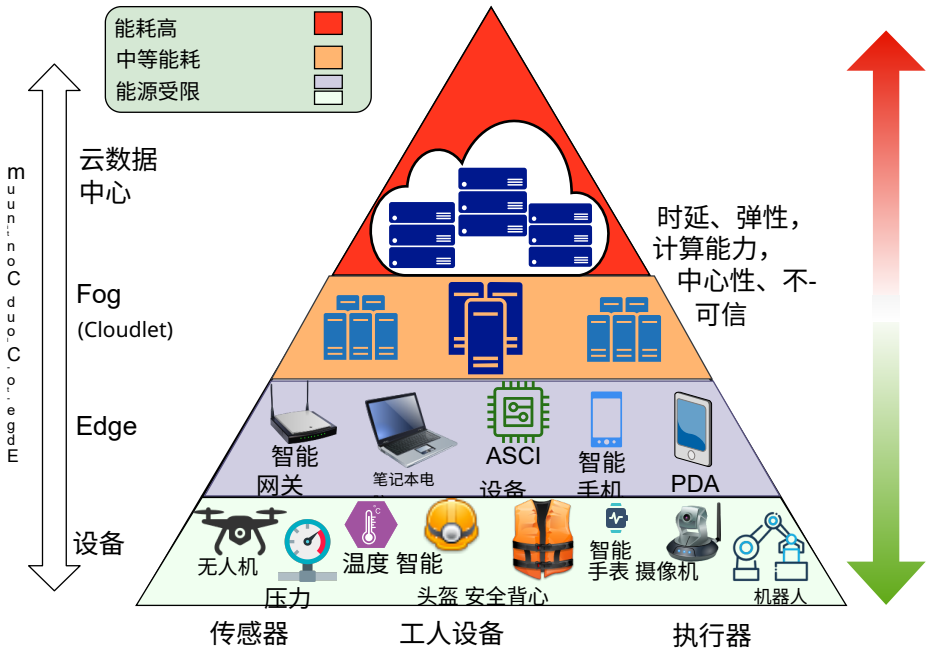
边缘计算的传统定义并不容易界定。不同组织或资料有不同的定义，而且定义会受到具体情境的显著影响。通常认为，边缘计算是通过分布式计算机系统而非集中式云数据中心来提供各种计算服务（例如，应用程序执行、数据预处理）。因此，边缘计算

能够在信息源头进行分析并收集知识。在网络设计中，“边缘计算机”位于网络端点（例如控制器和传感器）的正旁边，甚至直接部署在其上。随后，数据会在传输到云端进行存储或进一步处理之前，经过部分或全部处理。另一方面，边缘计算也可能导致大量数据直接传输到云端，从而影响系统容量、效率和安全性。雾计算 [26] 通过在边缘与云之间插入一个处理层来解决这一问题。因此，“雾计算”会在数据到达云端之前，于边缘收集并分析数据。因此，计算服务所提供的位置——即相对于数据源的位置——可以作为区分雾计算／边缘计算与云计算的关键因素。例如，一家配备了众多传感器的可再生能源公司，会在其运营领域利用雾计算进行传感器数据分析。通过将数据分析延迟从10分钟缩短到几秒，该公司的生产效率提升了15%。因此，在偏远地区工业的数据源附近部署雾计算，可以提高生产效率。

边缘计算与雾计算的出现并不会取代云计算服务；相反，它将其中一部分服务（例如，计算、存储、分析服务）带到了靠近终端客户的位置。尤其是随着物联网（IoT）设备不断增多，生成了大量数据 [2] ，这些数据对于工业4.0机器学习应用具有重要价值。生成的数据有时需要立即处理（即，边缘计算

支持），有时则需要复杂处理（即，云计算支持）。因此，需要一个连续计算平台（即，边缘到云连续体 [27]），以同时支持实时需求和复杂分析任务。

图 2.2. 以石油和天然气行业为例，展示工业4.0的边缘到云连续体。该连续体主要分为四个层级，即终端设备、边缘、雾和云数据中心。三角形底部是能源受限的终端设备，向上则是能耗更高的系统。



2.2.3边缘到云连续体

尽管边缘计算与云计算在距离和资源方面存在差异，但二者可以互为补充。对于石油与天然气这样规模庞大的行业而言，需要开展多种运营并提供多样化服务，这些运营和服务需要各种底层计算平台。因此，将边缘计算或雾计算与云计算相结合，是当今的迫切需求，体现了

边缘到云连续体[28]的可用性。因此，边缘到云连续体是一个服务平台，为支持油气行业所必需的不同类型服务提供各种计算资源和基础设施。

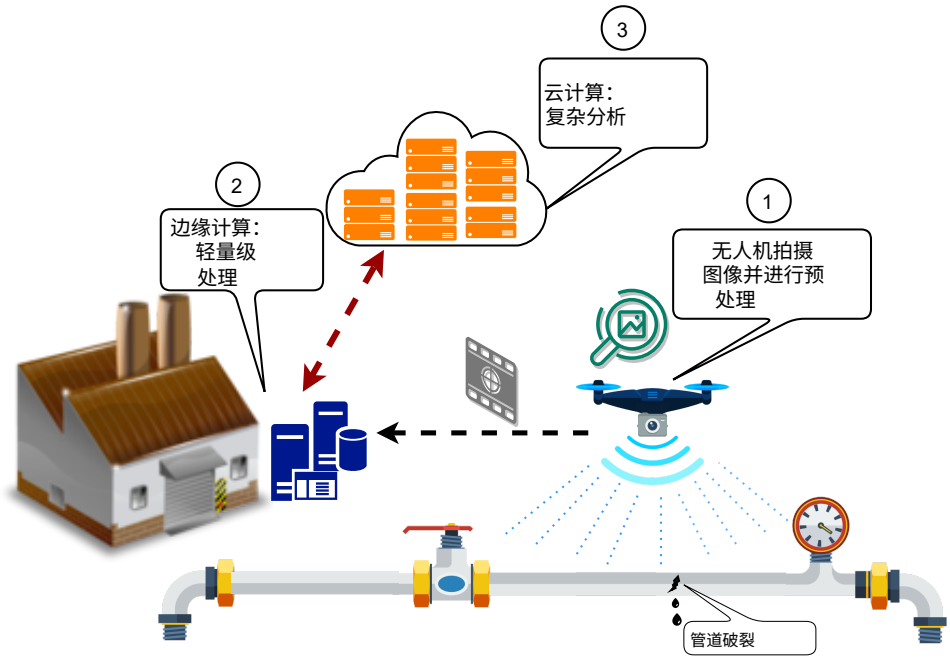
图 2.2 将边缘到云连续体展示为一个三角形：边缘设备位于靠近终端设备的位置（三角形底部），而云数据中心则是距离终端设备最远的计算实体。因此，这是一个垂直分布的分层架构。由此，云计算的时延高于边缘计算与雾计算。另一方面，就弹性和计算能力而言，云计算具有更高的可用性；而与云计算相比，边缘设备和雾设备具有更高的安全性和更强的隐私保护能力。因此，连续体中的各种计算平台可服务于工业运营的不同需求。

2.2.4智能油气中的边缘到云连续体用例

我们研究石油和天然气行业中的无人机管道巡检场景，以了解边缘到云连续体如何满足工业领域的计算需求。设想这样一种场景：4K无人机搭载摄像机每小时可采集数百 GB 的数据。目前的数据分析方法是将海量数据传输到云数据中心；这种做法成本高昂且不切实际，尤其是在需要实时分析时。因此，一个关键问题是：云服务商派出一辆装载拍字节级存储设备的集装箱卡车，这种做法对未来而言是否可行或可扩展？

石油和腾讯分分彩中相同场景的示例（如图2.3所示）以及

图2.3. 基于无人机的检查场景：无人机拍摄图像，并可在边缘计算资源中进行实时分析，而长期分析则在远程云计算设施中进行。



天然气行业的角度来看，同一场景的一个示例（如图2.3所示）是，基于无人机的检查系统可以利用边缘到云连续体进行多阶段价值提取。油气管道可能绵延数千英里，穿越广袤的地域。管道段通常配备模拟仪表和智能传感器，用于测量压力、流量及其他指标。通过使用边缘人工智能监控无人机，拍摄图2.3步骤1中所示的模拟仪表图像，可以分离仪表图像（步骤2），并仅将其中的关键信息传输到下一计算层。在这里，数据预处理（图像分离）是在边缘计算系统中实时完成的。因此，只需处理局部必要的数据，即可在云数据中心获得准确读数（步骤3）。随后，可执行情报的输出将被发送至

现场维护团队解决管道破裂问题。因此，数据预处理、轻量级处理和复杂分析会在边缘到云连续体中进行，以实现高效的基于无人机的管道监测。

2.2.5石油与天然气领域的计算概况

现代计算系统，如边缘、雾或云，使油气行业不同领域的各类容错流程得以顺畅运行。作为一种信息物理系统，油气行业的计算技术栈由以下组件构成：

- 传感器：各种类型的众多传感器（例如，用于测量压力、有毒气体排放量的传感器、安防摄像头等）持续采集多模态数据，形式包括信号、文本、图像、视频和音频。这些数据会被存储或传输，以供离线或在线处理，从而监控油田操作或辅助管理决策。

- 计算机网络：在智能油田中，必须配置短距离和长距离无线及有线计算机网络（例如，蓝牙、公民宽带无线电服务、卫星等），以便设备（例如，传感器、服务器和执行器）进行低时延、高速率通信，满足现场和非现场通信需求。

- 计算系统与中间件：所有收集到的数据最终都必须经过处理才能发挥作用。因此，在后端，智能油田依赖不同形式的计算系统（例如，高性能计算、云、雾、边缘和实时系统）来执行批次或在线数据处理

以用于监控、可视化，以及基于人的或自动化的决策。

- 数据处理与软件技术：智能油田的一条经验法则是："能够处理的数据越多，就越能做出明智的决策"。智能油田持续生成的大量多模态数据（文本、图像、视频和信号）构成了所谓的大数据。必须采用各种算法技术，尤其是机器学习，对这些格式多样的大数据进行处理，从中获取洞见或据此做出明智的决策。

- 执行器：一旦做出决策，就会将其传达给执行器（例如，钻头和压力阀），以执行该决策（例如，增加或降低压力）。

2.3 智能油气：数据与软件方面

2.3.1油气行业中的大数据

石油和天然气行业每天都会生成大量数据，因此需要大规模计算资源和云。油气行业中这类海量数据的三个主要来源如下：

- 烃类储层通常位于地球表面以下 5,000 至 35,000 英尺处。寻找并描述储层的主要手段是高分辨率图像和昂贵的测井记录（在钻出井后）。

- 流体必须穿过复杂的岩层才能到达井筒，而流体本身也很复杂，具有多种不同的物理性质。因此，要了解每口油井的独特特征，并评估采出的流体以进行适当处理，就必须通过安装在油井和钻头上的传感器收集海量数据。

- 石油生产涉及环境与人身安全危害。为防范这些危害，需要在广阔的地理区域部署大量传感器，定期收集数据，从而能够迅速应对任何生态污染排放。

大数据分析有助于实现石油和天然气关键作业的自动化，例如勘探、钻井、生产和输送。上游领域由勘探和钻井组成；由于大数据分析越来越多地用于探测非常规页岩气，因此它是所有其他领域中最主要的数据来源。此外，油价波动使石油和天然气行业变得愈发不稳定。因此，除了工程团队之外，业务团队也越来越多地采用数据驱动策略来预测市场并降低风险。

2.3.2机器学习作为石油与天然气领域的数据驱动应用

智能油气行业是工业4.0革命的一个分支，主要依托人工智能（AI）、物联网以及尖端计算系统（例如，边缘、雾和云计算）。大量且广泛的

相关数据因智能传感器和执行器在油气行业各领域的广泛应用而被采集。这些数据可通过机器学习模型进行评估，从而为该行业和环境带来宝贵的洞见与知识。因此，从广义上讲，人工智能是通过边缘到云计算将传感器生成的数据转化为新颖且有价值的信息和知识的重要工具。

“数据驱动方法”是指一系列技术，可用于整合不同类型的数据、评估不确定性、发现趋势并提取有用信息。基于机器学习和深度神经网络（DNN）模型运行的数据主导型软件应用，例如石油生产控制和应急监控系统 [29, 30, 31, 32, 33]„ 已成为工业4.0革命的基石 [34, 35, 36, 37]。尤其是在偏远地区，这些应用需要实时闭环自动化流程。基于机器学习的解决方案通常采用微服务流程的形式，每个流程可能有一个或多个关键路径，这些路径共同决定整个应用程序的时延 [29]。这些应用程序需要：

- 实时收集大量数据
- 即使无线链路存在不确定性，也能实现感知数据的无缝通信
- 即使面临意外的负载激增（例如紧急情况下），也能在延迟约束下可靠地执行机器学习应用
- Tran 应用程序和资源的透明部署与配置（也称为 o

“无服务器”）。

应对这些需求可能并不容易，尤其是在偏远地区（例如海上油田），那里连接性不稳定，云服务的访问也不稳定。当远程系统必须实时处理海量数据，以管理紧急情况的多个方面（例如溢油）时，这些通信和计算约束就变得至关重要。虽然微型数据中心（也称为雾系统）被用于满足此类远程系统的计算需求，但有时其能力不足以应对负载峰值[38]带来的实时数据传输和处理需求。在下一小节（参见：edgeAi）中，我们将重新探讨资源有限所带来的难题，即如何处理骤增的计算需求。

2.3.3数字孪生：石油与天然气领域的另一种数据驱动应用

“数字孪生”（DT）是指对现有系统进行的计算机仿真。孪生体的输入可由传感器采集现实世界仿真数据后设定。随后，孪生体可通过对物理对象进行仿真，向管理人员提供实时反馈，告知其预测的性能或其他影响。DT 是一种以数据为主导的应用程序，基于机器学习和云计算的可扩展性运行，从而使数据集成的目标更接近现实 [39]。数据对于 DT 系统至关重要，因为许多不同类型的分析、预测和自动化都离不开数据。要生成实用的副本，必须使用高质量、经过验证且有来源的数据。由于 DT 实时运行，所有此前收集的数据

和模型都必须保持准确并及时更新 [40]。DT 系统能够帮助石油和天然气行业的操作人员和管理人员将海量数据转化为洞见，从而预测资产故障并揭示隐藏的营收机会，进而助力实现卓越运营。

2.4 智能油气中的边缘到云人工智能及其他数据驱动应用程序

通过 Modbus、CAN总线、PROFINET 和 MQTT 等异构协议通信的传感器种类繁多， [41]这使得难以构建由连接到传感器的本地设备组成的边缘到云连续体。由于涉及数百个机构和石油钻井平台，实现这一目标本就十分困难。此外，下一代云原生应用需要不同的机器学习（ML）框架、配置和要求。应用程序还需要具备互操作性，才能在处理能力各异的多种设备（例如，中央处理器、几种类型的图形处理器、专用集成电路和现场可编程门阵列）上运行。此外，信息技术运营技术领域的人员、开发者和数据科学家都需要协同工作，管理部署在边缘到云连续体中的物联网应用。因此，边缘到云范围内的主要难题可概括如下：

1. 连接大量物联网设备，以及边缘和云。

2. 与无线通信技术相关的成本。

3. 按需获取高质量计算资源。

4. 无线连接停滞、不稳定或无法运行 .

5. 基于机器学习的应用程序和其他数据驱动应用实现实时运行的需求（例如，数字孪生）。

6. 跨边缘到云系统的数据完整性和隐私。

油气行业的边缘到云连续体问题广泛、复杂，并且不同于传统解决方案。因此，石油专业人员和技术专家是为智能油气行业开发有利可图的环保解决方案的主要推动力量。同时，涉及石油与计算机科学领域交叉点的学术出版物、研究论文和书籍并不常见。因此，缩小对问题空间的认知与提供高效解决方案之间的差距，有助于该行业提高生产力和安全性。

2.5 联邦雾计算及其在远程工业4.0中面临的挑战

本章前面的部分展示了边缘-雾-云连续体的层级架构，其中计算资源沿垂直方向分布。因此，可以设想多个层级的执行平台，其中较高层级（即 云）会带来显著时延，可能不适合处理延迟敏感型任务。为此，我们研究了采用点对点架构、可水平扩展的执行平台——雾系统，以减少时延问题。在工业领域，雾计算系统通常彼此距离较近，有时可作为候选对象，通过点对点方式组建联邦

，以支持紧急情况下的计算需求。例如，多座配有钻井船[42] 的石油钻井平台可以部署在海上烃类储层附近开采石油，并拥有各自的私有雾系统。此外，灾害时期配备移动数据中心的救援船组成部署在灾害区域附近的雾系统，其计算能力可通过组建雾联邦得到增强。因此，可以合理地假设，某些雾系统未得到充分利用，能够处理超出日常请求量的任务。由此，在联邦计算系统之间高效分配资源，可以提升联邦系统的服务质量（QoS）。在一项相关研究中， [43], Xu等人提出了一个面向边缘计算平台的资源分配实例。该方案采用解耦架构，将边缘计算基础设施（ECIs）的基础设施管理与服务提供商（SPs）负责的服务交付和管理分离。此外，作者还提出了一种基于拍卖的资源合约机制和一种延迟感知调度方法，以优化边缘计算系统及服务提供商的效用。因此，通过高效分配资源来联邦化边缘计算系统，可用于应对紧急情况下任务请求激增的问题。然而，要在紧急情况或灾难中建立稳健、高效的边缘计算联盟，还需解决其他若干问题。以下小节将讨论主要挑战。

2.5.1工业*4.0*实时服务

为缩短延迟敏感型服务（例如，传感器数据分析、生产监控）的响应时间，文献从网络时延的角度对雾计算系统进行了研究，

并提出了相关方案。Lorenzo 等人[44]为无线边缘系统提出了一种资源分配模型，通过利用移动设备的闲置资源来缓解网络拥塞。该模型采用物理层、接入层、网络层、应用程序层和业务层的解决方案，以增强网络鲁棒性。这项工作仅考虑网络时延，而未考虑端到端时延。在 [45],Chang 等人提出了一种优化的资源迁移方案，将移动物联网设备中的资源迁移至异构云-雾-边缘计算环境，并考虑了边缘设备资源受限的特点。该方案侧重于进程迁移带来的性能提升，并根据参与系统的运行时间预期分配任务。

2.5.2远程工业*4.0*中的异构雾系统

偏远地区工业中的雾系统可能具有异构性，研究界将异构性分为两种形式：一致异构性和不一致异构性 [46], 。当同类机器具有不同的计算资源时，就会出现一致异构性；当不同类型的机器具有迥异的计算能力时，就会出现不一致异构性。根据异构性，请求的作业可能有不同的执行时间，这会显著影响边缘系统中的任务完成时间。 [47] 讨论了智能油田各个环节（例如上游、中游和下游）中传感器异构数据采集问题，其中 khan 等人提出了一种基于物联网的架构，使数据采集过程更加简单、安全、稳健、可靠且迅速。另有一些研究（例如

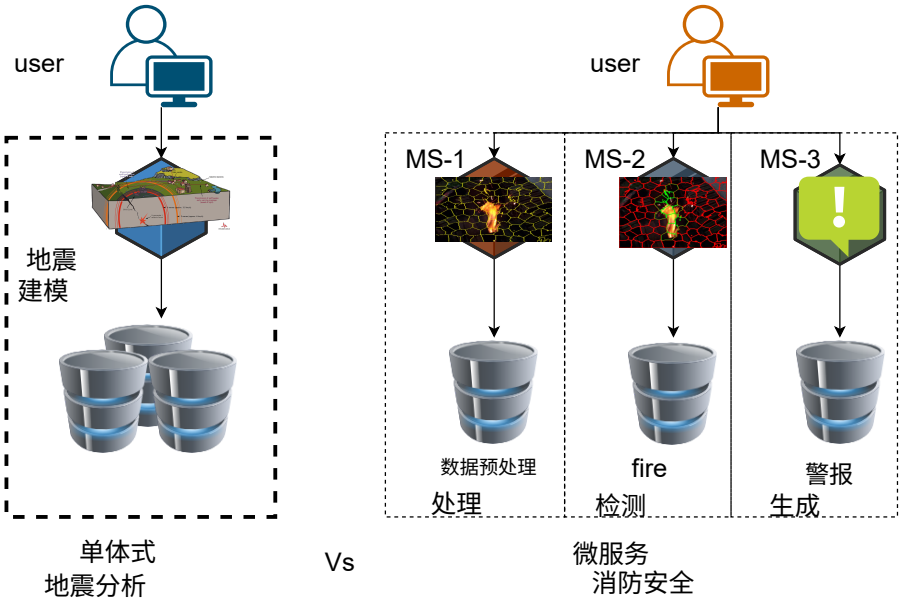
[48, 49, 50]）要么未考虑紧急情况（超额订阅），要么忽略了联邦雾环境中存在的不确定性。在另一项由同一作者开展的相关研究中[51]，研究重点是优化无线网络，而未在雾系统中进行资源分配。因此，同时考虑计算与通信时延对于维持雾联邦的服务质量至关重要。

2.5.3雾系统中的任务完成不确定性

雾联邦中任务完成的主要不确定性受执行和通信延迟的影响。因此，执行不确定性主要指执行所分配任务的计算资源所带来的不确定性，而通信不确定性则主要源于网络系统，尤其是上传和下载时间的不确定性。这两种不确定性都会显著影响雾联邦中的任务完成情况。Xu 等人在一项类似研究中 [52], 提到，由性能退化、服务故障和新增服务所导致的执行不确定性，仍是影响用户服务体验的一大障碍。为克服这种不确定性，本研究提出了一种基于软件定义网络（SDN）的雾计算架构和动态资源配置机制。此外，研究采用快速非支配排序遗传算法 III，对能耗和完成时间这两个目标进行优化，从而生成均衡的调度策略。在另一篇关于资源分配和不确定性的相关论文 [53]中，Li 等人提出了一个多目标优化问题。已有三种并行方法

被开发出来，以提升时延、性能和资源管理水平。首先，研究将排队模型与任务缓冲、卸载和资源分配算法结合起来进行探讨。作者使用李雅普诺夫漂移 [54]设计了资源分配策略。研究结果显示，为提升系统性能，需要在时延和吞吐量之间进行权衡。

图 2.4. 工业4.0智能应用的各种软件架构。地震分析采用单体应用，而消防安全应用则采用微服务架构。



2.5.4工业*4.0*应用的软件架构

工业革命催生了对采用不同软件架构的新兴智能应用的需求，如图 2.4 所示。因此，智能应用由各种微服务组成，这些微服务可以分别部署，且所需的管理工作最少。例如，基于微服务架构的“消防安全”应用包含数据预处理、火灾检测和警报生成，可部署在偏远地区工业中，以确保

现场工作人员免受火灾隐患的威胁。然而，微服务架构的兴起，主要是为了降低大型单体应用的复杂性，这类应用拥有庞大的代码库 [17]。因此，研究界建议将微服务应用的规模维持在适当水平 [55, 56, 57], ，不宜过大，否则会增加应用工作流的管理复杂度。相比之下，对于旧有运营系统，集中式云只能支持对时延容忍度较高的遗留应用[58, 59] 。而现代工业4.0应用对时延敏感，需要动态执行平台来实现智能化并支持快速响应。为此，Rao等人在 [60] 中提出了一种面向智能工业应用的动态运行时，该运行时利用5G技术和边缘云架构。该工作利用应用特定知识，将微服务映射到执行平台。因此，作者仅考虑关键路径的时延，而忽略了各种可能对完成智能解决方案发挥重要作用的通用微服务。此外，该工作考虑使用云数据中心，因此未考虑紧急情况和超额订阅情形。类似地，Faticanti 等人在一项相关研究 [61], 中分析了微服务应用卸载到不同雾系统时的吞吐量需求。该研究探讨了基于容器化微服务模块构建的雾原生应用架构所面临的资源分配挑战。解决方案由两个级联算法构成。第一个算法根据吞吐量对雾应用组件进行分类，第二个算法则负责在地理分布式数据中心之间进行应用编排。

2.6 工业4.0中雾联邦的范围

智能工业中的大量传感器会产生海量数据，但由于缺乏存储和处理能力，这些数据往往无法得到分析 [62]。此外，只有部分数据与分析结果相关。因此，可以在雾联邦中对数据进行预处理，并过滤噪声和异常 [63], ，从而有助于在云数据中心有效训练机器学习模型。

增强现实（AR）和实时视频分析需要快速响应，以及高效、安全的存储系统，而雾联邦能够提供这些支持 [64]。例如，严重的处理延迟可能会使工艺工程师在执行容错性要求极高的工作时产生困惑，进而导致事故。因此，雾计算支持的AR系统能够最大限度地提高处理和传输的吞吐量，并降低时延。相应地，K. Ha，等人在 [65] 设计并实现了一种可穿戴认知辅助系统，该系统以 Google Glass 和 Cloudlet 为后端支持，通过实时场景分析为用户提供社交互动提示。

为确保安全，智慧行业（例如，石油和天然气、交通运输、制造业）部署了大量摄像头传感器，全天候进行监控，以检测任何异常并监测危险区域。因此，捕获的视频需要存储和计算服务，而雾联邦能够提供这些服务。此外，在工业4.0应用中，视频直播流、转码和机器学习处理（例如，目标检测、分类、目标跟踪）也越来越频繁。完成对捕获的视频所需的服务后

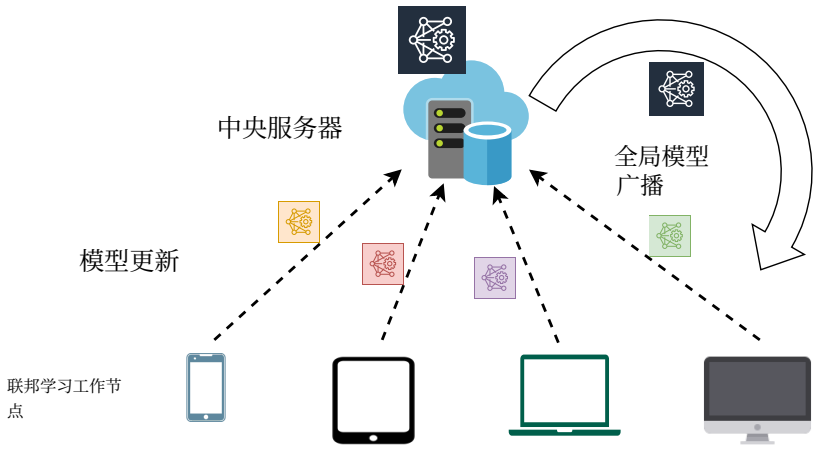
，响应可以通过通知、事件、描述或视频摘要的形式发送给用户。因此，雾联邦有助于对海量视频流进行实时处理（推理）并提供反馈。此外，还能确保低带宽输出数据的可扩展性。再者，也可以在雾侧应用隐私保护技术，以缓解公共监控系统中个人隐私泄露的顾虑。

2.7 联邦雾计算系统的数据隐私方面

智能物联网设备和智能手机的技术进步，提升了利用终端设备执行各种复杂机器学习应用程序（尤其是训练机器学习模型）的可能性。终端设备（例如，手机、个人数字助理、笔记本电脑、可穿戴设备）不断增强的计算和通信能力，使得在雾设备上训练复杂的机器学习模型成为可能。因此，考虑到雾联邦，以分布式方式利用各种雾系统的本地数据进行训练，可以提高机器学习模型的准确性。

不过，数据安全和隐私是这一场景中的主要挑战。因此，联邦学习 [66] 是一种解决方案，它共享机器学习模型，而不是共享不会离开所有者雾系统的数据。在联邦学习中，全局模型会被发送（全局模型广播）到参与工作节点的系统中，以便使用其本地数据进行训练，如图 2.5 所示。经过一段时间的训练后，更新会被发送回中央服务器，以将这些更新整合到全局模型中。随后，更新后的全局模型会再次发送回参与的联邦学习工作节点。该过程会持续进行，直到全局模型达到

图 2.5. 一种典型的联邦学习场景，由联邦学习工作节点和持有全局模型的中央服务器组成。在训练开始时，全局模型会广播给参与工作节点，以便使用各自对应的本地训练数据进行训练。联邦学习工作节点经过一段时间的训练后，会将更新后的模型发送回服务器，以便与全局模型进行整合。



一定的准确性。可以根据全局模型的准确性，采用不同的技术（例如，FedAvg、FedSGD、FedProx）来整合联邦学习工作节点的更新。此外，考虑到联邦学习工作节点在计算和通信能力方面的异构性，更新可能会在不同时间生成。因此，文献中考虑了两种不同类型的联邦学习技术：异步联邦学习和同步联邦学习，分别如此 [67]。鉴于联邦工作节点生成更新所需的时间不同，一些掉队节点需要赶上向服务器发送更新的规定时间。因此，异步联邦学习会尽可能多地整合更新，而同步更新则会丢弃延迟的更新。

2.7.1雾联邦中联邦学习的主要挑战

雾联邦中的联邦学习技术确保机器学习服务

同时保护数据所有者对终端客户的隐私。然而，由于雾设备异构且存在数据异常，仍需解决以下一些重大挑战：

- **训练数据中的类别不平衡问题：** 在联邦学习技术中，各个联邦学习工作节点的本地数据会用于机器学习网络模型训练。因此，部分参与工作节点的本地数据中可能存在类别不平衡问题，进而影响全局模型的鲁棒性。

- **将更新聚合到全局模型的通信成本**

  模型： 为进行联邦学习训练，需要通过互联网将全局模型传输给参与工作节点。训练完成后，更新会被发送回服务器，以便与全局模型同步，最后再将更新后的全局模型发送给联邦学习工作节点。所有传输操作都使用互联网协议，因此可能产生巨大的通信成本。

- **联邦学习工作节点的高效管理：** 参与联邦学习的工作节点数量可能非常庞大，而意外的网络连接问题和异构通信协议会使管理工作几乎无法开展[68]。

2.8 工业4.0中智能解决方案的弊端

硬件和软件技术的进步，使石油和天然气行业发展成为一个完全自动化且依赖机器的行业 [69]。尽管

这场数字革命提高了生产效率，但也可能带来多种漏洞和副作用，进而导致灾难性事件，例如有害气体排放、火灾危险和石油泄漏 [70]。此外，技术的持续进步也增加了信息技术（IT）平台遭到黑客入侵的可能性 [71] ；这些平台处理各种工业数据，并与外部网络通信。另一项关键技术架构是运营技术（OT）平台 [72], ，它主要用于直接开展石油和天然气生产及加工操作，且外部访问受限。因此，IT平台与运营技术（OT）平台之间的连接尤其会给油气作业带来网络威胁。因此，在为石油和天然气行业开发智能技术时，研究人员必须研究智能解决方案的缺点，或充分意识到这些缺点。因此，现有和新推出的智能解决方案都应采用更完善的安全方法，以确保系统的可靠性。此外，智能解决方案可能产生的副作用会妨碍运营效率，甚至适得其反。

因此，有必要探索并识别信息技术和OT平台的各种脆弱区域，以及它们在结构类别中的相互作用方面。因此，应妥善应对网络威胁和设备不兼容性，以识别网络犯罪分子可能利用的各种漏洞。石油和天然气行业面临的问题之一是，它依赖的系统在设计时并未考虑网络连接。例如，工业厂区从未被设计为连接到网络，但随着数字革命不断推进，如今它们已接入网络。这可能导致危险局面，因为针对这类系统的网络攻击可能会

损害运营并造成人员死亡。

工业革命提高了机器人或人类工人操作各类机器的使用率。此外，这些机器有时会与其他机器通信，以完成工业作业。因此，机器与机器、人与机器之间的交互可能出错，从而给网络犯罪分子提供破坏工业流程的机会。由此可见，识别工业交互挑战有助于构建安全可靠的智能解决方案。最后，开发任何实体或软件解决方案都需要人和机器的参与，而这会使各种偏见融入工业4.0的智能解决方案中例如，人工智能、自动化和与人相关的偏见。这些偏见可能导致意外事故，或为网络犯罪分子留下漏洞。因此，解决工业领域中的不同形式的偏见，有助于构建能够抵御网络威胁和攻击的智能解决方案。

2.9 学位论文总结与定位

本节介绍了边缘到云连续体以及雾计算范式联邦及其目标。首先，我们讨论了利用边缘-雾-云连续体支持不同工业4.0应用程序的各种范围，尤其是实时应用程序和基于机器学习（ML）的应用程序。然后，在第3章中，我们分析了各种工业4.0应用程序在广泛使用的AWS云和Chameleon雾服务器上的性能。之后，在第4章中，我们研究了联邦雾系统面临的挑战，并针对偏远工业现场中的单体工作负载，提出了一种跨联邦雾系统的统计资源分配方法

。然后，在第5章中，我们探讨了工业4.0应用程序的微服务软件架构，并提出了一种概率分区和资源分配方法，以提高雾联邦的鲁棒性。之后，在第6章中，我们通过应对石油和天然气行业中隐私保护机器学习应用训练的最新挑战，研究了雾联邦的数据安全和隐私方面。最后，在第7章中，我们识别了智能解决方案的不足，并为偏远石油和天然气行业提出了最新解决方案。最后，我们在第8章中总结了研究发现以及未来值得探索的方向，从而完成了本学位论文。

第3章：云与雾系统中基于深度神经网络的应用性能分析

3.1 概述

本章分析基于深度神经网络（DNN）的工业4.0应用的性能，以研究其在云和雾计算资源上的推理执行时间。作为工业4.0不可或缺的一部分，基于深度神经网络的智能应用会执行延迟敏感型推理，这类推理需要具备较高的准确性，并在特定截止期限内满足某些应用约束。即使推理准确率很高，只要错过了某个应用的截止期限，服务质量（QoS）仍可能受到影响。由于云和雾计算环境固有的多租户特性和资源异构性，基于深度神经网络的应用的推理时间具有随机性。如果未能捕捉这种随机性，可能会给石油和天然气行业等关键领域带来灾难。为了增强工业4.0的稳健性，解决方案架构师和研究人员需要了解基于深度神经网络的应用的行为，并捕捉其推理时间中存在的随机性。因此，本研究对热门云平台 Amazon 和作为雾系统的 Chameleon 中的推理时间进行了描述性分析。

我们采用两种统计方法评估基于深度神经网络的应用程序：以应用为中心和以资源为中心。首先，我们进行以应用为中心的分析，对分别在 Amazon 和

Chameleon 上运行的四种类别各异的深度神经网络应用程序的推理执行时间进行统计建模。其次，我们采用以资源为中心的方法，考察异构云雾系统中一种称为每秒百万条指令（MIPS）的速率指标。随后，我们使用刀切法和自助重采样等非参数建模方法，估算异构云雾系统中 MIPS 的置信区间。本研究的发现或有助于学者和云解决方案架构师构建稳健的解决方案，以应对云中推理时间的随机性，使他们能够在避免意外后果的同时，为用户提供更高的服务质量。此外，我们还提供了一个基于深度神经网络的应用程序基准测试一个，供系统架构师用于构建有效的资源分配方案。

3.2 工业4.0中的基于深度神经网络的应用

在工业4.0中使用的各种基于深度神经网络的应用中，我们选取了智能油气行业使用的四种不同应用。这些应用的总结如表3.1所示，其中列出了每种应用的缩写名称、其深度神经网络（网络）模型、输入数据类型、在油气行业中的部署范围 [73], 以及用于构建模型的代码库。为便于复现，这些应用的代码库、输入数据和分析结果均已在前文提到的GitHub代码库中公开。在本节的其余部分，我们将探讨每种应用类型的特征。

3.2.1火灾检测

火灾检测应用程序是监控的重要组成部分

[a]https://github.com/hpcclab/Benchmarking-DNN-应用程序-工业4.0

表 3.1. 油气工业4.0中使用的基于深度神经网络的应用程序及其网络模型、输入数据类型、使用范围和代码库。

应用类型	深度神经网络模型	输入类型	范围	代码库
火灾检测 *(Fire)*	定制版 AlexNet	视频片段	控制与监控	TensorFlow (tflearn)
人体活动识别（*HAR*）	定制序列神经网络	运动传感器	工作节点安全	Keras
溢油检测（*Oil*）	FCN-8	SAR图像	灾难管理	Keras
声阻抗估计 *(AIE)*	时序卷积网络	地震数据	地震勘探	PyTorch

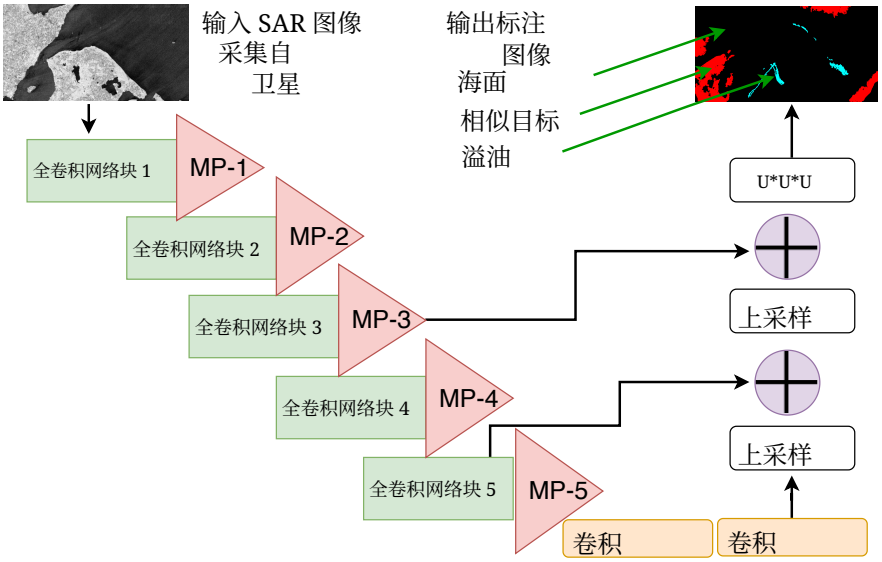
系统旨在为工业4.0提供安全性和韧性。我们使用卷积神经网络 (CNN) 研究了 Dunnings 和 Breckon 提出的基于深度神经网络的火灾检测应用程序 [74]。它能够实时识别受监控视频各帧中的火灾区域（像素）。在作者提供的多个火灾检测模型中，我们使用 FireNet 模型，它能够在给定视频片段的每一帧中正确识别并定位火灾。FireNet 是 AlexNet 模型的简化版本 [75], ，包含三个卷积层，尺寸分别为 64、128 和 256。为了从前一层提取具有显著响应的高频特征，该模型的每个卷积层都配有一个最大池化层和一个局部响应归一化层。我们创建了一个包含 240 个视频、背景各异的基准数据集，以考察火灾检测应用程序的推理时间。为确保评估公平、准确，所有视频均按两秒时长处理。

3.2.2人体活动识别

人体活动识别（HAR）系统广泛应用于工业4.0，以确保工作节点在危险区域的安全。在人体活动识别系统中，各种传感器

会分析人工工作者在进行各种身体运动时由不同传感器采集的数据。在这种情况下，我们使用手持式 PDA 设备上普遍配备的运动传感器（如加速度计和陀螺仪）来采集人体活动相关传感器数据。我们使用的人体活动识别系统采用包含四层的序列神经网络模型，以识别工作节点的活动（步行、上楼、下楼、坐姿）。在分析中，我们使用 UCI 机器学习库中的一个数据集，即基于智能手机的人体活动识别 [76]。

图 3.1. FCN-8 模型的框图由 5 个全卷积网络模块和 2 个上采样模块组成。该模型接收 SAR 图像作为输入，并执行逐像素分类，输出一幅标注图像。

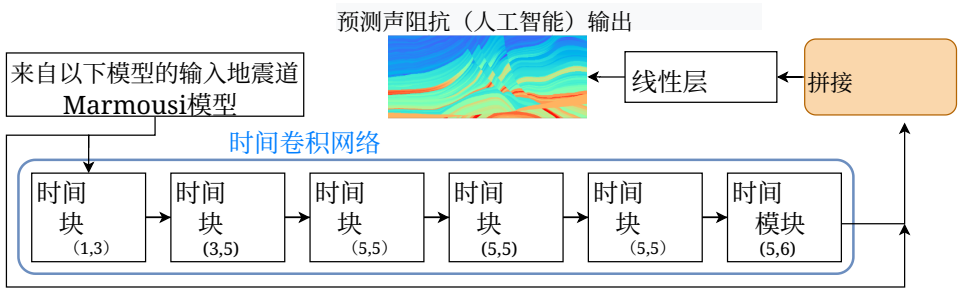


3.2.3溢油检测

检测溢油对于实现安全、清洁的油气工业4.0至关重要。基于深度神经网络的石油泄漏检测系统的准确性令人期待 [77]。我们采用一种基于 FCN-8 的检测系统

模型 [78], ，如图 3.1 所示。正如我们所见，该模型包含五个全卷积网络（FCN）模块和两个上采样模块，它们共同对输入图像进行语义分割（即，对每个像素进行分类），并输出一幅标注图像。FCN-8 模型基于卫星（又称 SAR） [79] 图像运行。我们对分析进行配置，以获取 MKLab 收集的 110 幅 SAR 图像的推理时间 [77]。

图 3.2。 时间卷积网络（TCN）模型的示意图，该模型由六个时间模块、输入数据以及以预测的声阻抗形式呈现的输出组成。



3.2.4声阻抗估计

从地震数据估计声阻抗（AI）是油气勘探中的重要步骤。为了从地震数据估计声阻抗，我们采用一种基于时间卷积网络的解决方案函数 [80], ，如图 3.2 所示。该解决方案在避免梯度消失和过拟合方面优于其他方案。我们使用 Marmousi 2 数据集 [81]来估计声阻抗。

3.3 工业 4.0 计算平台

3.3.1亚马逊云

AWS 是云计算行业的先驱，在大量分布式数据

中心提供 175 多种服务，包括亚马逊 EC2。亚马逊 EC2 提供异构程度不一致的机器（例如，中央处理器、图形处理器和 Inferentia），并以各种虚拟机实例类型（例如，通用型、计算优化型和机器学习（ML））的形式提供。在每种虚拟机类型中，还提供一系列虚拟机配置（例如， `large`、 `xlarge`、 `2xlarge`），以体现该虚拟机类型内部的一致异构性。为了了解机器异构性对各种应用程序推理时间的影响，我们从所提供的每种机器类型中选择一种有代表性的虚拟机类型。表 3.2 展示了机器类型及其核心数量和内存规格。我们注意到，所有机器类型都使用 SSD 存储单元。虽然通用型机器被认为不适合延迟敏感型基于深度神经网络的应用，但我们研究它们的原因在于，它们与雾计算平台中常用机器类型的规格相似。因此，考虑这些机器类型（以及 Chameleon 云中的类似机器 $m1.small$ ）使本研究的结果适用于采用雾计算来支持延迟敏感型应用的情形 [83]。

表 3.2. Amazon EC2 中用于对基于深度神经网络的应用程序进行性能建模的异构机器类型和虚拟机配置。在此表中，机器学习优化型表示 AWS 提供的 Inferentia 机器类型。

机器类型	**虚拟机配置。虚拟 CPU**	**图形处理器**	**内存（GB）**
内存优化型	`r5d.xlarge`	4	0	32
机器学习优化型	`inf1.xlarge`	4	0	8
GPU	`g4dn.xlarge`	4	1	16
通用型	`m5ad.xlarge`	4	0	16
计算优化型	`c5d.xlarge`	4	0	8

表 3.3。 Chameleon 云中的各种虚拟机配置经过配置，用于表示一个始终保持异构性的系统。

虚拟机配置。	虚拟 CPU	内存（GB）
m1.xlarge	8	16
m1.large	4	8
m1.medium	2	4
m1.small	1	2

3.3.2作为雾计算系统的 *Chameleon*

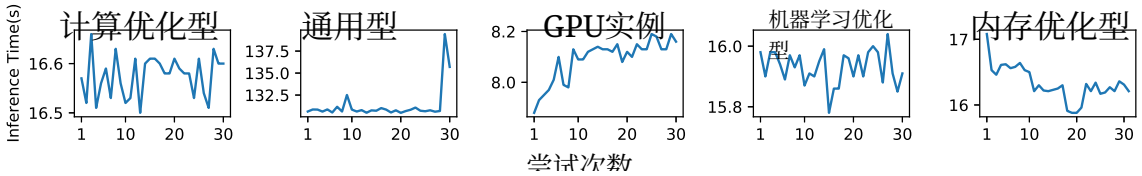
Chameleon [84] 是由美国国家科学基金会（NSF）维护的大规模公共计算平台，通常用于学术研究。由于 Chameleon 的维护问题（例如，瞬时故障、意外停机、资源匮乏）、较少的大规模虚拟机配置以及其分布式特性，我们将 Chameleon 视为雾计算系统。Chameleon 支持基于虚拟机的异构计算，并提供灵活性来管理虚拟机实例的计算、内存和存储容量。在本研究中，我们使用 Chameleon 配置一组始终保持异构性的机器（雾系统）。我们配置了四种虚拟机类型，即 `m1.xlarge`、 `m1.large`、`m1.medium` 和 `m1.small`，具体如表 3.3 所示。需要注意的是，Chameleon 提供的虚拟机使用机械硬盘作为存储设备。

3.4 性能建模的环境设置

本研究聚焦于工业4.0中广泛使用的延迟敏感型基于深度神经网络的应用。因此，我们考虑一个动态（在线）系统，该系统已加载上一节所述的预训练的基于深度神经网络的应用，并在相关应用上执行到达的请求。这

意味着我们测量所考虑应用程序的热启动推理时间 [85] 。这些基于深度神经网络的应用程序大多使用 TensorFlow，AWS 和 Chameleon 中的虚拟机均配置为在 Ubuntu 18.04 上运行该框架。

图3.3。 溢油检测应用在 AWS 的异构虚拟机上运行时，推理执行时间的随机性。对于每个虚拟机实例，溢油检测应用均执行 30 次，并将这些执行作为横轴上的尝试次数绘出。纵轴表示每次尝试的推理时间。



我们在 AWS 中进行的初步评估（如图3.3所示）表明，在不同的尝试中，同一机器类型上的应用程序（溢油检测）的推理执行时间可能具有很强的随机性。在 Chameleon 云上也发现了类似的随机性：我们在同一虚拟机实例中运行溢油检测应用 30 次。因此，为了捕捉由瞬时故障或多租户等多种因素导致的这种随机性（又称一致异构性）， [86, 87],我们基于同一虚拟机中同一请求的 30 次执行进行分析。

3.5 以应用为中心的推理时间分析

3.5.1 概述

在本部分中，我们记录四种深度神经网络应用的推理时间，并尝试使用各种统计方法识别其潜在的统计分布。随后，为了用单一指标描述推理执行时间的表现，我们探究这些分布的集中趋势。

3.5.2推理执行时间的统计分布

在各种统计方法中，正态性检验被广泛用于了解收集样本的分布。因此，我们首先使用夏皮罗-威尔克检验 [88] 来验证每种机器类型上的推理时间分布是否符合正态分布。接下来，我们采用柯尔莫哥洛夫-斯米尔诺夫检验 [89] ，根据抽样得到的推理执行时间确定拟合效果最佳的分布。

3.5.2.1 用于验证抽样数据正态性的夏皮罗-威尔克检验。零假设认为推理执行时间服从正态分布。为了了解随机样本是否来自正态分布，我们进行夏皮罗-威尔克检验。该检验的结果记为 W；其值较低（低于 w_α 阈值）表示抽样数据不服从正态分布，反之亦然。

使用 W 进行显著性检验（即，计算 P 值）。P 值越高，尤其是高于阈值（通常为 0.05）时，越能支持零假设，即抽样数据服从正态分布。

表3.4。使用夏皮罗-威尔克检验得出的 AWS 云中机器上基于深度神经网络的应用程序的执行时间分布。

AWS云中使用夏皮罗-威尔克检验的执行时间分布					
应用类型	Mem. Opt.	ML Opt.	GPU	Gen. Pur.	Compt. Opt.
Fire	非高斯分布 (P=$2.73e^{-16}$)	非高斯分布 (P=$5.42e^{-16}$)	非高斯分布 (P=$6.59e^{-16}$)	非高斯分布 (P=$2.06e^{-13}$)	非高斯分布 (P=$3.9e^{-16}$)
HAR	非高斯分布 (P=$7.12e^{-8}$)	非高斯分布 (P=$1.04e^{-8}$)	高斯分布 (P=0.19)	非高斯分布 (P=$1.76e^{-8}$)	非高斯分布 (P=$0.4.62e^{-5}$)
Oil	非高斯分布 (P=$8e^{-4}$)	非高斯分布 (P=$2.9e^{-16}$)	非高斯分布 (P=0.012)	非高斯分布 (P=$1.27e^{-16}$)	非高斯分布 (P=$5.86e^{-14}$)
AIE	高斯分布 (P=0.46)	高斯分布 (P=0.23)	高斯分布 (P=0.08)	非高斯分布 (P=$1.99e^{-10}$)	高斯分布 (P=0.96)

AWS 收集到的推理时间的夏皮罗-威尔克检验结果

表3.5。 使用夏皮罗-威尔克检验得出的 Chameleon 云中深度神经网络应用的执行时间分布。

Chemeleon 中通过夏皮罗-威尔克检验分析的执行时间分布				
应用类型	`m1.xlarge`	`m1.large`	`m1.medium`	`m1.small`
Fire	非高斯分布 (P=$4.05e^{-5}$)	非高斯分布 (P=$1.e^{-4}$)	非高斯分布 (P=$7.58e^{-6}$)	非高斯分布 (P=$1.32e^{-7}$)
HAR	高斯分布 (P=0.74)	非高斯分布 (P=0.02)	高斯分布 (P=0.18)	高斯分布 (P=0.84)
Oil	非高斯分布 (P=0.01)	非高斯分布 (P=$5.5e^{-7}$)	非高斯分布 (P=0.01)	N/A
AIE	非高斯分布 (P=$2.77e^{-10}$)	非高斯分布 (P= $3.46e^{-6}$)	非高斯分布 (P= $1.4e^{-4}$)	非高斯分布 (P=$2.46e^{-6}$)

如表3.4所示，其中各列列出不同的机器类型，各行则定义应用类型。该表表明，我们最初的假设并非完全成立。表3.5展示了 Chameleon 云的夏皮罗-威尔克检验结果，结果显示，在所有案例中，只有三个案例符合正态性假设。鉴于若干案例不符合正态分布，下一节将采用柯尔莫哥洛夫-斯米尔诺夫检验，更细致地探究各应用程序推理时间的最佳拟合分布，并使用另一种统计方法对先前获得的结果进行交叉验证。

3.5.2.2 使用柯尔莫哥洛夫-斯米尔诺夫检验识别执行时间分布。柯尔莫哥洛夫-斯米尔诺夫检验的零假设是：给定机器类型上的某种应用类型，其推理时间服从某种统计分布。柯尔莫哥洛夫-斯米尔诺夫拟合优度检验（又称*K-S* 检验）用于判断来自某个总体的一组样本是否符合特定分布。具体而言，该检验测量已知的假设概率分布与根据观测推理时间生成的分布（又称经验分布

函数（EDF））之间的最大垂直距离（称为检验统计量 ）。然后，如果 D 大于从 K-S 检验 P值表中获得的临界值，则拒绝零假设。

表 3.6. 使用柯尔莫哥洛夫-斯米尔诺夫检验分析 AWS云机器上基于深度神经网络的应用程序的推理时间分布。

AWS云中使用柯尔莫哥洛夫-斯米尔诺夫检验的执行时间分布					
应用类型	Mem. Opt.	ML Opt.	GPU	Gen. Pur.	Compt. Opt.
Fire	无分布。	无分布。	无分布。	无分布。	无分布。
HAR	学生 t 分布 (P=0.08)	学生 t 分布 (P=0.77)	学生 t 分布 (P=0.99)	学生 t 分布 (P=0.57)	学生 t 分布 (P=0.95)
Oil	学生 t 分布 (P=0.44)	学生 t 分布 (P=0.96)	学生 t 分布 (P=0.5)	学生 t 分布 (P=0.20)	指数分布 (P=0.21)
AIE	正态分布 (P=0.99)	正态分布 (P=0.54)	正态分布 (P=0.47)	指数分布 (P=0.16)	正态分布 （P=0.99）

表 3.7。 使用 K-S 检验对 Chameleon 机器上的基于深度神经网络的应用进行推理时间分布分析。

Chameleon 中使用柯尔莫哥洛夫-斯米尔诺夫检验进行的执行时间分布分析				
应用类型	m1.xlarge	m1.large	m1.medium	m1.small
Fire	无分布	无分布	无分布	对数正态分布
HAR	正态分布 (P=0.98)	学生 t 分布 (P=0.88)	正态分布 (P=0.66)	正态分布 (P=0.96)
Oil	对数正态分布 (P=0.36)	对数正态分布 (P=0.99)	对数正态分布 (P=0.81)	N/A
AIE	学生 t 分布 (P= 0.47)	学生 t 分布 (P=0.12)	学生 t 分布 (P=0.25)	学生 t 分布 (P=0.83)

AWS 和 Chameleon 云中观测推理时间的 K-S 检验结果分别列于表 3.6 和表 3.7。从表 3.6 可以看出，在 AWS 中，大多数条目对应正态分布或具有相似特性的学生 t 分布。然而，我们观察到，火灾检测应用的推理时间不符合任何一种具有可接受 P 值的特定分布。我们还观察到，计算优化型机器上的溢油检测应用和通用型机器上的 AIE 应用的推理时间都符合指数分布。不过，这两种情况下较低的 P 值表明，零假设仅得到较弱的支持。

相反，表 3.7 显示，在 Chameleon 云中，对数正态分布（即，随机变量的对数服从正态分布）和学生 t 分布占主导地位。分析执行轨迹后发现，Chameleon 中的推理时间普遍长于 AWS 中的推理时间，因此呈现右偏特性。因此，这种分布趋向于对数正态分布。与 AWS 中的观察结果一致，我们发现火灾检测应用在大多数情况下不符合任何分布。进一步分析表明，缺乏明确分布的原因在于输入视频的多样性（例如，帧率和分辨率）。当我们降低输入视频的自由度，并将其限定为配置相同（帧率相同）的视频时，发现推理时间符合对数正态分布。这一观察结果表明，输入数据的特征和变化会显著影响推理时间的统计行为（如重点见解中所述）。最后需要指出的是，由于内存有限，溢油检测应用无法在 `m1.small`机器上运行。

洞见： 我们对识别

推理时间分布所开展的分析得出的关键洞见如下：

- 针对 AWS 和 Chameleon 的夏皮罗-威尔克检验拒绝了以下原假设：基于深度神经网络的应用的推理时间服从正态分布。
- K-S 检验反映出，在 AWS（表 3.6）和 Chameleon（表 3.7）中，推理时间均以学生 t 分布为主。
- 输入数据的各种配置对推理时间的统计分布起着决定性作用。

表3.8。 AWS 上观测推理时间的集中趋势指标（μ）和数据离散度指标（σ）的测量。

AWS 上推理执行时间的均值和标准差（毫秒）					
应用类型	Mem. Opt.	ML Opt.	GPU	Gen. Pur.	Compt. Opt.
Fire	μ=1461.8 σ=457.3	μ=1281.7 σ=387.93	μ=1349.5 σ=418.9	μ =1534.8 σ=494.7	μ=1421.4 σ=441.8
HAR	μ=1.27 σ=0.082	μ=0.66 σ=0.006	μ=0.51 σ=0.006	μ =1.17 σ=0.042	μ=0.66 σ=0.003
Oil	μ=269.9 σ=1.01	μ=218.8 σ=0.66	μ=65.98 σ=0.47	μ=667.1 σ=2.26	μ=242.9 σ=0.68
AIE	μ=7.02 σ=0.02	μ=6.41 σ=0.03	μ=7.55 σ=0.04	μ=9.35 σ=0.06	μ=7.95 σ=0.02

表 3.9。 Chameleon 云中推理时间的集中趋势指标（μ）和数据离散度指标（σ）。

Chameleon 中推理执行时间（毫秒）的均值和标准差				
应用类型	m1.xlarge	m1.large	m1.medium	m1.small
Fire	μ=2155.20 σ=725.48	μ=2213.30 σ=731.50	μ=2330.80 σ=742.20	μ=3184.80 σ=1033.30
HAR	μ=22.14 σ=0.76	μ=47.69 σ=1.26	μ=49.24 σ=0.57	μ=52.69 σ=0.78
Oil	μ=147.16 σ=5.23	μ=222.22 σ=2.89	μ=412.78 σ=4.99	N/A
AIE	μ=6.23 σ=0.25	μ=6.23 σ=0.15	μ=6.40 σ=0.13	μ=7.72 σ=0.24

3.5.3集中趋势和离散程度度量分析

本部分利用推理时间的统计分布，探讨其集中趋势指标，即用单个数值概括所收集观测值的行为。此外，为了分析这些观测值的统计离散度，我们测量了推理时间的标准差。具体而言，我们估算了推理时间的算术平均值和标准差。AWS 和 Chameleon 系统的推理时间集中趋势指标分别列于表 3.8 和表 3.9。**关键洞见**如下：

- 机器学习优化型和图形处理器实例通常优于其他 AWS 机器类型。

- 在这两个云平台上，与其他应用程序相比，Fire 和 Oil 的推理时间标准差更高。这种高度不确定性归因于其输入数据的特性；Oil Spill 的输入图像存在类别不平衡 [77], ，而 Fire 的输入视频则变化很大。

- 在具有一致异构性的 Chameleon 虚拟机中，采用密集网络模型的深度神经网络应用（例如，Oil 和 Fire）可以利用性能强大的机器（例如， `m1.xlarge`），显著缩短推理时间。
- 总体而言，AWS 的推理时间比 Chameleon 更短。这是因为 AWS 使用固态硬盘，而 Chameleon 使用机械硬盘。此外，我们发现 Chameleon 更容易发生瞬时故障，从而拖慢应用程序的运行。

3.6 以资源为中心的推理时间分析

在计算系统的性能分析中，基于速率的指标[90]定义为将已执行的计算机指令数量归一化到标准时间单位。MIPS 是一种常用的基于速率的指标，可用于比较两个或多个计算系统的计算速度。鉴于计算系统（例如，AWS 机器学习优化型和图形处理器）越来越多地为机器学习应用使用指令级设施，本部分旨在分析不同机器类型处理基于深度神经网络的应用时的性能。这项分析的结果可能对致力于为工业4.0应用场景开发定制化资源分配方案的研究人员和云解决方案架构师特别有参考价值。对于基于速率的指标，我们不假设任何分布 [91], ，而是采用非参数方法。除 MIPS 外，我们还以每种情况下的置信区间（CI）形式给出 MIPS 的范围。

表 3.10. AWS 中异构机器针对各类基于深度神经网络的应用程序的每秒百万条指令值。

AWS云中深度神经网络应用的每秒百万条指令					
应用类型	Mem. Opt.	ML Opt.	GPU	Gen. Pur.	Compt. Opt.
Fire	1938.63	2196.35	2092.72	1862.04	1989.56
HAR	838640.65	1595874.34	2040057.33	891754.48	1581709.12
Oil	164.54	168.58	331.98	20.46	162.01
AIE	145.58	180.28	150.25	131.25	160.32

表 3.11. Chameleon 云中异构机器针对各类基于深度神经网络的应用程序的每秒百万条指令值。

Chameleon 中深度神经网络应用的每秒百万条指令				
应用类型	m1.xlarge	m1.large	m1.medium	m1.small
Fire	1327.81	1282.33	1249.63	871.36
HAR	91.78	102.51	124.76	136.62
Oil	18267.35	11233.41	6243.94	N/A
AIE	246366.52	249551.29	236300.93	201807.49

设应用程序 i 包含 n_i 条指令，其在机器 m上的推理时间为 t_{im}。那么，机器 m 执行该应用程序的 MIPS 定义为$MIPS_{mi} = n_i/(t_{im} \times 10^6)$。因此，在计算任何机器的 MIPS 之前，我们需要估算每个基于深度神经网络的应用程序的指令数（n）。为此，我们在一台 MIPS 已知的机器上执行每个任务 t_i，并估算n_i。然后，对于每台机器 m，我们测量 t_{im}，随后计算 $MIPS_{mi}$。表 3.10 和表 3.11 分别展示了 AWS 和 Chameleon 的 MIPS 值。

为了测量每种机器类型中每种应用类型的 MIPS 置信区间（CI），我们采用非参数统计方法 [91]，该方法基于样本数据进行预测，而不对其底层分布作任何假设。由于我们处理的是基于速率的指标，因此采用调和平均数；与算术平均值相比，它能更准确地分析此类指标。我们使用刀切法 [91] 这一重抽样方法，并通过

Bootstrap [91], 这一另一种广为人知的重抽样方法进行验证。这两种方法都采用调和平均数来测量 MIPS 的置信区间。

表 3.12. 采用刀切重采样法得到的 AWS 机器上基于深度神经网络的应用的每秒百万条指令值的置信区间。

AWS 云中采用刀切法计算的每秒百万条指令置信区间					
应用类型	`Mem. Opt.`	`ML Opt.`	`GPU`	`Gen. Pur.`	`Compt. Opt.`
Fire	[1549.42, 1975.65]	[1770.81, 2243.04]	[1671.78, 2131.66]	[1465.31, 1889.77]	[1594.78, 2028.36]
HAR	[812040.26, 856355.96]	[1592214.75, 1599426.64]	[2033084.47, 2046727.57]	[880417.69, 901345.49]	[1580275.10, 1585598.85]
Oil	[163.55, 165.47]	[168.36, 168.81]	[330.68, 333.22]	[20.35, 20.57]	[161.86, 162.17]
AIE	[139.02, 141.04]	[155.56, 156.01]	[141.57, 142.03]	[118.06, 119.82]	[148.35, 149.00]

表 3.13. 采用刀切重抽样法得到的 Chameleon 机器上不同深度神经网络应用程序的每秒百万条指令值置信区间。

Chameleon 云中使用刀切法计算的每秒百万条指令置信区间				
应用类型	`m1.xlarge`	`m1.large`	`m1.medium`	`m1.small`
Fire	[1032.11, 1341.75]	[1010.62, 1303.02]	[964.76, 1259.68]	[670.82, 872.85]
HAR	[88.27, 94.20]	[99.84, 104.49]	[122.33, 126.67]	[135.13, 137.92]
Oil	[18083.59, 18628.64]	[11159.71, 11662.41]	[6139.59, 6262.15]	N/A
AIE	[237710.12, 252686.82]	[247166.73, 251673.68]	[168804.58, 268273.11]	[199676.71, 203681.17]

3.6.1使用刀切法估计置信区间

设 p 为观测推理时间的数量。刀切法通过 p 次迭代计算调和平均数，每次迭代都剔除一个样本。也就是说，每次都会创建一个大小为 $p-1$ 的新样本（重抽样）。设 x_j 为第 j 个观测推理时间。那么，重抽样 i 的调和平均数称为伪调和平均值（记为 y_i），并根据公式 3.1 计算。

$$y_i = \frac{p-1}{\sum_{j=1, j\neq i}^{p} \frac{1}{x_j}} \tag{3.1}$$

接下来，计算 p 个伪调和平均值的算术平均值（记为 $\bar{y}$），并用它来估计标准差。最后，使用 t 分布表计算置信水平为 95% 时的置信区间边界。AWS 机器的刀切法结果如表 3.12 所示，与表 3.10 中的 MIPS 计算结果一致。同样，表 3.13 所示的 Chameleon 云刀切法分析结果，验证了表 3.11 中先前的 MIPS 计算结果。然而，在下一部分，我们将使用自助法对这些结果进行交叉验证。

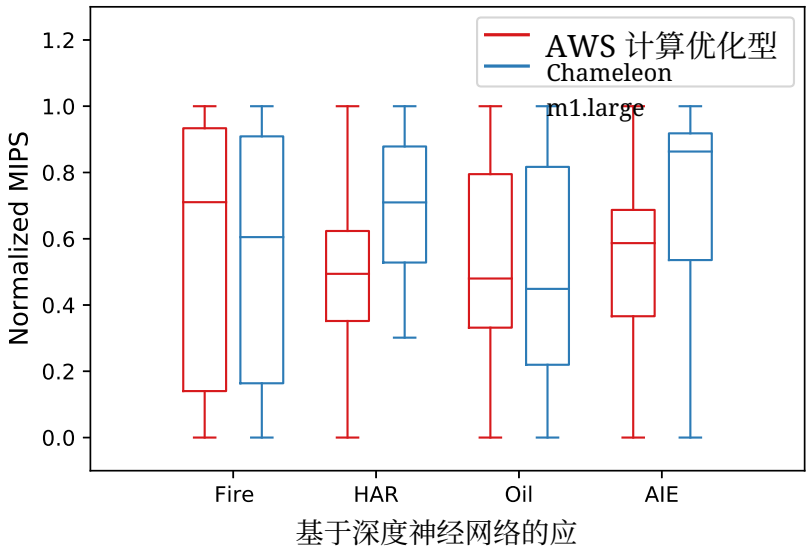
3.6.2使用自助法估计置信区间

Bootstrap 使用有放回抽样技术[91] 对观测推理时间反复进行随机抽样。随机抽样是指以非零概率选取一个样本，重采样数据的数量（表示为 k）取决于用户的考量。重采样后，计算 k 个样本的调和平均值，并按升序排列，以估计置信区间。最后，对于指定的置信水平，从排序后的样本中选取第（$\alpha/2 \times k$）个值和第（（$1-\alpha/2$）$\times k$）个值，作为置信区间的下界和上界。我们将 k 值设为 100，并将 α 设为 0.05，以获得 95% 的置信水平。

对于 AWS 和 Chameleon，使用自助法进行置信区间分析所得的结果与刀切法估算的范围相似，从而验证了这些范围。因此，由于篇幅有限，我们不报告自助法的 MIPS 值表。不过，我们注意到，无论

应用类型和云平台如何，自助法给出的置信区间范围都更短（<style id='1'>即</style>不确定性更小）。范围较短的原因是，自助法会按照用户指定的样本数量进行重抽样，而该数量可能大于原始样本量。

图 3.4。 AWS 和 Chameleon 机器在各种基于深度神经网络的应用中的 MIPS 值比较分析。为便于展示，MIPS 值已在 $[0,1]$ 之间归一化。



为了对 MIPS 值进行跨平台分析，在图 3.4 中，我们比较了 AWS `Compute Optimized` 与 `m1.large`（即 Chameleon 中一种兼容的机器类型；参见表 3.2 和表 3.3）的 MIPS 值范围。该图的横轴表示不同的应用类型，纵轴表示 MIPS 值，并为便于展示，基于最小 - 最大缩放将其归一化到 $[0,1]$, 范围内。由于输入视频存在较大差异，我们观察到，在两个云平台上，Fire 火灾检测的置信区间范围都较宽。然而，对于 HAR、Oil Spill 和 AIE 应用，我们观察到第一和第三

四分位数的置信区间范围在 Chameleon（其机器更容易发生瞬时故障 [92]）中大于 AWS。如此宽的范围表明，除了输入数据的变化之外，底层资源的可靠性也是推理时间随机性的决定性因素。

3.7 总结与讨论

准确估计延迟敏感型基于深度神经网络的应用的推理时间，对于工业4.0的鲁棒性和安全至关重要。准确的估计使云服务提供商和解决方案架构师能够设计资源分配和负载均衡方案，以应对基于深度神经网络的应用执行时间中存在的不确定性。本研究针对部署在两个计算平台（AWS 和 Chameleon）的异构机器上的若干基于深度神经网络的应用的推理时间不确定性，开展了以应用和资源为中心的分析。首先，我们使用夏皮罗-威尔克检验，验证推理时间是否符合正态分布的假设。结果发现，推理时间往往不符合正态分布。因此，在第二部分中，我们扩大了分布检验的范围，并使用柯尔莫哥洛夫-斯米尔诺夫检验验证各个情形下的潜在分布。分析表明，在这两个计算平台上，推理时间通常符合学生 t 分布。然而，在 Chameleon 系统的若干情形中，我们观察到对数正态分布，并将其归因于该平台上虚拟机性能的不确定性。接下来，为开展以资源为中心的分析，我们对每种应用类型的异构机器的 MIPS（作为基于速率的性能

指标）进行了建模。在分析中，我们采用适用于基于速率指标的非参数方法，并结合调和平均数使用刀切法和自助重采样方法，确定各情形下 MIPS 值置信区间的范围。计算得到的 MIPS 值及其置信区间范围反映了不同基于深度神经网络的应用在云与雾系统的各种机器类型上的行为。对 AWS 和 Chameleon 的置信区间范围进行比较分析后发现，推理时间的不确定性源于输入数据的差异以及底层平台的不可靠性。未来，我们计划将本研究的发现用于设计物联网和边缘计算系统中的准确资源分配方法。此外，我们计划开展预测分析，以确定每个推理任务到达后的执行情况。

第4章：联邦雾计算在偏远工业现场管理单体工作负载的优势

4.1 概述

在前几章中，我们的初步研究发现，对于具有随机执行行为的远程智慧行业而言，雾联邦可以成为工业4.0应用的潜在计算平台。因此，工业4.0应用的随机执行行为会影响任务完成时间。在这种情况下，能够感知工业4.0应用随机执行行为的高效资源分配与负载均衡技术，可通过确保接收任务按时完成来保证系统的鲁棒性。因此，在本章中，我们首先 战略性地开发一种负载均衡方法，用于将到达任务分配给雾联邦。因此，我们的首要目标是确保系统（雾联邦）的鲁棒性，即满足到达任务的截止期限。为实现这一目标，我们估计雾系统中接收任务的端到端时延，并利用该时延预测任务在雾联邦中的完成时间。因此，我们提出一种用于各雾系统负载均衡器的概率任务分配方法，该方法能够感知接收任务的延迟约束。接着，在第二 部分，我们使用 EdgeCloudSim [18]模拟器的合成工作负载（针对工业任务工作负载定制）评估所提出的负载均衡方法。

4.2 联邦雾系统中的端到端时延

当任务请求到达雾系统的负载均衡器时，通信延迟与计算延迟共同构成端到端时延。此外，多种因素会影响这些延迟，导致其呈现随机性。因此，在雾计算系统中计算端到端时延并捕捉其随机特性颇为困难。在接下来的章节中，我们将介绍影响通信延迟和处理延迟的因素。此外，我们还将介绍一种在考虑端到端时延随机特性的同时对其进行估计的模型。

4.2.1估算通信延迟

处理任务请求并返回响应所需的时间称为通信延迟。更具体地说，通信延迟由传输时延和传播时延造成。任务 t （类型为 i）在任意两点 m 和 n （例如，雾联邦中的两个雾系统）之间的传输时延记为 $\Theta_i(m,n)$，定义为上行传输时延（记为 $\tau_u(m,n,i)$ ）与下行传输时延（记为 $\tau_d(m,n,i)$ ）之和。也就是说，我们有 $\Theta_i(m,n) = \tau_u(m,n,i) + \tau_d(m,n,i)$。设 $I_u(i)$ 为传感器最初采集的数据载荷大小（以比特为单位），该数据载荷作为任务类型 i 的输入。请注意，对于某些传感器（例如摄像头），每次传感器读数中采集的数据大小可能存在随机性。此外，设 $R_u(m,n)$ 表示传输数据所经过的上行带宽。 T 表示将每个数据包传输到上行信道所需的时间（称为传输时间间隔（TTI））。接下来，

上行时延根据公式 4.1 计算。

$$\tau_u(m,n,i) = \lceil \frac{I_u(i)}{R_u(m,n)\cdot T} \rceil \tag{4.1}$$

同样，下行时延定义如公式 4.2 所示。

$$\tau_d(m,n,i) = \lceil \frac{I_d(i)}{R_d(m,n)\cdot T} \rceil \tag{4.2}$$

总带宽为 W 的正交频分复用 (OFDM) 被平均划分为一组 k 个子信道（其中 $k \in K$），每个子信道的带宽为 w。因此，下行带宽根据公式 4.3 定义。

$$R_d(m,n) = w\cdot \sum_{k\in K} y_{mnk} \log_2(1+\gamma_d(m,n,k)) \tag{4.3}$$

其中，如果分配了子信道 k，则 $y_{mnk} = 1$；否则为 $y_{mnk} = 0$。由于无线通信容易受到联邦中其他雾系统产生的噪声和干扰，因此 $R_d(m,\ n)$ 的值也取决于下行信号与噪声加干扰比（也称为 $SINR$ [93]）。信号与噪声加干扰比定义为特定信号的功率除以干扰功率（来自所有其他干扰信号）与背景噪声功率之和。需要指出的是，上行传输时延的计算细节（$\tau_u(m,\ n,\ i)$）与下行时延的计算方法相似。

在雾联邦中，由于距离较近，雾系统之间的传播时延可以忽略不计。相比之下，雾系统与云数据中心之间通常通过卫星进行通信，因此会产生显著的传播时延 [94]。传播时延记为 τ_p，其计算依据

方程式 4.4。

$$\tau_p = 2 \cdot \frac{d(n, st)}{S_l} \tag{4.4}$$

在方程式 4.4 中， $d(n,\ st)$ 表示雾 n 到卫星 st 之间的距离， S_l 表示介质或链路中的传播速度。要计算往返时间中的传播时延，应将该分数值加倍。得知传播时延后，可根据方程式 4.5 计算访问云数据中心的总体通信延迟，记为 d_{comm}。

$$d_{comm} = \Theta_i(m, n) + \tau_p \tag{4.5}$$

正如我们所注意到的，多个因素共同构成了具有随机行为的通信延迟。为刻画这种随机行为，我们将通信延迟视为随机变量，并使用统计分布对其建模。也就是说，我们使用基于历史通信信息 [46] 构建的概率密度函数（PDF），表示任意两点之间的通信延迟（例如，联邦中的两个雾系统）。根据中心极限定理，通信延迟可以用正态分布建模。

4.2.2估计计算延迟

负载均衡器将到达任务请求 t 分配给雾系统后，任务必须先在该雾系统的调度队列中等待，然后才能执行。对于给定的类型为 i的任务 t ，记作 t_i，其完成时间（即计算延迟）受队列中的等待时间（排队时延）以及任务在所分配雾系统的机器上的执行时间（执行时延）影响。

重要的是，这两个因素都是随机的，因此任务完成时间也呈现出随机行为。

任务 t_i 的排队时延取决于雾系统中排在它前面的任务数量及其执行时间。执行时间的随机性可能源于不同的任务类型，以及不同雾系统中机器的特性。即使是在同一雾系统的同构机器上执行相同任务类型的任务，其执行时间也具有随机性。这可能是由于待处理数据量的变化，以及雾系统 [95]中的任务多租户所致。机器故障等其他因素也可能导致任务执行时间具有随机性。

为捕捉计算延迟的随机性，我们将每个雾系统上每种任务类型的任务完成时间视为随机变量。随后，我们使用统计分布对计算延迟进行建模。也就是说，我们基于每个雾系统上每种任务类型的历史完成时间信息，使用概率密度函数（PDF）对计算延迟进行建模。根据中心极限定理，每个雾系统上每种任务类型的计算延迟可以使用正态分布进行建模。

4.2.3估计端到端延迟

一旦估算出通信延迟和计算延迟，它们的复合时延便构成端到端时延。更具体地说，复合时延可通过将通信延迟的概率密度函数与计算延迟的概率密度函数进行卷积来获得。对于到达负载均衡器的任务 t_i ，令 N_i

表示该任务到联邦中另一个雾系统的通信延迟的概率密度函数。此外，令 M_i 表示 t_i 在另一个雾系统中的计算延迟的概率密度函数。那么， t_i 的端到端时延记为 E_i，计算如下： $E_i = N_i \circledast M_i$。

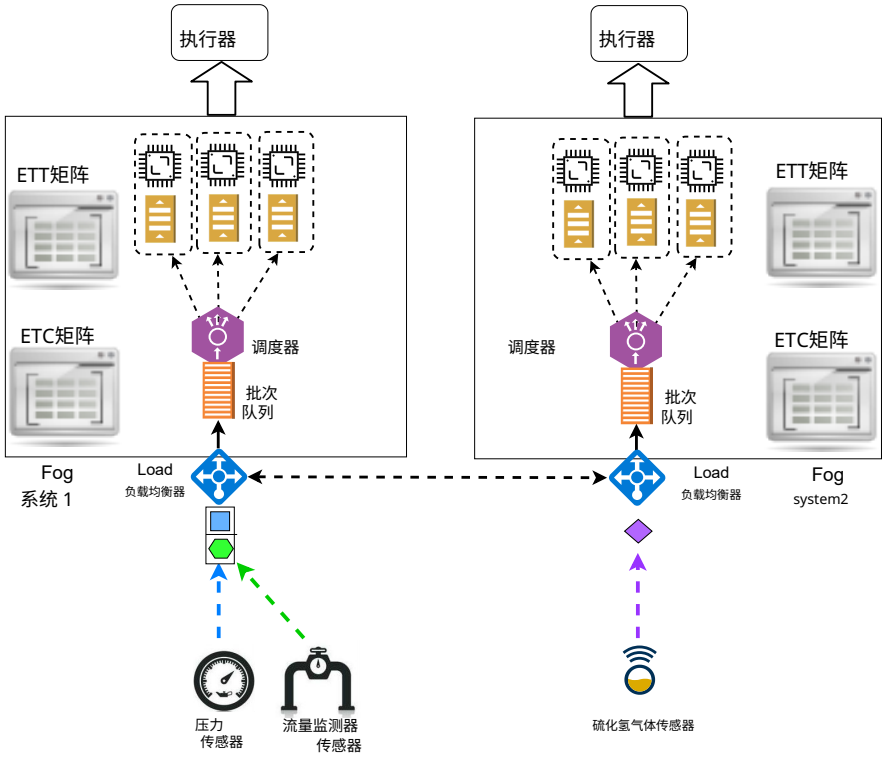
4.3 联邦雾计算系统中的鲁棒资源分配

图4.1展示了所提出的联邦雾计算系统资源分配模型的概要。该资源分配模型采用负载均衡器模块，这是实现雾联邦的关键。每个雾系统都配备一个负载均衡器，它会针对每个到达任务，确定最有可能在截止期限前完成该任务的雾系统（接收雾节点或邻近雾系统）。

负载均衡器的功能对于应对任务到达的不确定性（例如，灾害时期）并增强雾系统对此类不确定性的鲁棒性尤为重要。负载均衡器以即时模式 [96]运行，并在到达任务抵达时立即将其分配给适当的雾系统。适当性根据最有可能使任务在截止期限前完成的雾系统来确定（即成功概率）。对于每个邻近雾系统，都可以利用在该系统上执行任务 t_i 的端到端时延分布，计算截止期限为 δ_i 的任务 t_i 的成功概率。为避免反复重新分配任务和产生复合时延，我们规定，一旦作出任务分配决策，该任务便不能重新分配。

每个雾系统的资源分配利用历史

图4.1。 一种配备负载均衡器模块、可促进雾联邦的雾系统。传感器生成的任务请求由负载均衡器模块接收，并分配给最有可能成功完成该任务的雾系统。



计算延迟和通信延迟信息来构建其分布的概率密度函数。为此，每个负载均衡器维护两个矩阵，即估计任务完成时间（ETC） [97] 和估计任务传输时间（ETT），用于跟踪每种任务类型在每个邻近雾系统上的计算延迟和通信延迟。条目 $ETC(i,\ j)$ 保存任务类型 i 在雾系统 j 上的计算延迟概率密度函数。同样，条目 $ETT(i,\ j)$ 保存任务类型 i 到达雾系统 j 的通信延迟概率密度函数。ETC和ETT矩阵的条目会定期以离线方式更新，且不会干扰负载均衡器的实时运行。

到达任务 t_i 时，接收雾节点的负载均衡器可以利用 $ETC(i,\ j)$ 和 $ETT(i,\ j)$，计算在任一邻近雾节点 j 上执行 t_i 的端到端时延分布。端到端分布可用于获取在任一雾系统上于截止期限前完成 t_i 的成功概率，记为 $p_j(t_i)$。我们有：$p_j(t_i) = \mathrm{P}(E_i \leq \delta_i)$。需要注意的是，在接收雾节点上计算任务 t_i 的概率不会产生额外的通信延迟。因此，对于接收雾节点 r，我们有：$p_r(t_i) = \mathrm{P}(M_i \leq \delta_i)$。下一步，选择成功概率最高的雾系统作为分配任务 t_i 的合适目标节点。这意味着，只有在考虑通信延迟后邻近雾系统仍能提供更高成功概率时，才会将任务 t_i 分配给邻近雾系统。

值得注意的是，邻近雾节点上的成功概率可能仅略高于接收雾节点，差异并不显著。实际上，只有当邻近雾系统提供的成功概率明显更高时，才应将任务分配给邻近雾节点。为了判断概率差异是否显著，我们利用底层端到端分布的置信区间（CI），据此计算接收雾节点和远端雾节点的成功概率。具体而言，对于给定任务，只有当邻近雾节点端到端分布的置信区间与接收雾节点端到端分布的置信区间不重叠时，我们才认为邻近雾节点提供的成功概率显著更高。

算法1中提供的伪代码描述了鲁棒任务

算法1： 负载均衡器的任务分配算法。

输入： 任务 t_i； ETC和 ETT矩阵； G（邻近雾节点集合系统）

输出： 选定的雾节点 $j \in G$ 以分配 t_i

1 p_r（t_i） ← 接收雾节点 r上的成功概率
2 **对于** 每个雾系统 $j \in G$ **执行**
3 $p_j(t_i)$ ← 邻近雾上的成功概率 j
4 **如果** $p_j(t_i) > p_r(t_i)$ **则**
5 将 $p_j(t_i)$添加到 P，作为分配的候选雾
6 **end**
7 **结束**
8 按降序排列集合 P 中的元素
9 将接收雾节点 r 视为 t_i的默认分配
10 **对每个** $p_j \in P$ **执行**
11 **如果** E_j 的置信区间与 N_r 的置信区间不重叠**则**
12 选择雾 j 作为目的地，并将 t_i 分配给它
13 退出循环
14 **end**
15 **结束**

分配启发式算法，负载均衡器利用该算法充分发挥联邦雾系统的优势，提高系统的鲁棒性。该启发式算法称为 最大鲁棒性 ***(MR)***，在新任务 t_i 到达雾系统的负载均衡器时调用。算法根据到达任务的截止期限（δ_i），首先计算 t_i 在接收雾节点及其相邻雾系统上的成功概率（算法1中的步骤1-7）。然后，在步骤8中，按成功概率降序排列。如果任务在接收雾节点上的成功概率更高，则将该任务分配给接收雾系统（步骤9）。否则，将成功概率最高的邻近雾节点的端到端时延分布的置信区间与接收雾节点的置信区间进行比较。如果置信区间不重叠，则将任务 t_i 分配给邻近雾节点（步骤12）。否则，

对接收雾系统的其余邻居执行相同的过程。如果未找到不重叠邻居，则将任务 t_i 分配给接收雾系统（步骤 9 中的默认分配）。

4.4 联邦雾计算的性能评估

我们使用了 EdgeCloudSim [18], ，它是一种用于性能评估的离散事件模拟器。我们模拟了五个雾系统（微型数据中心），每个系统具有八个核心和 $[1500, 2500]$ 每秒百万条指令（每秒百万条指令）的计算能力。每个雾系统的核心均为同构；但是，不同雾系统具有不同的每秒百万条指令数，这体现了雾系统之间的异构性。我们还考虑了一个具有 40,000 每秒百万条指令的云数据中心，用于处理非紧急任务。每个雾中的任务按照先到先服务的方式进行映射。访问云的带宽基于卫星通信，设置为 200 Mbps，传播时延为 0.57 秒 [98]。

在每次工作负载试验中，为模拟智能油田的负载，我们假设一半任务为紧急任务，另一半为非紧急任务。每个任务都属于某种特定类型，该类型代表其服务类型。在每次工作负载试验中，紧急任务由两种不同的任务类型实例化，非紧急任务则由另外两种任务类型实例化。从某种特定类型实例化的每个任务，其执行时间均从代表该任务类型的正态分布中抽样。我们假设每个任务均为顺序执行（需要一个核心），并以每秒百万条指令（MIPS）模拟其执行时间。我们使用泊松分布（不同任务类型的均值各不相同）生成任务的

任务间到达率，并模拟超额订阅时段内的任务到达情况。每次工作负载试验中的任务数量有所不同，以代表不同的超额订阅级别。

工作负载试验中任务 i 的截止期限按以下方式生成： $\delta_i = arr_i + \beta \cdot avg^i_{comp} + \alpha \cdot avg^i_{comm} + \epsilon$，其中 arr_i 是任务到达时间， avg^i_{comp} 是该任务类型在各雾系统中的平均计算延迟， avg^i_{comm} 是平均通信延迟。 β 和 α 分别是表示计算和通信不确定性的系数， ϵ 则是系统中其他不确定性的松弛量。我们考虑在每个雾系统中维护 ETC 和 ETT 矩阵，并在工作负载执行每达到 10% 时更新一次。根据系统模型，这些矩阵的条目视为正态分布。为确保准确性，每项实验均重复进行 30 次，并报告结果的均值和 95% 置信区间。

4.4.1负载均衡器的基准任务分配启发式方法

最小预期完成时间（*MECT*）：此启发式算法 [46] 使用 ETC 矩阵计算到达任务在每个雾系统上的平均预期完成时间，并选择预期完成时间最短的雾系统。

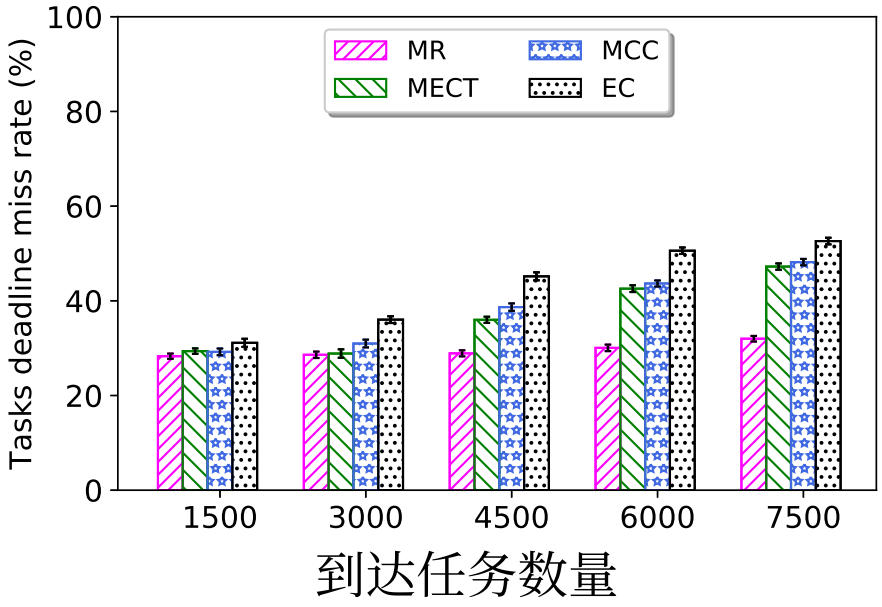
最大计算确定性（*MCC*）：该启发式算法（用于 [99]）利用 ETC矩阵计算任务的截止期限与平均完成时间之间的差值（称为确定性）。随后，将任务分配给确定性最高的雾节点。

边缘云（*EC*）：该启发式算法基于传统的雾计算模型运行，不考虑联邦。具体而言，紧急任务分配给接收雾节点，非紧急任务分配给云数据中心。

4.4.2实验结果

4.4.2.1 分析超额订阅的影响。衡量智能油田中超额订阅雾系统鲁棒性的主要指标是任务的截止期限错失率。在本实验中，我们通过增加传感器生成的任务数量（即超额订阅级别）来研究系统性能。图4.2展示了在采用不同任务分配启发式算法时，到达任务数量（横轴，从1,500到7,500）变化对截止期限错失率（纵轴）的影响。

图4.2. 使用负载均衡器中的不同任务分配启发式算法时，超额订阅级别（到达任务数量）增加对截止期限错失率的影响。



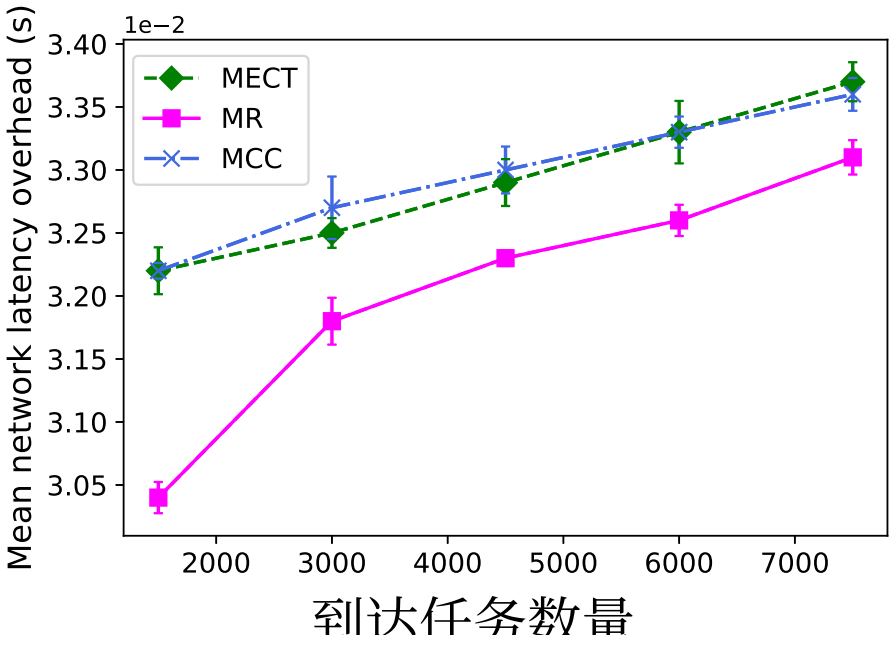
从图4.2可以看出，随着任务数量增加，所有启发式算法的截止期限错失率都会上升。在超额订阅级别较低时（1,500个任务），MR、MECT和MCC的表现相近。然而，当系统的超额订阅程度加剧时（4,500个任务），差异便十分显著。任务数量达到7,500时，MR的截止期限错失率比MECT和MCC低约16%，比EC低约21%。这是因为MR能够捕捉端到端时延，并且只有在联邦能显著提高成功概率时，才会主动利用联邦。相比之下，EC不考虑联邦，而其他基准启发式算法只考虑计算延迟。由此可见，考虑端到端时延并捕捉其潜在的不确定性，能够显著提高鲁棒性，尤其是在系统超额订阅时（例如，处于灾害时期）。

4.4.2.2 分析雾联邦的通信开销。

尽管我们在上一个实验中表明，采用联邦可以提高系统鲁棒性，但我们尚不清楚联邦环境中的任务分配所产生的通信开销。因此，在本实验中，我们评估应用不同任务分配启发式算法所带来的通信时延。具体而言，我们测量每个任务所产生的平均通信延迟开销（图4.3中的纵轴），并考察不同到达任务数量下的情况（图4.3中的横轴）。

图4.3显示，MECT 和 MCC 会导致更高的平均通信延迟。原因是这些启发式算法没有考虑通信

图4.3。 不同启发式算法在雾联邦中为每个任务引入的平均通信延迟开销。

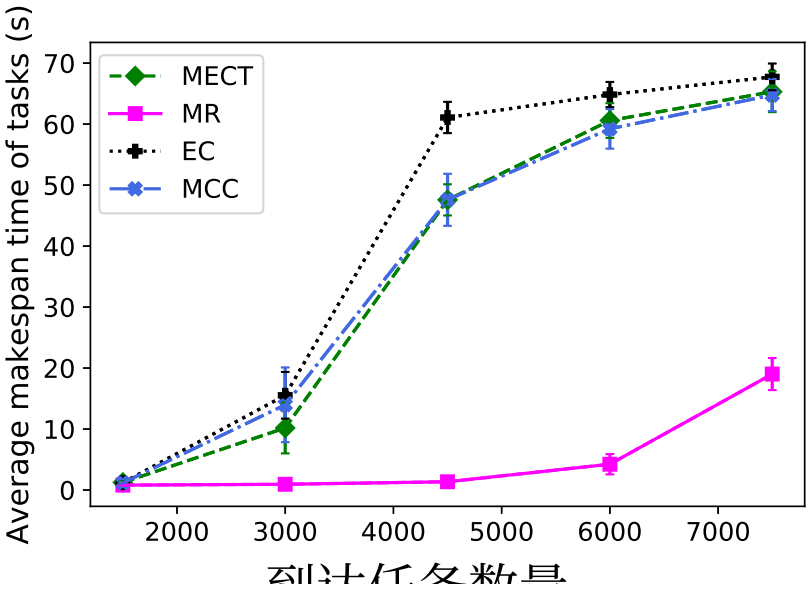


时延，并且会激进地将任务重定向到同一个雾系统，导致特定网络链路（接收雾系统与被重定向至的雾系统之间）发生拥塞。相比之下，MR 会考虑通信时延，并且只有在成功概率显著提高时，才会较为保守地重定向任务。

4.4.2.3 分析任务的平均完工时间。 不同的任务分配启发式算法会导致任务产生不同的计算延迟。为了了解计算延迟，我们测量了应用各种任务分配启发式算法后任务的平均完工时间。

图4.4表明，EC 导致平均完工时间最长。这是因为 EC 不使用联邦，导致接收雾系统超额订阅，而其他相邻雾系统则未得到充分利用。

图4.4。 使用不同任务分配启发式算法时任务的平均完工时间（秒）。



因此，当接收雾节点达到 3,000 个任务的饱和状态后，平均完工时间会迅速上升。MECT 和 MCC 未考虑任务完成时间的随机性，因此可能会将到达任务分配给同一个雾节点，导致该节点超额订阅。因此，任务的平均完工时间会上升。相比之下，MR 会考虑端到端时延的随机性，并计算相邻雾系统上的成功概率。此外，只有当相邻雾系统能够提供显著更高的成功概率时，MR 才会将任务分配给该系统。因此，与其他启发式算法相比，MR 的平均完工时间最低。

4.5 总结

本章探讨了在灾难情境下，雾联邦在智能工业（石油和天然气）中的可用性。为了满足紧急情况下的计算需求，将各种任务分配到合适的雾系统中

颇具挑战，因为雾系统之间存在异构性。因此，若系统不采用高效的负载均衡技术，要确保系统在每项实时紧急任务的截止期限方面保持鲁棒性可能十分困难。为此，我们提出了在邻近工业设施中部署的雾计算系统动态联邦。在联邦环境中，我们捕捉了两种不确定性来源，即通信和计算；否则，它们会对实时服务造成不利影响。每个雾系统中的负载均衡器模块负责实现联邦，该模块能够感知雾系统之间的端到端时延，并捕捉其中的随机性。负载均衡器利用这种感知能力，找出能够显著提高每项到达任务成功概率的雾系统。实验结果表明，我们提出的联邦系统能够增强雾计算系统抵御到达时间、通信和计算延迟方面不确定性的鲁棒性。我们得出结论，对于超额订阅程度较高的情况，负载均衡器尤其有用（提升幅度最高可达 27%）。即使在联邦中采用朴素的负载均衡方法（MCC 和 MECT），性能也能提升约 13%。

第5章：将远程工业4.0智能微服务应用适配到联邦雾计算系统

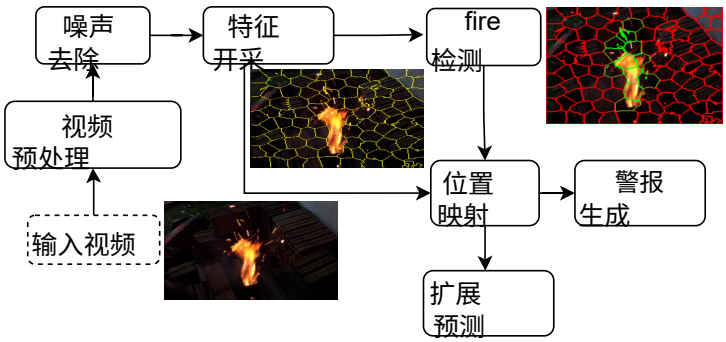
5.1 概述

物联网技术与智能应用的发展推动着工业4.0 [71] 革命。各行各业（例如，制造业、食品加工、石油和天然气）都部署了各种智能传感器、执行器和智能设备，以控制运营技术平台 [100, 101]。因此，工业运营中使用的传感器每天都会产生大量数据[102]。石油和天然气行业就是一个例子，它会产生海量传感器数据，因此有必要在靠近数据源的位置进行处理。例如，典型的海上石油钻井平台每天会产生 1 至 2 TB 的数据 [103]。这些数据大多会输入高级计算应用（例如，机器学习、报告生成、自动化），以便作出智能的延迟敏感型决策，从而提高能源效率、产量和安全措施。例如，工作场所空气质量估算 [104] 等保障工作节点安全的应用会使用环境传感器，测量工作场所周围可呼吸空气的质量（即，空气中有害颗粒物的存在情况）。因此，必须快速估算空气质量，以避免潜在事故的发生。

在偏远海上工业中，一些服务（例如，数据采集、警报生成、目标跟踪）对于需要同步执行的复杂操作或安全相关操作至关重要。当任何意外紧急情况导致有限时间内需要完成的计算活动大幅增加时，情况可能会进一步恶化。

在这种情况下，我们关注的是智能石油和天然气行业。该行业因勘探、钻井和生产运营中的工业流程复杂且无法容忍故障，曾遭遇各种灾难和灾害（例如，深水地平线（2010年）[19],Usumacinta自升式平台灾难（2007年） [20], 孟买高北灾难（2005年） [21], Ocean Ranger灾难（1982年） [105] ）。因此，偏远海上工业需要时延感知支持 [106] ，但由于工业作业地点位于偏远地区，典型的云数据中心无法提供这种支持。目前的解决方案采用不稳定的卫星通信[107] 将数据发送到内陆云数据中心，导致服务质量（QoS）下降，并增加工业安全风险。因此，面临的高层次挑战是缺乏计算资源，无法支持偏远地区工业中超额订阅的情况。

图5.1. 基于微服务的工作流结构以框图形式呈现。完成消防安全应用程序需要处理每个微服务。



5.1.1面向工业*4.0*的智能微服务应用

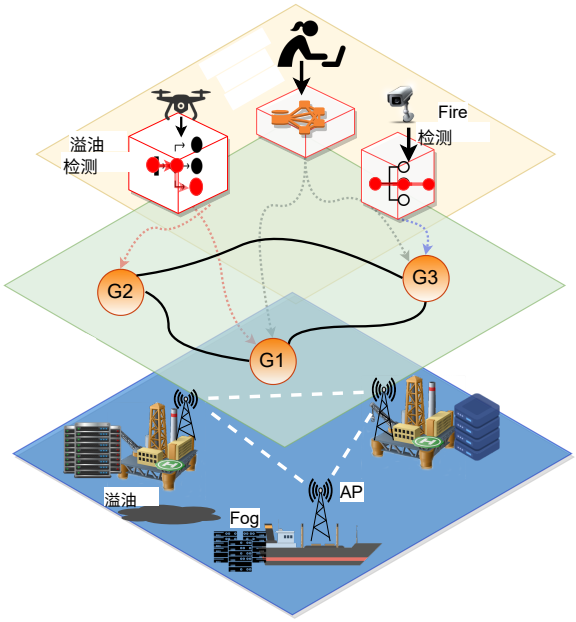
工业4.0智能应用通常遵循现代软件架构[16, 108] ， 其中各种微服务 [17] 需要按顺序执行。然而，

微服务可以使用自动化部署流程分别进行部署，只需极少的管理工作，可以使用各种编程语言和数据存储技术进行开发，并且每个微服务都可以独立更新、修改和扩展。因此，我们重点关注偏远地区工业中经常使用的微服务应用。例如，如图5.1所示，“消防安全”应用可以包含用于采集视频监控数据、对采集的视频进行预处理、噪声去除、特征提取、火灾检测、位置映射、警报生成和扩展预测的微服务。相比之下，许多工业此前已经部署了具有不灵活软件架构的遗留应用 [109] 。因此，执行平台应同时支持单体遗留应用和现代微服务，以确保工业安全和容错运行。然而，现代工业4.0应用由微服务组成，这给执行平台带来了新的挑战。在这种架构下，应用程序的时延约束取决于底层软件架构所定义的微服务完成时间。因此，为了开发稳健的工业4.0执行平台，系统架构师需要了解接收应用程序的软件架构。

5.1.2面向工业*4.0*微服务应用的联邦雾系统

新兴的工业物联网和通信技术的发展，使计算资源更加接近数据源和终端设备。例如，如今，偏远地区工业中的雾计算系统 [110]通常

图5.2。 海上油气行业拥有雾联邦基础设施，可支持智能微服务应用。



执行工业计算流程，以实现顺畅生产和工作场所安全。然而，如图5.2所示，可以从第4章中将联邦雾平台概念化为通过无线网关连接而成的系统，这些无线网关表示为 G_i。因此，具有异构延迟约束的各种应用程序需要联邦雾计算系统提供计算支持。相应地，联邦应充分考虑通信与计算不确定性，以及应用程序的软件结构和延迟要求。因此，需要针对单体式和微服务软件结构制定应用执行计划，同时考虑随机执行时间，以及源自执行平台和通信技术的不确定性。

因此，在我们此前发表于第4章的工作中 [38] ， 资源

分配方法针对单体应用得到了深入探讨。此外，考虑到由各种微服务执行的复杂操作流程，主要问题之一是确保整个应用工作流在称为截止期限的时限内完成。因此，了解应用工作流的最佳划分点至关重要，以便按时完成。相应地，需要回答的问题是：“如何在雾联邦中分配工业4.0应用程序（例如，单体式应用、微服务应用），才能使应用工作流在规定的时间范围内完成？”因此，从执行智能微服务应用的系统管理角度来看，还需提出两个问题：1）如何划分微服务工作流，才能满足其截止期限约束？2）如何在雾联邦中分配划分后的微服务，才能最大限度地提高其按时完成的可能性？

我们之前的工作 [38] 表明，联合附近的资源是解决石油和天然气等偏远行业的边缘计算系统所面临的资源限制（即 超额订阅）的一种方法。此外，我们还探讨了智能软件架构（又称微服务工作流）带来的新挑战。因此，为了应对整个海上石油和天然气行业面临的困难，我们提出了一种面向联邦雾系统中工业4.0应用程序的资源配置方法，该方法同时考虑软件架构和底层执行平台的结构。此外，该方案还满足微服务工作流的截止期限限制，从而提高执行平台的可靠性。

因此，我们的方法分为两个阶段：理解接收应用程序的软件架构，以及为这些应用程序的成功完成分配计算资源。因此，本研究的贡献如下：

- 提出一种概率分区方法，能够感知工业4.0应用底层的软件架构。

- 提出一种统计资源分配启发式算法，同时考虑应用程序的时间约束。

- 对分区技术以及跨雾联邦的资源分配进行了广泛评估。

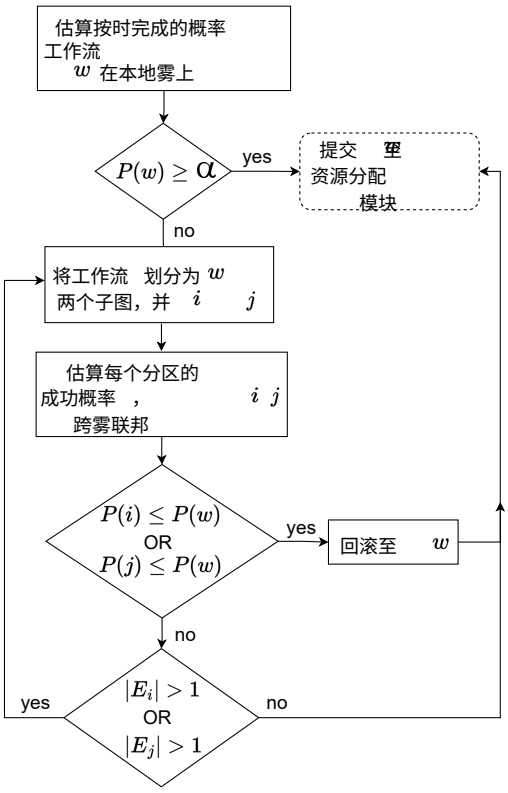
所提出的解决方案可作为基础，让系统架构师或行业研究人员据此构建更完善的解决方案，以应对需求高峰时偏远离岸行业的需求。此外，该解决方案兼容单体遗留应用，有助于传统行业向工业4.0转型并适应其带来的变化。

5.2 微服务应用工作流的分区方法

要维持由多个微服务组成的智能应用的延迟约束，取决于底层软件架构和计算资源映射。例如，将微服务应用部署到单个雾系统中执行可能不可行，也可能无法满足其截止期限约束。

另一方面，单体应用无法进行分区，可视为仅包含一个微服务的应用。对于微服务软件架构，将应用划分为多个分区并将其分配到雾联邦中的不同位置，可以提高其在延迟约束内完成的可能性。此外，为分区后的微服务分配适当的计算资源，也能确保整个应用工作流顺利完成。

图5.3。 工作流分区方法的流程图。已分区工作流被发送至资源分配模块，在该流程图中以终止框表示。



分区方法的主要目标是以一种能够使应用程序满足截止期限的方式对微服务应用进行分区。因此，我们

将具有微服务架构的应用程序视为一组以某种方式相互连接、构成图 $G = (V,\ E)$ 的微服务，其中顶点集合 $V = (m_1,\ m_2,\ ..m_n)$ 表示微服务，边 $e(m_i,\ m_j) \in E$ 表示微服务 m_i 与 m_j之间的通信。作为分区方法的第一步，我们考虑在不进行分区的情况下，将整个微服务工作流在本地雾系统中执行。因此，分区方法会估算工作流 w 在本地雾中按时完成的概率，这对应于流程图 5.3 中的第一个处理框。为估算整个应用工作流 w的截止期限，我们将各个微服务的截止期限相加，可定义为 $\delta_w = m_1^\delta + m_2^\delta + ... + m_n^\delta$。因此，每个微服务都有一个负载均衡器预先知晓的截止期限（m_i^δ）。此外，对于每种微服务类型，我们都有计算延迟分布（m_i^d），用于表示雾联邦中的执行时间。因此，为了估算整个 w的成功概率，我们对应用程序中各微服务的计算延迟分布进行卷积，可定义为

$$D_w = m_1^d \circledast m_2^d \circledast ....m_n^d \tag{5.1}$$

最后，使用卷积分布 D_A，我们按如下方式衡量成功概率，

$$P(w) = \mathbb{P}(D_A \leq \delta_A) \tag{5.2}$$

将公式 5.2 的输出与条件变量 α 进行比较，如流程图 5.3 的第一个条件所示。我们选择平均成功率（即 50%）作为 α

我们的实验评估场景。当完成工作流的可能性低于 α时，分区服务会使用最小割 [111]图划分算法执行，该算法会在图5.3 中将工作流分区 w 为两个子图i 和 j 流程框。因此，考虑到应用程序中的动作流，我们在所提出的解决方案中采用了广泛使用的图论定理之一——最大流最小割e [112]。

由于寻找最小分区数量属于 NP-hard 问题，我们针对工业4.0微服务应用开发了定制化解决方案。因此，在流程图 5.3 的第三个流程框中，使用公式 5.1 和 5.2 对最小割产生的分区在雾联邦中成功的可能性进行估算。由于我们利用概率来确定分区，因此将所提出的分区方法命名为概率分区 （ProPart）。如果新子图的完成成功率低于先前的成功率（流程图的第 2 个条件），我们将之前的分区视为最优分区（回滚至 w流程框）。相应地，将启动针对这些分区的资源分配方法，这也是流程图的结束流程框。

另一方面，如果最新分区按时完成的概率高于此前的成功率，我们就会评估每个分区的微服务架构，这是流程图中的第三个条件。如果不满足该条件（第三个条件的“否”分支），对于仅包含一个微服务的分区，分区过程将停止，并由资源分配服务接管。相反，对于包含多个微服务的分区（第三个

条件的“是”分支），将重复执行分区过程，直到每个分区都只包含一个微服务。因此，分区方法是一个重复过程，其输出结果为提交至资源分配模块的最优分区数量。

5.3 雾联邦中分区微服务应用程序的资源分配方法

雾联邦之间的.....ങ്ങളിൽ

分区完成并确定最优分区数量后，即进行资源分配。对于被视为单个微服务工作流的单体应用，分区方法会将整个应用程序作为一个部分返回给资源分配模块。资源分配方法在应对应用程序到达时间的不确定性（例如，灾害时期）并增强雾系统的韧性方面发挥着重要作用。资源分配模块以即时模式 [96]运行，并迅速将输入应用或微服务分区分配给相关雾系统。相关性由雾系统确定，从而提高微服务在截止期限前完成的可能性（即成功概率）。因此，可以利用历史端到端时延分布来估算微服务在特定雾系统上 m_i 按时完成的可能性。此外，为了尽量减少频繁的应用重新分配和复合延迟，我们决定，一旦选定分配方案，就不能迁移微服务。

每个雾系统的资源分配模块都会利用各类微服务类型在雾联邦中的计算延迟和通信延迟历史数据，

生成其分布的概率密度函数。为此，每个负载均衡器都会通过两个矩阵跟踪每种微服务类型在每个邻近雾系统上的计算延迟和通信延迟：估计任务完成时间（ETC） [97] 和估计任务传输时间（ETT）。微服务类型 i 在雾系统 j 上的计算延迟概率密度函数存储在条目$ETC(i, j)$中，该条目此前已用于分区方法。同样，条目$ETT(i, j)$保存了微服务类型 i到达雾系统 j的通信延迟概率密度函数。因此，资源分配模块也能感知通信延迟，而分区方法只能感知计算延迟。ETC和ETT矩阵的条目会定期离线更新，不会干扰负载均衡器的实时功能

.

资源分配模块可以在微服务分区到达时，通过对 $ETC(i, j)$ 和 $ETT(i, j)$ 进行卷积，计算雾联邦中的端到端时延分布。在任意雾系统 j 上，都可以利用端到端分布来计算每个微服务分区 mp_i 在截止期限前完成的概率，记为 $p_j(mp_i)$。因此，我们将该分区内每个微服务的截止期限相加，以估算给定分区 mp_i 的截止期限 δ_i 。然后，将每个微服务的计算延迟分布 d_{comp} 与通信分布 d_{comm} 进行卷积，以测量其在特定雾系统中的完成时间 e_i 。为估算分区 mp_i 的完成时间（记为 E_i），我们对给定分区内每个微服务的完成时间分布进行卷积。具体如下： $p_j(mp_i) = P(E_i \leq \delta_i)$。我们看到

在接收雾节点上， mp_i 的概率不涉及任何额外的通信延迟。因此，对于接收雾系统，在完成时间估算中，我们不对通信时延分布进行卷积。在后续阶段，选择完成可能性最高的雾系统作为可行目的地，以分配 mp_i。这意味着，只有在考虑通信延迟后，周边雾系统能提供更高的按时完成概率时，才会将微服务分区 mp_i 分配给相邻的雾系统。

需要注意的是，邻近雾节点上的成功率可能高于接收雾节点。这是因为，只有在将微服务分区分配给距离较近的雾系统能够显著提高该分区成功完成的可能性时，才应进行此类分配。因此，我们使用底层端到端分布的置信区间（CI），并据此推导接收雾节点和远端雾节点的成功可能性，以评估这种差异是否具有显著性。具体而言，只有当邻近雾节点对于给定的微服务分区能够提供显著更高的成功可能性时，该邻近雾节点端到端分布的置信区间才不会与接收雾节点的置信区间重叠。

算法 2 中提供的伪代码描述了负载均衡器利用联邦雾系统来提高系统鲁棒性的资源分配方法。该方法称为 最大鲁棒性（*MR*），当分区方法发送微服务分区时会调用该方法

算法 2： 资源分配算法

输入： 微服务分区集 M； ETC 和 ETT 矩阵； G（相邻雾系统集合）

输出： 选定雾节点 $f \in G$，用于分配微服务分区 $mp_n \in M$

1 **对于每个** 微服务分区 $mp_i \in M$ **执行**
2 $p_r(mp_i) \leftarrow$ 接收雾节点 r 上的成功概率
3 **遍历**雾系统 $f \in G$ **执行**
4 $p_f(mp_i) \leftarrow$ 邻近雾上的成功概率 f
5 **如果** $p_f(mp_i) > p_r(mp_i)$ **则**
6 将 $p_f(mp_i)$ 添加到 P，作为分配的候选雾
7 **end**
8 **end**
9 按降序排列集合 P 中的元素
10 将接收雾节点 r 视为 mp_i的默认分配
11 **对** $p_f \in P$ **中的每一项执行**
12 **如果** E_f 的置信区间与 N_r 的置信区间不重叠**则**
13 选择雾 f 作为目的地，并将 mp_i 分配给它
14 退出循环
15 **end**
16 **end**
17 **结束**

M 用于资源分配。首先，将微服务分区分离，以便进一步处理。然后，根据每个微服务分区的截止期限（δ_i）mp_i，算法计算接收雾节点及其相邻雾系统上的成功概率（算法 2 的步骤 2-8）。接下来，步骤 9 按降序排列计算出的概率。如果接收雾节点上的成功概率较高，则考虑将微服务分区 mp_i 分配给接收雾系统（步骤 10）。否则，将成功概率最高的邻近雾节点的端到端时延分布的置信区间与接收雾节点的置信区间进行比较。如果两个置信区间不重叠，则将分区 mp_i 分配给邻近雾节点（步骤 13）。否则，执行相同的过程

对接收雾系统的其余邻居执行上述操作。如果未找到不重叠的邻居，则将分区 mp_i 分配给接收雾系统（步骤 10 中的默认分配）。

5.4 软件架构感知型联邦雾系统的性能评估

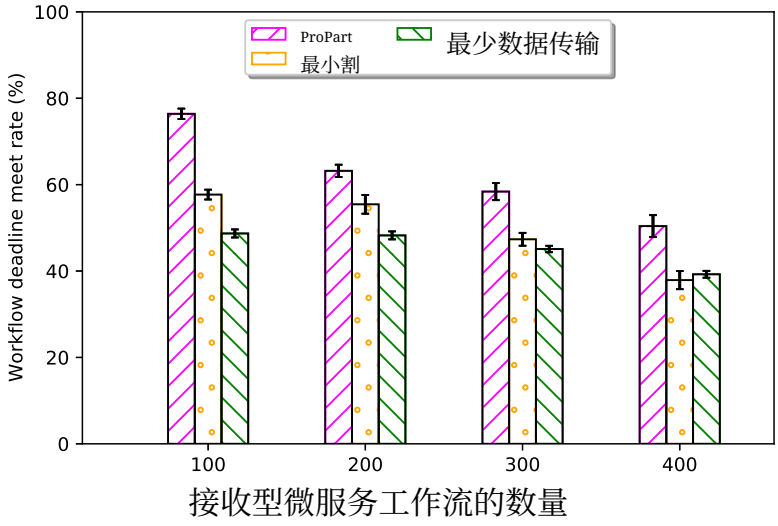
所提出技术的分区和资源分配组件在负载均衡器模块内依次执行。因此，我们在各种实验中分别评估每个组件。在第一个实验中，将推荐的分区方法与不同的基线方法进行比较，以考察基于微服务的工作流应用的截止期限约束得到了怎样的改善。在完成分区后，还必须评估向这些分区分配资源的效果。我们执行第二类试验来评估系统的有效性，将所提出的资源分配技术与其他基线方法进行比较。随后进行第三个实验，以确定扩展雾联邦如何影响所提出的解决方案。最后，对于微服务和单体应用，我们考察由分区和资源分配方法产生的计算延迟。后续小节将对这些实验进行详细介绍。

5.4.1微服务工作流分区方法的比较 *s*

在本实验中，我们采用所提出的分区技术（概率分区，定义为 ProPart）来接纳基于微服务的工作流应用，并将其与另外两种基线方法（最小割、最少

数据传输等）进行比较。在本实验中，我们增加提交到系统的微服务应用数量，以产生超额订阅的情况，并记录每轮请求提交中应用程序的截止期限达成率，如图 5.4 的柱状图所示。图中的横轴表示系统接收的基于微服务的应用程序数量，纵轴表示应用程序截止期限的达成率。

图 5.4. 采用所提出的概率分区技术时，各分区技术在工作流截止期限达成率方面的比较。横轴表示不断增加的到达工作流执行请求数量，纵轴表示工作流截止期限达成率。



本实验的结果如图 5.4 所示，表明对于所有分区技术，随着系统收到的工作流请求数量增加，截止期限达成率会降低。然而，在每一轮提交中，ProPart 都优于其他基线方法。在负载较低的场景（例如，100 和 200 个请求）中，效率最低的策略（最少数据传输）与所提出的技术 ProPart 之间的性能差距，大于完全超额订阅场景中的差距

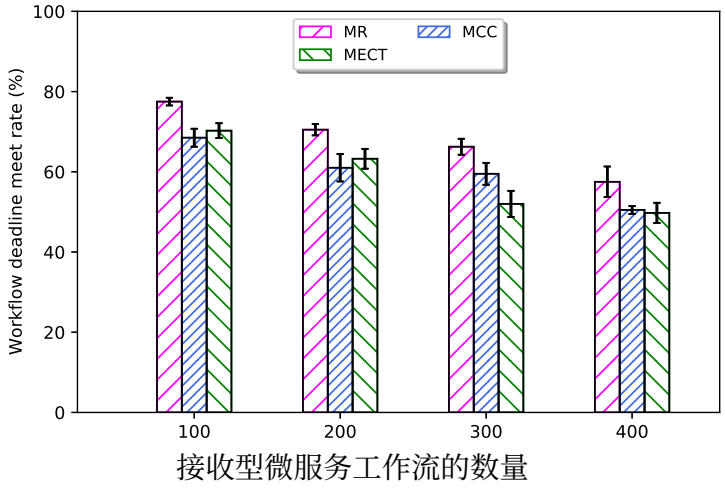
（例如，300 和 400 个请求）。ProPart 性能更优的主要原因，是它会对每个分区的成功可能性进行统计评估。在应用请求数量不超过 200 时，最小割策略的表现优于最少数据传输。相反，在完全过载的场景中，最少数据传输的表现略优于最小割，因为它会考虑拆分时产生输出数据最少的连接。相比之下，最小割在分区时会寻找通信量最小的通道。最后，由于 ProPart 会反复对所有微服务能否满足截止期限进行概率计算，因此它在完全超额订阅的情况下表现更好。

5.4.2资源分配方法比较

每个雾系统中的负载均衡器在基于微服务的工作流应用完成分区步骤后，都会采用一种资源分配技术。相比之下，对于单体应用，负载均衡器一旦收到请求，就会通过在雾联邦中进行概率估计来执行资源分配。因此，为了比较所提出的资源分配技术，我们分别针对微服务应用和单体应用，采用三种不同的资源分配方法进行了以下实验。

基于微服务的工作流应用：与前一个实验类似，逐步增加接收型微服务应用的数量，以产生更多超额订阅情形（即图的横轴）。为直观展示资源分配技术的性能，我们记录了接收应用的截止期限满足率，并将其绘制在图5.5中。

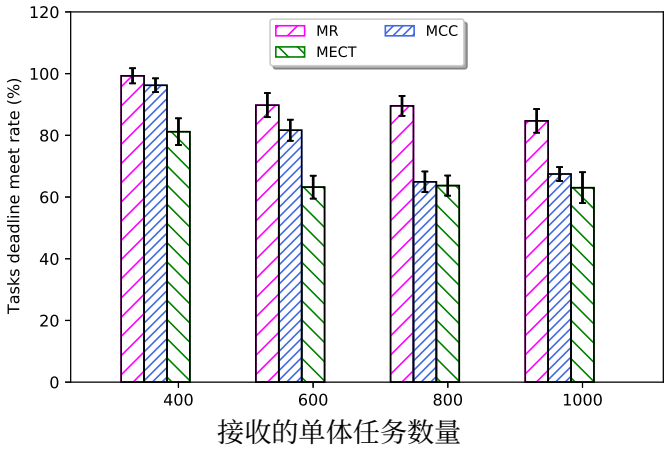
图5.5. 在微服务工作流应用中采用所提出的工作流分区技术时，对资源分配技术的比较。



结果显示，随着超额订阅情形增多，所有资源分配技术的表现均呈下降趋势。因此可以看出，在每种超额订阅情形下，所提出的资源分配技术 MR 均优于其他基线方法。这是因为 MR 能够感知接收型微服务在计算和通信方面的不确定性。相比之下，MECT仅能感知计算情况，而Certainty在其资源分配技术中利用截止期限，却缺少通信信息。

单体独立应用：在本实验中，我们通过增加传感器生成的单体应用程序数量（即超额订阅级别），研究系统的性能。图5.6展示了采用各种资源分配启发式方法时，输入应用程序数量（横轴从400到1000）变化对期限满足率（纵轴）的影响。

图5.6. 单体式应用程序资源分配技术的比较。所提出的资源分配技术 MR 在每次应用到达试验中均优于其他基线方法。



从图5.6可以看出，随着应用程序数量增加，所有启发式算法的截止期限达成率都会下降。在超额订阅水平较低时（400个任务），MR、MECT和MCC的表现相似。然而，随着系统超额订阅程度加剧（800个应用程序），差异变得显著。当应用程序数量达到1000时，MR的期限满足率比MECT和MCC高出约18%至20%。这是因为MR能够捕获端到端时延，并且只有在联邦能显著影响成功概率时才会主动利用联邦。相比之下，其他基准启发式算法仅考虑计算延迟。因此，我们可以得出结论：对于单体应用，考虑端到端时延并捕捉其内在不确定性，可以显著提升鲁棒性，尤其是在系统超额订阅时（例如，灾害时期）。

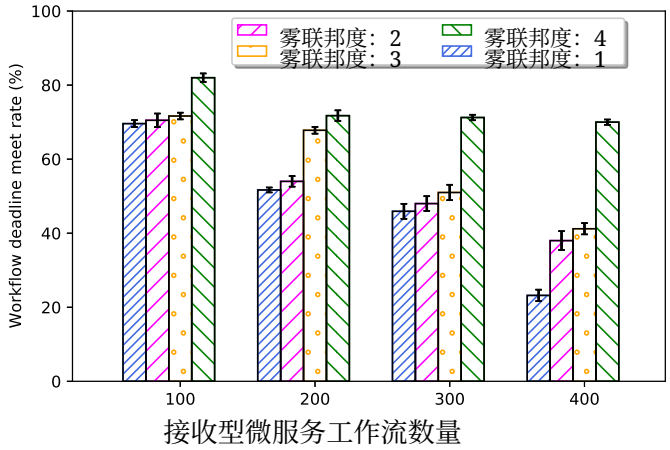
5.4.3雾联邦扩展影响

偏远海上地区的雾联邦可以在

紧急情况下，通过利用安装在船只或其他交通工具上的移动雾系统进行扩展。相反，缩小雾联邦规模可能会降低系统鲁棒性。因此，为了了解联邦规模变化对所提出方案的影响，我们提高了雾联邦度（表示邻居数量），并在超额订阅情形不断加剧时，记录所接收应用程序的截止期限达成率。该实验的结果见图5.7。此外，我们还针对单体应用进行了类似实验：固定接收任务的数量，并逐步提高雾联邦度。单体应用的结果

pp应用是　　p如图　　g5.8所示。

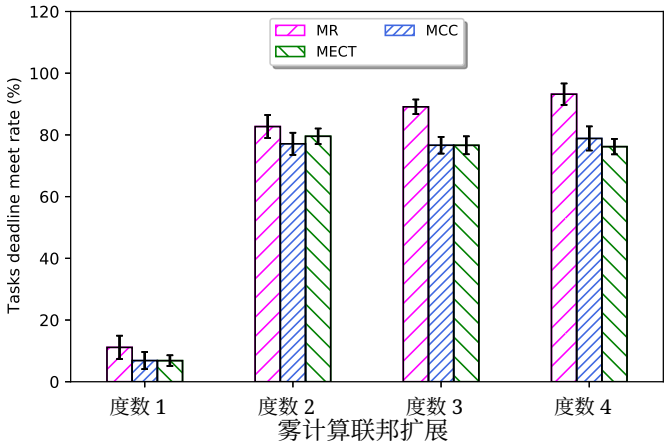
图5.7。 在微服务应用程序的超额订阅情形逐渐加剧时，针对所提出的分区和资源分配技术，扩展雾联邦的影响。度数表示每个雾系统拥有的邻居数量，用于执行工业4.0应用程序。



基于微服务的应用： 5.7 中所示的结果展示了扩大雾联邦的优势。因此，在任何过载情况下，邻居数量最多的联邦（即 雾联邦度数 4）表现最佳。尽管如此，在

大多数超额订阅情况下（即 处理 400 个基于微服务的工作流的系统），性能提升仍然显著。对于负载较低的情形（例如，接收 100 至 200 个基于微服务的工作流的系统），小幅扩大雾联邦带来的性能差异可以忽略不计。这是因为在超额订阅程度较低的情况下，所提出的方法，尤其是分区技术，试图将整个应用程序放入一个雾系统，而不是对其进行分区并将其分散到整个联邦中。因此，在邻居数量最多且完全超额订阅的情形下，性能提升十分显著。

图 5.8。 在单体应用上，扩展雾联邦对所提出的资源分配技术的影响。度数表示每个雾系统为执行工业4.0应用程序而拥有的邻居数量。



单体应用：在本次实验中，我们比较了雾联邦规模扩大时针对单体应用的资源分配技术。与基于微服务的工作流类似，单体应用也会对联邦扩展产生积极影响，这一点在图 5.8 中清晰可见。结果显示，当联邦规模从度数 1 扩大到

度数 2 时，所有启发式算法的性能均显著提升。因此，度数 1 表示只有一个邻居，联邦由两个雾系统组成。在这种情况下，所有启发式算法的表现都不佳，尽管所提出的 MR 方法优于基线方法。相比之下，在联邦度数最高时，所提出的 MR 启发式算法比 MECT 和 MCC 的表现约高出 18%–20%。然而，在联邦规模的所有扩展情况下，所提出的 MR 启发式算法都优于其他方法。主要原因在于，MR 能够高效利用雾联盟资源，同时考虑通信和计算延迟，从而确保每个单体应用都能按时完成。

5.5 总结

软硬件栈的发展推动了工业革命——工业4.0，它改变了许多遗留系统架构，并带来了延迟约束。因此，复杂的工业流程每天都在采用智能解决方案。由此，数据源附近计算为智能微服务解决方案提供支持，而这类方案面临着严峻的资源匮乏和延迟约束挑战。尤其是在偏远海上产业（例如，石油和天然气、挖矿）中，复杂且容错性低的工业流程（例如，油气勘探、钻井）的时延问题可能至关重要。此外，在紧急情况下，计算执行平台会因承载各种类型的微服务而出现超额订阅。为应对智能微服务解决方案带来的挑战，本研究提出了一种鲁棒的任务分配方案，该方案能够感知解决方案的软件架构以及

雾联邦带来的不确定性。因此，所提出的解决方案在联邦中每个雾系统的负载均衡器模块内分两个层级运行。第一级考虑接收应用程序的软件架构，并在必要时进行分区，同时利用概率成功率来完成应用程序。第二级则将接收到的应用程序（例如，单体应用或分区微服务）映射到雾联邦中的各个节点，同时考虑计算和通信约束。评估结果表明，在超额订阅情形下，所提出的解决方案优于基线分区和资源分配技术，效果约提升 15∼20%。未来，我们计划引入基于机器学习的资源配置方法，以提高联邦雾系统的鲁棒性。

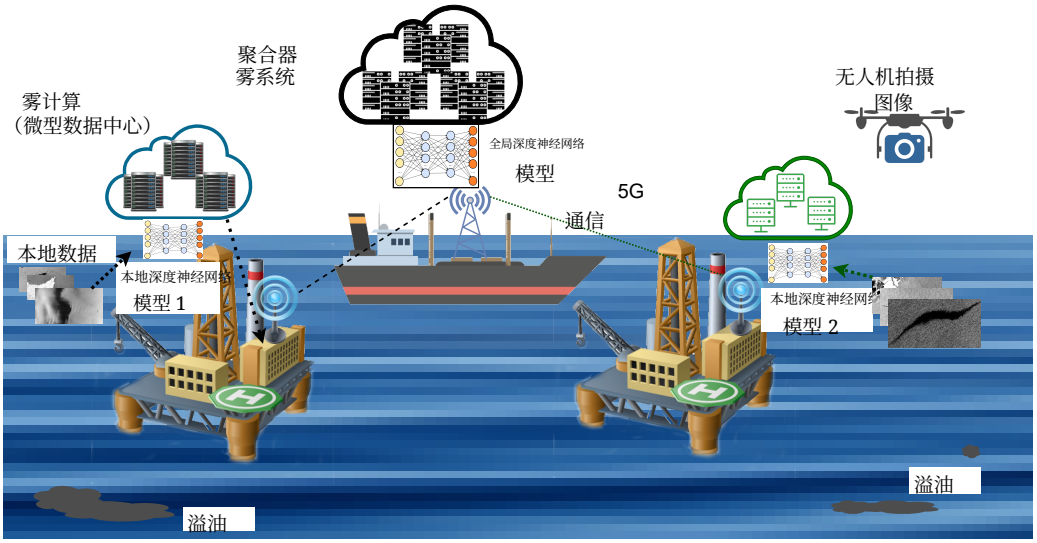
第6章：联邦雾计算系统中的数据安全与隐私方面

6.1 概述

工业4.0 [113] 的兴起推动了物联网设备（例如，传感器、执行器）和雾计算的应用，以便在各个工业领域（例如，智能油田、智能农场、智能工厂）开发深度神经网络（DNN）应用程序。这些基于深度神经网络的应用程序主要依托机器学习网络模型，而这类模型需要使用大量数据进行训练，才能达到较高的准确性。尽管训练数据可能需要保护隐私（也就是说，对任何公司而言都很敏感），而且数据采集（例如，卫星图像数据、高分辨率相机数据）有时既昂贵又耗时，但通过互联网将数据传输至云数据中心以进行训练操作，也会增加开发这些基于深度神经网络的应用程序的成本。因此，由多家私营公司雾系统组成的雾联邦，可以作为满足机器学习模型训练计算需求的潜在方案；为实现高效的机器学习训练，还需要解决联邦中参与方私有雾系统的数据安全与隐私问题。在这种情况下，联邦学习（FL）技术 [66] 能够在不让数据离开其源设备的前提下，将机器学习模型带到参与方的终端用户处，因此可用于克服私营公司所拥有雾系统的隐私约束。尽管如图6.1所示，联邦学习缓解了隐私保护约束，但它也可能给机器学习模型训练操作带来一些新的挑战。这里的问题在于，数据来自各种

来源，而且可以合理地假设数据分布趋于非独立同分布。因此，在图6.1所示的联邦学习环境中，缺少优先类别（即在溢油检测问题中考虑溢油类别），也就是所谓的类别不平衡 [114]，可能会降低全局深度神经网络模型的性能。因此，忽视类别不平衡问题的现有联邦学习方法 [66]，在溢油检测中提供的深度神经网络模型鲁棒性较差。在这种情况下，目标检测模型可能会对所有其他类别给出具有误导性的高准确性，却在目标类别（溢油）上表现不佳。

图6.1. 雾联邦中的联邦学习设置。多家公司共享其雾系统，以训练用于溢油检测的深度神经网络模型，同时通过联邦学习保障数据安全。



联邦学习是分布式机器学习的一个特殊分支，其中全局模型需要以受限的速率收敛。因此，联邦学习的收敛主要取决于本地工作节点的聚合，而聚合会影响全局模型的性能。在两种联邦学习类型（同步和

异步）中，我们提议采用同步联邦学习方法，因为它已被证明有效，尤其适用于类别不平衡问题 [115, 116]。因此，为了应对联邦学习在溢油检测方面面临的挑战，我们采用目标函数（损失函数）训练本地模型，并考虑类别不平衡问题。考虑到优先类别（即 溢油），我们为每个参与工作节点引入一个权重，以增强或减弱其对全局模型的影响。第 6.5 节通过实证评估验证了基于工作节点权重进行相关工作节点选择的有效性。最后，我们提出了一种动态阈值机制，该机制综合考虑全局模型的性能和快速收敛，以高效选择相关工作节点。

6.2 联邦学习问题表述

溢油检测问题可以在深度学习的语义分割领域中得到良好定义：该领域从原始源图像中以像素级识别各种类别。在溢油检测训练中，真实世界的卫星图像数据集[77]中包含各种类别（例如，溢油、相似目标、陆地、船舶、海面）。这里，真实标注图像中的每个类别都以一种独立颜色标记。在联邦学习设置下训练深度神经网络（DNN）模型（例如，U-Net）时，考虑一组工作节点（即，雾联邦中的雾系统） $S = 1$，2，3，...，S，每个工作节点都有自己的本地数据集 D^L，其中 $L \in S$ 包含 n_L 个样本。这里，$D = \bigcup D^L$ 是完整的训练数据集。对于$\mathcal{L} \in S'$个工作节点的随机集合，其数据集的总大小为 $N(S') = \sum n_L$。模型m 和样本z 上的目标损失函数可表示为 $\mathcal{L}(m, z)$。

因此，在大多数先前的联邦学习工作中，目标是求解以下问题

$$\min_{w} \quad f(w) = \sum_{m=1}^{M} p_m F_m(w)$$

其中， $p_m = \frac{n_m}{D}$ 是工作节点总数据量的占比，因此 $\sum_{n_m} p_m = 1$。m本地目标函数 F_m 通常定义为本地数据上的经验损失，$F_m(w) = \frac{1}{n_m} \sum_{j=1} \mathcal{L}_j(m, z)$。此处， w 是用于预测样本数据损失的模型参数，目标是找到使损失最小化的最优 w 。因此，我们着重使用能够考虑本地数据样本中类别不平衡问题的损失函数，并选择一组具有一定准确性水平的客户端工作节点（雾系统）模型进行聚合（即 语义分割的平均交并比（mIoU）），以确保全局模型的鲁棒性。因此，本研究提出的新目标如下:

$$\begin{aligned} \min_{w} \quad & f(w) = \sum_{m=1}^{M} p_m F_m(w) \\ \text{s.t.} \quad & mIoU(m) >= \gamma, \\ & \theta > 1 \end{aligned}$$

其中， γ 是动态阈值（初始值设为 50% 或 0.5），用于利用辅助测试数据检查本地训练模型的平均交并比； θ 是用户定义的工作节点在油污类别上的权重。这两个参数用于选择相关工作节点的模型并将其聚合到全局模型中，从而确保聚合模型的鲁棒性和收敛一致性。

表 6.1. 溢油检测数据集中各类别的像素分布

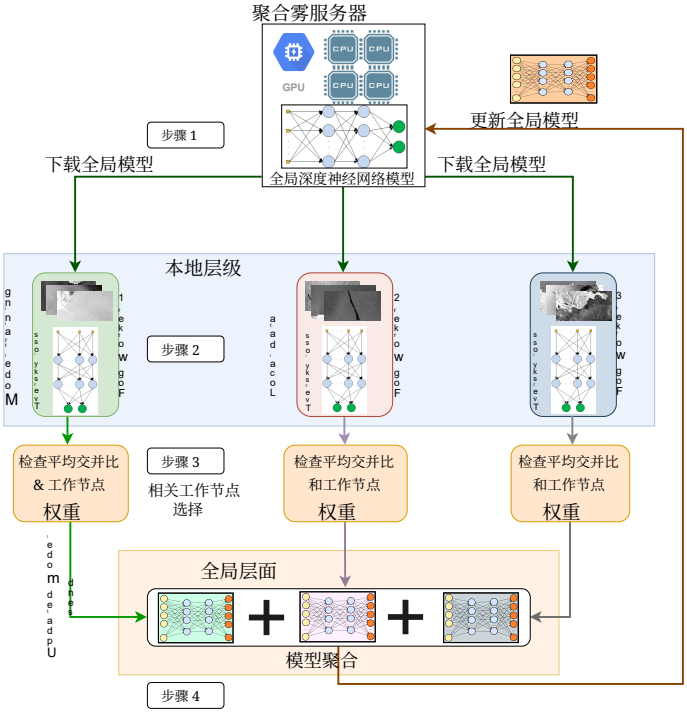
类别	像素
海面	797.7 M
溢油	9.1 百万
相似目标	50.4 百万
Ship	0.3 百万
Land	45.7 百万

6.3 利用联邦学习缓解类别不平衡

在典型的联邦学习设置中，服务器（例如，雾设备、云）存储一个全局机器学习模型，以便使用参与工作节点（即，雾系统）的本地数据进行训练。在联邦学习设置中，我们使用一种名为 U-Net [117] 模型的常见语义分割深度神经网络模型来进行溢油检测。图 6.2以示意图的形式展示了我们的解决方案。首先，一些雾节点同意参与联邦学习训练，并在图 6.2的步骤 1 中从雾服务器下载全局模型（即，U-Net）。然后，在步骤 2 中，使用各自的本地数据训练下载的机器学习模型。因此，我们采用 *Tversky* 损失 函数进行本地训练；研究界已证明，该函数能有效处理类别不平衡问题 [118, 119, 120, 121, 122]。在步骤 3 中，对机器学习模型进行检查，以选择相关工作节点的模型。最后，在步骤 4 中，所选工作节点的更新模型被聚合（形成新模型），并相应更新先前的全局模型。整个过程称为一个联邦轮次。参与工作节点会再次下载更新后的模型，以进入下一联邦轮次，训练也随之继续。所提出的解决方案见算法 3。

U通常，聚合雾服务器会提供全局模型，并聚合

图 6.2. 考虑类别不平衡和全局收敛的联邦学习训练。在考虑类别不平衡的训练中使用 Tversky 损失。每轮训练结束后，使用动态阈值检查平均交并比（平均交并比）。



工作节点雾发送的更新。*FedBal* 算法首先初始化全局模型 m_g、相关工作节点列表 r_f，并将阈值 t_h 设为 0.50。随后，联邦轮次以 for 循环的形式继续进行，并通过变量 f表示。接着，从 K 个参与工作节点中选出 m 个工作节点，并将其分配到已选工作节点列表 S_t，以便在算法 3 的第二个 for 循环中使用其本地数据进行训练（“ClientUpdate”函数）。最后，使用“selectionCriteria”函数选出相关工作节点，并通过“averageModel”函数将其聚合到新的全局模型 m_g 中。“ClientUpdate”函数使用*Tversky* 损失 函数进行训练，并采用设定的训练轮次，以缓解本地层级的类别不平衡。所提出方案的全局层面是

算法 3： K 工作节点的索引为 k， C 是初始工作节点
选择百分比

初始化全局模型、 m_g、测试数据、 D_{test}、相关工作者列表、 r_f;
设置阈值， $t_h = 0.50$;
对每个联邦轮次 $f = 1,2,...$ **执行**
    $m \leftarrow$ 最大值(C.K,1);
    $S_t \leftarrow (random\ set\ of\ m\ workers)$;
    对于每个工作节点 $k \in S_t$ **并行地执行**
        $ClientUpdate(k, m_g)$;
    $r_f = selectionCriteria(S_t, D_{test}, t_h)$;
    $m_g = averageModels(m_g, r_f)$;

在“selectionCriteria”函数中触发，该函数根据训练后的工作者模型的权重、 θ 和 $mIoU$ 值对其进行评估。动态阈值机制也在“selectionCriteria”函数中发挥作用，以确保全局模型的鲁棒性。这样，在每个联邦轮次中， f，全局模型都会更新并收敛到一个能够抵御类别不平衡、同时保证优先类别——溢油——性能的模型。

6.4 实验设置

联邦学习设置可以通过 PyTorch 的一款热门库 pysyft [123], 以及 TensorFlow 的联邦学习库 tff [124]来实现。考虑到 pysyft 的定制能力，我们选择 pysyft 作为开发库。溢油检测被视为语义分割问题，通常使用真实世界的 SAR 图像数据集来训练深度神经网络模型（在本研究中，数据集由希腊的一家研究机构 MKLab [77],收集）。为执行深度神经网络训练操作，我们使用了 Google Colab [125]运行时环境，该环境提供图形处理器平台，并配备高速随机存取存储器，容量为

24 GB，存储容量为 128 GB。

Colab 为付费“专业版”提供 Tesla P100、T4 或类似的图形处理器。它还提供高内存选项，以便在使用图形处理器时更快地执行。我们利用 pysyft 的虚拟工作节点概念来模拟雾设备。本研究的主要目标是减少类别不平衡问题，并确保全局模型稳健。因此，我们专注于联邦学习的计算部分，而忽略传统联邦学习设置中的通信（即 网络）部分。我们的联邦学习设置可用于任何聚合算法（例如 FedAvg、FedSGD、FedProx），因此我们基于这些基线算法开发了代码库。由于我们的联邦学习设置致力于减少类别不平衡，我们将其命名为“FedBal”。在大多数实验中，我们使用 20 个联邦轮次，每轮包含 50 个训练轮次。选择这些训练参数（例如，训练轮次数、联邦轮次数）的原因，是为了观察不同聚合算法之间的显著差异。最后，由于时间约束，我们将实验限制在 20 个联邦轮次的聚合之内。

6.5 性能评估

对于数据往往频繁生成的雾设备而言，联邦学习设置总是有益的。因此，为了了解采用联邦学习的优势，我们进行了实验，记录使用联邦学习和单机训练时每个训练轮次的损失。联邦学习设置 (a) 可以使用更多数据，因为有四个工作节点参与深度神经网络模型（U-Net）的训练。另一方面，非联邦学习设置

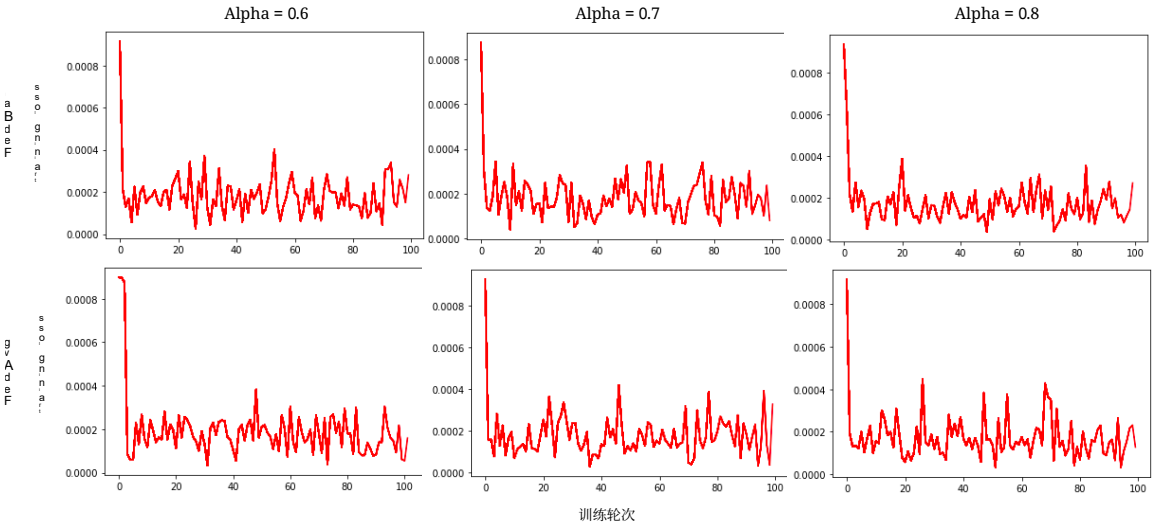
使用单个雾设备训练模型，所用数据较少。此外，我们在初步实验 (b) 中发现，联邦学习设置的不确定性比非联邦学习更小。就训练模型的收敛速度而言，联邦学习也比非联邦学习更快。尽管联邦学习的性能优于典型的机器学习，但本地数据中的类别不平衡问题可能导致全局模型的性能退化。因此，我们的本地工作者级解决方案采用 Tversky损失函数，其中 α 用于惩罚假阴性、 β 用于惩罚假阳性的参数需要进行调优，以获得更好的性能。有关这些参数的实验将在下一节中介绍。

6.5.1损失函数调优

为了找到最佳的 Tversky 损失函数，我们将 alpha参数的值从 0.6 调整到 0.8，并记录每个训练轮次的损失。主要目标是找到使损失最小的最佳 alpha 值。这些实验的结果如图 6.3 所示。

图 6.3 展示了在 FedAvg 和 FedBal 联邦学习设置中分别使用 0.6、0.7 和 0.8 的 alpha 值时每个轮次的训练损失（即，图的纵轴）。从图 6.3 可以看出，与 FedAvg 相比，FedBal 的表现相近。还可以看出，当 alpha 值为 0f 0.7 时，FedBal 的训练损失最低。当我们将 alpha 值增加到 0.8 时，训练损失并未降低，这意味着我们只能在一定程度上惩罚假阴性（即， $\alpha = 0.7$）。原因在于，当我们惩罚假阴性时，假阳性预测会被忽略（即， $\alpha + \beta = 1$）

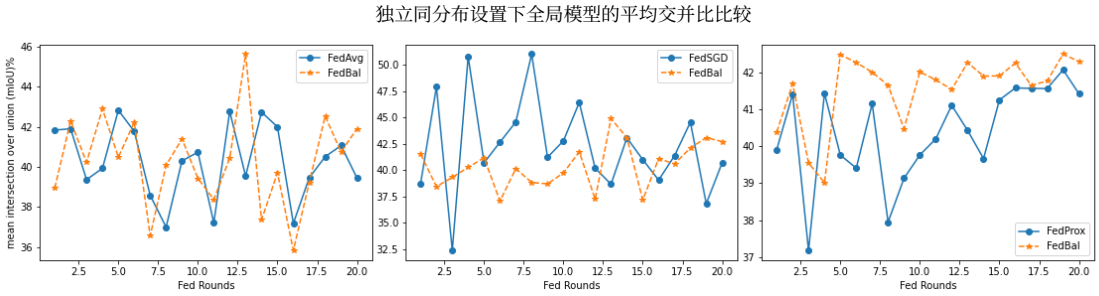
图 6.3. 使用 Tversky 损失函数比较 FedAvg 和 FedBal 的训练损失。将 Tversky 指数的 alpha参数从 0.6 调整到 0.8（从左到右），并记录 FedAvg 和 FedBal 算法的每轮损失。



as 同样如此。因此，alpha 值越高，通过惩罚所获得的益处就越少

假阴性预测。因此，在本文接下来的所有实验中，我们使用 $\alpha = 0.7$ 。

图 6.4. 在独立同分布设置下比较 FedBal 与 FedAvg、FedSGD 和 FedProx 方法的全局模型性能。



(a) 与 FedAvg 的比较　(b) 与 FedSGD 的比较　(c) 与 FedProx 的比较

6.5.2使用独立同分布数据分布的影响

联邦学习设置的优势体现在联邦学习完成聚合步骤后全局模型的性能（即准确性（平均交并比））上。

因此，我们分别在 FedBal 与 FedAvg、FedSGD 和 FedProx 的每次通信或联邦轮次结束后，测量全局模型的平均交并比。随后，我们将结果绘制成图 6.4 中的折线图。在此实验中，我们假设联邦学习工作节点之间的数据分布为独立同分布（即 IID），这意味着每个工作节点在本地数据中都包含所有图像类别，以用于训练。

图 6.4 的横轴表示联邦轮次，纵轴表示全局模型的平均交并比。从图 6.4 (a) 可以看出，在大多数联邦轮次中，FedBal 的表现优于 FedAvg。就 FedSGD 而言，从图 6.4 (b) 可以看出，FedBal 在最后几个轮次中表现显著更好。不过，在最初几个轮次中，FedSGD 的表现优于 FedBal。最后，从图 6.4 (c) 可以看出，与 FedProx 相比，我们的 FedBal 方法表现显著更好。这一改进主要得益于将未参与工作节点纳入全局模型。此外，FedBal 的工作节点选择会考虑类别不平衡问题和优先类别（即，油污类别），以便将其聚合到全局模型中。相比之下，其他方法会随机选择活跃工作节点进行聚合，因此其全局模型的鲁棒性不如 FedBal。

6.5.3使用非独立同分布数据分布的影响

在现实场景中，联邦学习工作节点之间的数据分布通常是非独立同分布的。这意味着每个工作节点在本地训练时都会获得固定数量的类别（并非所有类别）。因此，我们考虑为每个工作节点提供两个类别，并且每个工作节点获得的类别都不相同。与之前的实验类似，

我们在每个联邦轮次中测量 FedAvg 和 FedBal 算法的全局模型的平均交并比。结果如图6.5所示，包含20个联邦轮次和六个联邦雾计算工作节点。

图6.5。使用 FedAvg 和 FedBal 方法时，全局模型在平均交并比方面的性能比较。数据分布为非独立同分布，工作节点数量为 6，每个联邦轮次都进行了 50 轮训练。

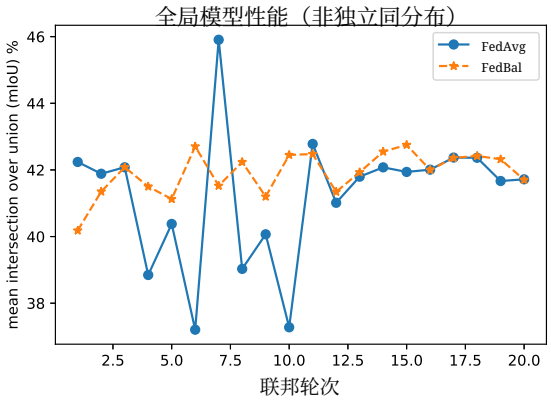


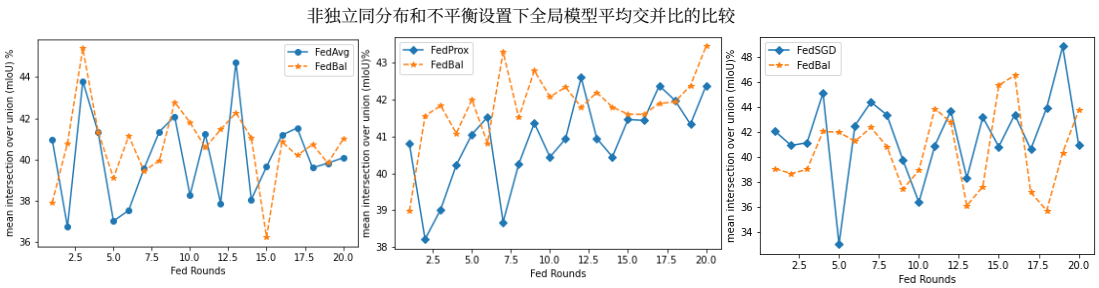
图6.5显示，经过20个联邦轮次后，FedBal的性能（平均交并比）比FedAvg更加稳定。在每个联邦轮次中选择相关工作节点时，FedBal方法在各联邦轮次中的不确定性更低（图6.5中橙色曲线的尖峰更少）。尽管FedBal的性能提升不如FedAvg显著，但FedBal全局模型的平均交并比高于FedAvg。FedBal的这种稳定表现体现了我们的方法在各联邦轮次中的鲁棒性。

6.5.4使用非独立同分布和不平衡数据分布的影响

非独立同分布且不平衡的数据分布意味着每个联邦学习工作节点拥有不同数量的类别。例如，工作节点一的训练数据中可能包含两个类别，而工作节点二的训练数据中可能只有一个类别。因此，我们测量

全局模型的平均交并比，并将我们的方法（FedBal）分别与另外三种基线方法 FedAvg、FedProx 和 FedSGD 进行比较。由于我们的方法被视为对任何联邦学习聚合方法（例如， FedAvg、FedProx、FedSGD）的改进，因此我们在三个不同的子图中分别比较这些基线方法。结果以折线图的形式展示在图 6.6 中，其中横轴表示联邦轮次，纵轴表示全局模型的平均交并比（mIoU）。

图 6.6。FedBal 与 FedAvg、FedProx 和 FedSGD 方法在非独立同分布和不平衡数据分布下的全局模型性能比较。



（a）与 FedAvg 的比较　（b）与 FedProx 的比较　（c）与 FedSGD 的比较

图 6.6 显示，在最终轮次中，FedBal 的表现分别优于 FedAvg、FedProx 和 FedSGD。不过，在第19个联邦轮次中，FedSGD 和 FedBal 的表现相近。与 FedProx 相比，FedBal 的性能提升十分显著，这得益于将被排除的工作节点纳入全局模型。还可以看出，基线方法的表现存在严重的不确定性（波动次数多于 FedBal），而 FedBal 在整个联邦轮次中表现相对稳定。这种稳定性的主要原因是 FedBal 采用了结合动态阈值机制的相关工作者选择，从而维持一定的性能，并提供一个稳健的全局

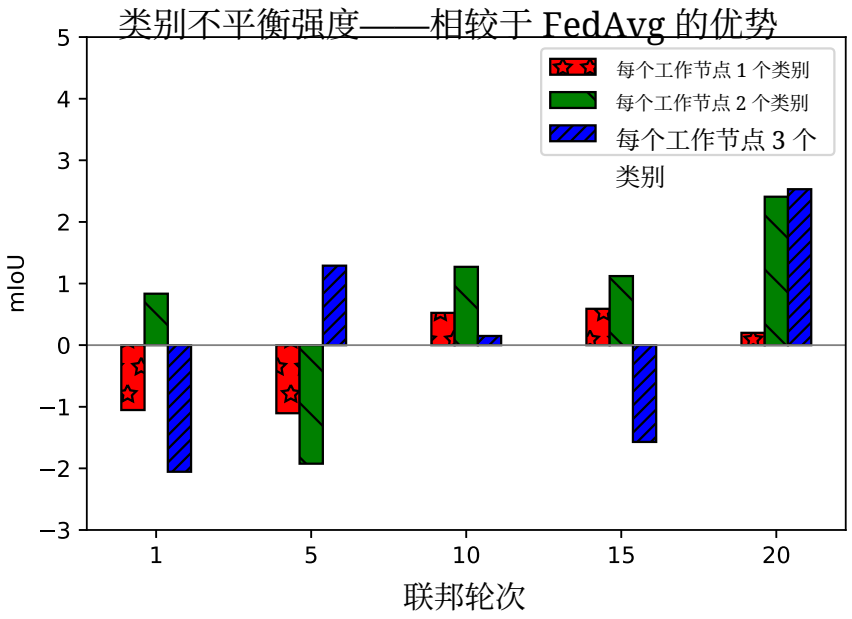
模型，以应对本地数据集中的类别不平衡问题。

6.5.5类别不平衡强度的影响

在石油泄漏数据集中，类别不平衡是一种常见现象，而在联邦学习设置中，不平衡程度可能十分严重。因此，我们认为，与基线算法 FedAvg 相比，我们的方案 FedBal 在各种类别不平衡强度下都能保持稳定的良好表现。为探究类别不平衡强度的影响，我们在非独立同分布设置下，按照类别不平衡程度从高到低分布类别，并测量 FedBal 和 FedAvg 的全局模型在联邦学习训练各联邦轮次中的平均交并比。我们估算了三种不平衡数据分布情形（一个类别、两个类别和三个类别）的每个联邦轮次平均交并比值之差。此外，我们绘制了柱状图，如图 6.7 所示，其中正值表示 FedBal 优于 FedAvg，负值则表示相反。

图 6.7 展示了在 20 个联邦训练轮次中，FedBal 算法相较于 FedAvg 算法的优势或劣势。在前九个联邦轮次中，FedBal 相较于 FedAvg 的提升并不显著。在第 10 个联邦轮次中，差值均为正；而在最终联邦轮次（第 20 轮）中，我们发现 FedBal 的表现优于 FedAvg，且优势最大。我们还注意到，对于每个工作节点 2 个类别的分布，FedBal 始终优于 FedAvg。对于类别分布高度密集的情况（每个工作节点仅 1 个类别），FedBal 在第 10 轮之后才开始表现良好。在最终轮次中，我们发现，对于 3 个类别的分布，FedBal 的提升最为显著。其背后的主要原因

图 6.7。 比较 FedAvg 与 FedBal 方法在非独立同分布数据分布下的全局模型性能，类别不平衡强度从高（每个工作节点仅 1 个类别）到低（每个工作节点 3 个类别）。图中以条形图形式展示了 3 种情形（1 个类别、2 个类别和 3 个类别）下 FedBal 与 FedAvg 的平均交并比差值。

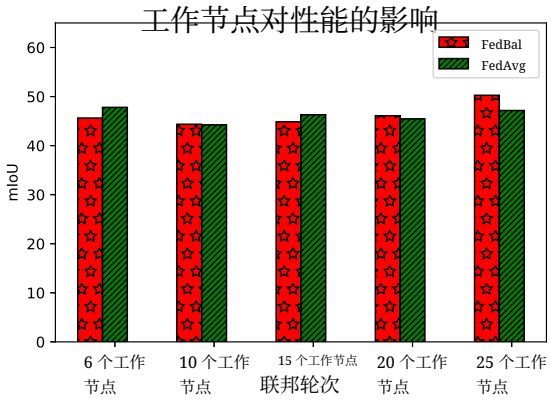


表现提升不显著的原因可能是动态阈值机制：该机制以较高的平均交并比值（50%）起步，并在联邦轮次中动态变化，从而提升全局模型的表现。第 10 轮之后，在相关工作节点数量充足的情况下，阈值趋于稳定；在联邦训练的最后十轮中，我们观察到了性能提升。

6.5.6工作节点数量对全局模型的影响

为了了解联邦学习工作节点对我们的联邦学习方法的影响，我们逐步将工作节点数量从 6 增加到 25，并测量 FedBal 和 FedAvg 方法的全局模型的最大平均交并比。本实验主要侧重于比较逐步增加工作节点数量时，我们的方法 FedBal 相较于 FedAvg 的性能，

图 6.8. 通过将联邦工作节点的数量从 6 个增加到 25 个，测量 FedBal 和 FedAvg 中联邦工作节点对全局模型性能（平均交并比）的影响。对于每种工作节点池情况，FedBal 和 FedAvg 方法均进行 20 轮训练，并选取两种方法各自的最高平均交并比绘制条形图。



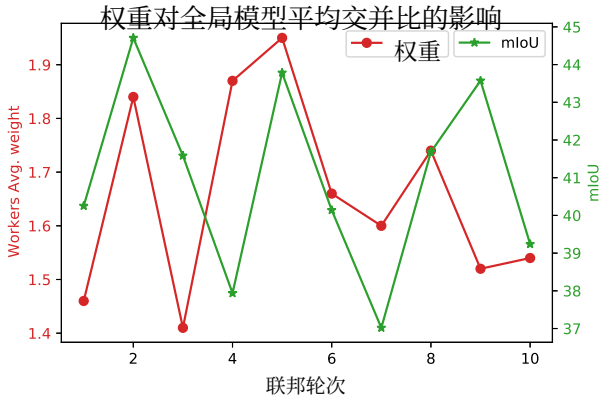
同时逐步增加联邦学习工作节点的数量。本实验的结果见图 6.8。

图 6.8 显示，随着联邦学习工作节点数量增加，FedBal 的性能显著提升。可以看出，当工作节点数量不超过 15 个时，FedBal 的性能提升不及 FedAvg。这是因为 FedBal 会从活跃工作节点池中选择相关工作节点，有时相关工作节点数量很少，导致全局模型的改进幅度较小。相比之下，FedAvg 在整个联邦轮次中始终选择相同数量的联邦学习工作节点，因此在工作节点池数量较少时表现更好。因此，使用 FedBal 方法增加工作节点数量，可以显著提升全局模型的性能。

6.5.7验证聚合方案对全局模型的影响

FedBal 算法的选择技术会考量相关

图 6.9. 工作者权重（在每个联邦轮次上取平均）对全局模型平均交并比的影响。



工作节点，并检查其由优先类别（油污类别）定义的权重。因此，表现出色的工作节点会对全局模型的平均交并比产生积极影响。为了探究工作节点权重对全局模型的影响，我们测量了每个联邦轮次中工作节点的平均权重，并估算聚合后全局模型的平均交并比。该实验结果如图 6.9 所示，其中横轴表示联邦轮次，左侧纵轴表示工作节点的平均权重，右侧纵轴表示全局模型的平均交并比。

图 6.9 显示，所选工作节点的权重对全局模型有正向影响。可以看出，随着平均权重增加，全局模型的平均交并比会上升；随着平均权重降低，平均交并比也会下降。因此，FedBal 方法根据优先类别“溢油”设置的用户定义权重，对其相关工作节点起到了支持作用。

6.6 总结

联邦学习技术彻底改变了分布式机器学习，尤其是在数据隐私和计算灵活性方面。随着物联网、边缘计算和雾计算的兴起，数据生成速度越来越快，而通过网络带宽传输数据的成本也越来越高。因此，与云数据中心相比，联邦学习可以将机器学习模型部署到数据生成源，成本更低且更加安全。在考虑为偏远地区提供机器学习支持时，这种方式比传统的集中式深度神经网络模型训练更适用。此外，偏远油田采集或收集的数据具有敏感性，因此隐私保护对油气公司至关重要。尽管联邦学习能够克服这些挑战，但类别不平衡问题可能会降低深度神经网络模型的性能。因此，我们着重在本地层级训练模型时减少类别不平衡的影响，并在将联邦工作节点聚合到全局模型时减少其在全局层面的影响。在本地层级，我们使用带有适当参数的 Tversky 损失函数（例如， α，β）进行调优，以考虑类别不平衡问题并训练每个联邦工作节点的模型。随后，我们根据优先类别“溢油”为每个工作节点分配权重。最后，我们将每个联邦工作节点模型的平均交并比与预先设定的、广泛认可的平均交并比值（50% 或 0.50）进行比较，并动态调整阈值，以确保全局模型的鲁棒性。

在考虑全局模型平均交并比的实证评估中，我们发现，对于独立同分布设置，FedBal 的性能比 FedAvg 提高了约 3%。对于

非独立同分布设置，我们发现，在最终联邦轮次（第 20 轮）中，其性能与 FedAvg 相近。不过，FedBal 在非独立同分布设置下的平均性能优于 FedAvg。对于非独立同分布且不平衡的设置，在第 20 个联邦轮次中，FedBal 的性能分别优于 FedAvg、FedProx 和 FedSGD。尽管 FedSGD 的性能与 FedBal 相近，但其不确定性高于 FedBal（图 6.6 (c)）。在类别不平衡强度方面，我们发现，在三种情况下的最终联邦轮次（第 20 轮）中，FedBal 的表现优于 FedAvg。不过，对于高类别不平衡强度（每个工作节点仅包含一个类别），FedBal 的改进不太显著（0.25%）；而对于低类别不平衡强度（每个工作节点包含三个类别），FedBal 则表现出显著的性能提升（超过 2%）。随着联邦工作节点数量增加的实验表明，增加联邦工作节点数量可以提升 FedBal 的性能（最高达 2%）。由于时间约束和网络漏洞，我们无法扩大实验规模，尤其是在增加工作节点数量和类别不平衡强度的情况下。最后，FedBal 方法中联邦学习工作节点权重对全局模型的影响呈现出正相关关系，验证了选择方法的可接受性。机器学习训练参数（每个联邦轮次的训练轮次、优化器、批量大小）可以进行更细粒度的调优，以利用 FedBal 方法探索改进空间；这被视为本研究的未来工作。因此，我们未来还计划为深度学习的语义分割领域开发自定义损失函数，并将我们的方法（FedBal）增强为可用于

顶部　　以及提升机器学习模型鲁棒性的联邦学习算法　　l.

第 7：工业 4.0 中智能解决方案的威胁和副作用。 0

7.1 概述

新型物联网技术、云计算系统、改进的无线网络和机器学习解决方案的收敛，使大规模信息物理工业系统得以顺畅运行。尤其是无线连接，它改变了运营模式，使得如今几乎所有生产活动都可以通过各种传感器和执行器进行远程管理。此外，这些技术显著提高了各类复杂工业运营的产量和效率。然而，数字化和智能化范式转变并非全是好处！我们也需要考虑一些弊端——数字化转型和无处不在的连接带来了弱点，犯罪分子可能利用这些弱点发动网络攻击，从而危及工业生产、分布，甚至安全。正如我们在近期的多个事件中所看到的，例如 Colonial Pipeline [126],阿姆斯特丹—鹿特丹—安特卫普（ARA）网络攻击 [127],以及挪威能源公司 [128],的案例，恶意软件系统（即恶意软件）能够接管系统控制，并阻止其正常运行，直至入侵者的要求得到满足。事实上，这些近期的网络攻击已经证明，无论从影响还是严重程度来看，网络攻击都可能与物理攻击一样有害。

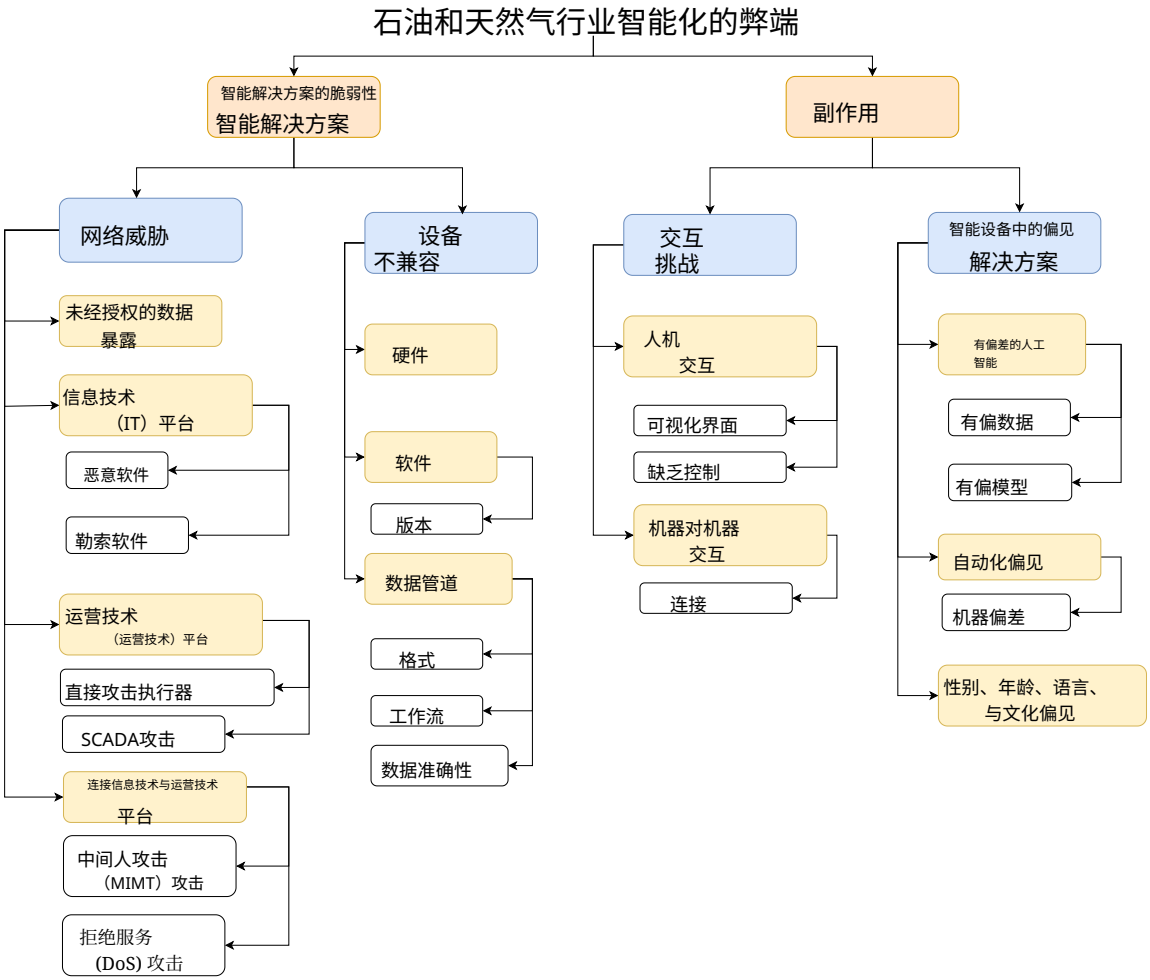
石油与天然气等智慧行业面临着独特的安全挑战，只有通过对整个系统进行深入研究和诊断才能加以应对。然而，

由于实施复杂性高，智慧行业尚无针对性的安全解决方案。这种安全真空使近年来无数网络攻击得以肆虐，危及全球各地的人们和社区。因此，鉴于这些行业的重要性以及它们之间在管理和技术方面存在的差距，亟须在工业4.0革命中研究适用于智慧行业的全方位安全解决方案

.

随着石油和天然气行业日益复杂和数字化，我们正在考虑研究智能石油与天然气领域中存在安全风险的关键方面。上游、中游和下游部署的大规模互联“物”（物联网设备）网络，对整个石油和天然气行业构成了重大安全风险。预测性维护和现场工人安全只是两个例子，说明处理智能传感器生成的大量实时、现实世界数据，能够带来哪些效率提升。使用连接互联网的设备可能会危及石油与天然气基础设施的物理安全和安保。例如，配备目标跟踪功能的互联摄像头、地理围栏周界防护解决方案、第三方渗透以及其他访问控制系统，都可能导致运营现场发生安全漏洞。因此，本论文部分聚焦于智能解决方案对工业4.0造成的不利影响，以及最大限度减少这些影响的策略。

图7.1. 该分类体系以方框流程图的形式呈现了采用先进技术实现的智能解决方案的缺点。主要分为三个层级，分别以橙色、蓝色和黄色标示。白色方框表示其上级方框中的不同类型（示例）。



7.2 智能石油与天然气行业中的网络威胁分类体系及副作用——

行业

为了系统地探讨油气行业中智能解决方案的各种缺点，我们构建了一个分类体系，如图7.1所示。在该分类体系中，我们将智能解决方案可能带来的缺点分为两组：漏洞和副作用。漏洞部分探讨网络威胁

以及智能油气系统中由设备不兼容导致的挑战，重点关注石油和天然气行业中的软件、硬件、基础设施和数据相关漏洞。另一方面，副作用类别关注因与智能解决方案的交互（例如，人机交互和机器间交互）以及智能系统中的偏见而产生的困难。

图7.1对已实施或预计近期实施的各种智能解决方案的缺点进行了分类。该分类体系为本章提供了蓝图，使读者能够跟踪复杂的智能解决方案及其伴随的结果。因此，在接下来的部分中，我们将逐一探讨分类体系的主要部分，以了解智能解决方案缺点的严重程度。

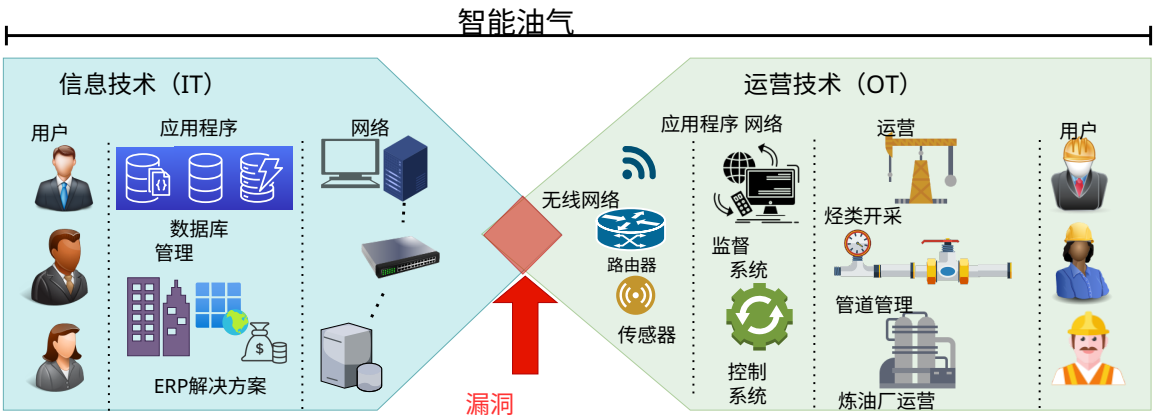
7.3 信息技术与运营技术交互作用引发的漏洞

智能石油和天然气行业的技术运营分为两个关键技术平台：信息技术（IT）和运营技术（OT）。图7.2概述了一家石油和天然气公司的IT和OT组件。如图所示，IT组件主要负责在公司内部传输数据和信息。由于其运营环境主要涉及业务逻辑，IT组件经常访问外部网络。相比之下，OT组件负责石油和天然气生产中的物理过程操作，以及执行这些过程所需的机器。因此，网络窃贼主要以IT和OT平台为目标，以满足自身需求。传统上，与

OT平台相比，IT更容易受到攻击，因为IT有许多开放窗口（例如，操作系统、电子邮件服务器、直接通信应用程序），攻击者可能会加以利用。

另一方面，OT平台主要处理直接的石油和天然气生产及加工活动，外部访问受限。值得注意的是，IT和OT平台的连接点经常成为网络攻击的目标，系统架构师必须对此进行有效处理。此外，随着物联网的发展，基于复杂计算技术的智能物联网解决方案正在逐步开放对OT平台的访问。因此，我们将探讨这两个平台中的漏洞范围及其重叠部分。

图7.2. 智能石油和天然气公司采用不同的网络来运行智能油气行业的整个运营体系，其信息技术（IT）和运营技术（OT）平台各司其职。IT 平台与业务应用和油气行业的财务方面密切相关，而 OT平台则直接涉及石油或天然气开采及生产运营。IT 和 OT 平台在某些节点相互连接，为网络攻击者渗透整个系统创造了可乘之机。



OT平台由积极参与石油终端产品生产的技术构成。这些活动包括开采、炼化、管道运输、生产、控制和监控系统。另一方面，

油气信息技术领域主要涉及金融、数据库管理、数字资产管理，以及利用不同计算机平台和通信协议开展的其他业务运营。在这种情况下，运营技术实体提供石油终端产品，而信息技术实体则通过利用运营技术实体的产出来开拓商业前景并制定金融政策。因此，与运营技术实体相比，油气公司内部网络中的信息技术实体与外部网络的联系要密切得多。由于这种关系，石油公司可能成为勒索软件及其他网络攻击的受害者。

运营技术（OT）传统上处于“物理隔离”环境中，不与公共网络或其他数字技术连接。几十年来，传统运营技术一直依赖计算机监控或改变系统的物理状态，例如使用 SCADA系统监控和控制设备，以提高运营效率。传统运营技术安全主要包括简单的物理任务，例如确保机器正确执行相同的操作，以及确保装配线持续运行。尽管如此，近年来工业4.0的兴起改变了传统的运营技术环境。企业开始在其网络中部署新的数字化解决方案，通过添加能够更有效地收集数据并接入网络的“智能设备”，提升自动化水平。为实现这种连接，并在运营技术数据生成时对其进行处理和分析，信息技术与运营技术系统因此实现了集成。这种技术范式变革（称为信息技术与运营技术融合 [129, 130]）带来了新的可能性，开拓了新的应用场景，但也扩大了网络安全漏洞的范围

例如，Colonial Pipeline 遭受的攻击 [131]表明，密码管理不善可能会危及全国最大的汽油管道。黑客找到了一个旧的但仍在使用的 VPN账户密码。鉴于这一威胁，油气公司应制定严格的网络安全防护措施，包括员工培训。震网病毒 [132] 是首个引起广泛关注的针对工业控制系统（ICS）的专门攻击，尽管它并非针对工业环境的首例网络攻击。震网病毒是一种计算机蠕虫，被指控在伊朗核计划中制造混乱，损坏了该国 20% 以上的核离心机。此后，针对工业组织的网络攻击逐渐增多，波及电网（Industroyer）、能源（Black Energy）、石化行业（Havex）以及石油和天然气（Havex）（TRISIS）等众多行业。黑客入侵工业网络，目的包括关闭机器、索要赎金和窃取数据。

7.4 智能油气行业中的网络威胁

石油和天然气行业面临的挑战在于，其系统设计必须考虑网络连接。例如，工厂在设计之初从未考虑过连接网络。然而，随着数字革命不断发展，如今工厂已接入网络。这可能造成危险局面，因为针对这类系统的网络攻击会破坏运营，甚至造成生命损失。在网络安全方面，石油与天然气行业落后于其他行业。尽管网络安全对企业可持续性至关重要，许多公司仍需要

在可靠系统上投入更多资金。本节其余部分将讨论石油与天然气行业面临的一些安全问题。

7.4.1敏感数据的漏洞

存储在工业物联网设备（传感器和执行器）上的敏感信息必须通过严格的安全协议加以保护。因此，油气公司如今会定期检查从各种来源收集的私人信息。以下是此类数据源的一些示例：

- 历史石油和天然气勘探、输送及定价数据
- 人口统计数据
- 招聘信息的响应数据
- 网页浏览模式（在信息网站上）
- 社交媒体
- 来自运营系统的传统企业数据
- 油气钻探、勘探、生产、交通运输和炼化过程中的传感器数据

上述内容是任何私营公司都高度机密的信息示例。由于石油和天然气行业竞争激烈，一家公司的各类机密信息可能会成为其竞争对手觊觎的宝贵资产。因此，道德操守存疑的黑客越来越专注于获取这些敏感记录。

7.4.2智能系统的脆弱性

在前面的章节中，我们介绍了助力工业4.0发展的智能解决方案。尽管这些技术令人着迷且代表着未来，可能有助于整个石油和天然气行业，但我们也应认识到它们的弱点。以下是智能解决方案可能失效或遭到入侵的一些方式：

机器学习方法中的固有偏见：训练数据集的质量对任何机器学习模型的成功都至关重要。有偏数据集是指以某种方式进行选择，使某些类型的示例比其他示例获得更大权重的数据集。例如，假设用于训练无人机管道泄漏检测模型的照片数据集主要涵盖晴朗天气场景。在这种情况下，训练好的模型在雨天或雪天的表现会很差。在使用某些条件下特定品牌设备的数据训练模型时，也可能采用预测性维护。因此，该模型未能正确泛化此前观察到的数据。因此，在将其用作预测性维护模型之前，还需要提高其准确性。**机器学习模型中存在不确定性：**机器学习模型容易受到其固有歧义的影响。然而，该模型可能会产生假阳性或假阴性结果，这可能带来灾难性的后果。例如，如果炼油厂的智能火灾探测系统未能发现火灾，可能会在很短时间内造成严重损失。

智能应用工作流故障：智能解决方案通常由许多协同工作的部分组成，共同构建有向无环图，或

DAG。例如，在石油钻井平台上进行人脸检测，需要拍摄视频、提取帧，然后逐帧进行分析。在任何环节中断此类智能应用的运行周期，都可能导致整个应用程序失效，使其易受攻击。同样，如果未向执行器发送命令，整个工作流可能会停用，从而导致失去对系统的控制。

7.4.3信息技术（*IT*）恶意软件与漏洞

恶意软件是“恶意软件程序”（malicious software）的简称，指网络犯罪分子（又称“黑客”）编写的任何侵入性程序，旨在窃取数据以及损坏或摧毁计算机和计算机系统。恶意软件包括病毒、蠕虫、木马、间谍软件、广告软件和勒索软件。近期的恶意软件攻击导致了大规模数据泄露。因此，必须迅速识别恶意行为者，以清除恶意软件。在众多恶意软件类型中，以下段落将介绍四种主要类型。

病毒：为了感染其他计算机，病毒通常会附着在能够运行宏的文件上。病毒会潜伏在下载的文件中，直至文件被打开。病毒是一种会干扰系统正常运行的恶意程序。这意味着感染可能会中断运营并导致数据丢失。

蠕虫： 蠕虫是一种恶意软件，能够迅速自我复制，并感染网络上的任何系统。与病毒不同，蠕虫无需借助任何宿主软件即可传播。例如，蠕虫可以通过文件下载或网络连接感染设备，然后迅速复制并扩散。蠕虫，

例如 vi 毒，可能会严重影响设备的功能并导致数据损失 .

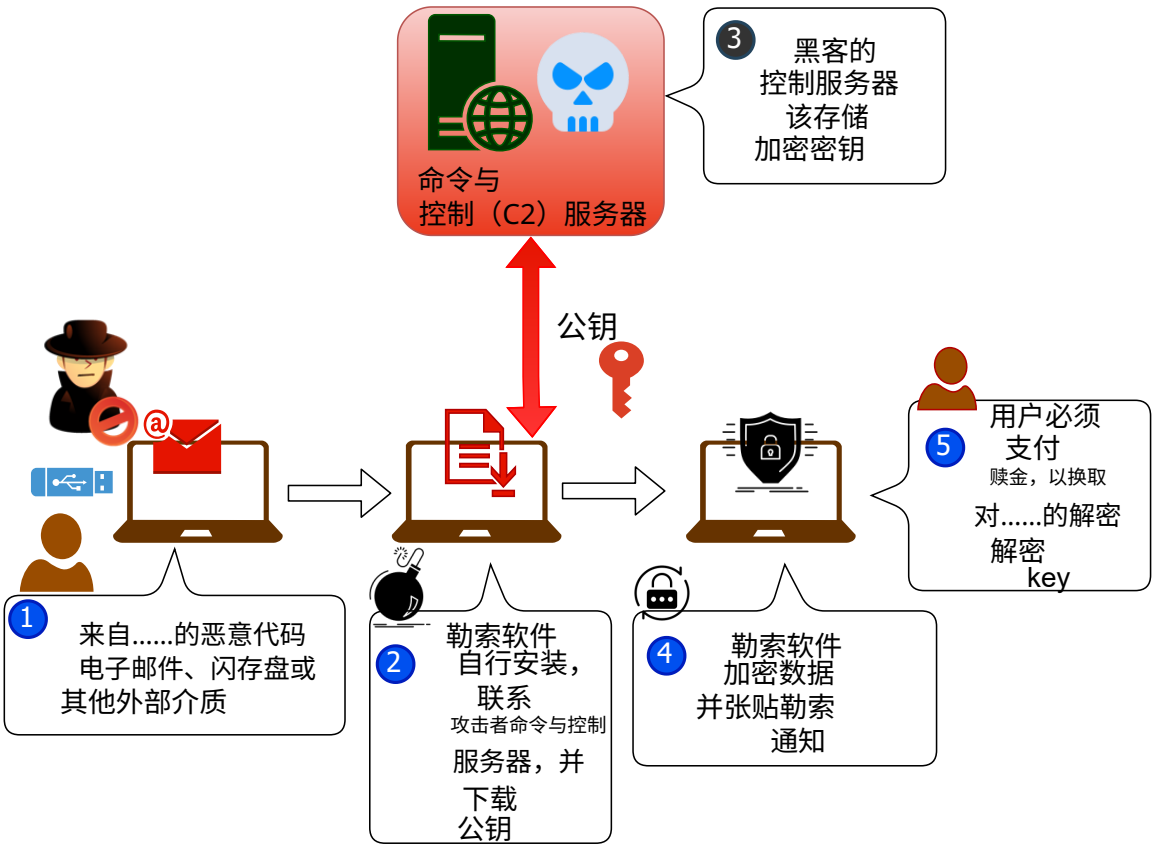
木马：通常，木马恶意软件可能会伪装成看似有价值的软件。然而，一旦下载，木马病毒就可能访问用户的私人信息，并对其进行更改、阻止访问，甚至将其删除。因此，设备功能可能会受到严重影响。与常见的病毒和蠕虫不同，木马病毒不会被编程为自我复制。

间谍软件：间谍软件是一种恶意软件，会在计算机上秘密运行，并将数据发送回外部来源。间谍软件尤其危险，因为它会影响设备性能、窃取敏感数据，并允许潜在的攻击者远程访问。间谍软件通常以财务或个人数据为目标。例如，键盘记录器是一种间谍软件，会记录用户的击键，以窃取密码和其他机密信息。

勒索软件：勒索软件是一种恶意软件，会侵入系统并加密其中的数据，使用户无法访问，然后以解密数据为条件索要付款。勒索软件的使用通常与网络钓鱼方案有关。图7.3展示了一次真实勒索软件攻击的各个阶段。如图所示，在这类攻击中，受害者会因误点伪造链接而下载勒索软件。随后，攻击者使用只有自己知道的加密密钥加密目标数据。最后，黑客会以收取钱款为条件释放信息。鉴于勒索软件在过去几年日益猖獗，尤其是在石油和天然气行业，我们将进一步深入分析这一威胁。

7.4.3.1 勒索软件攻击事件。在定向网络攻击中，可能会使用单个病毒来窃取数据、传播和渗透等。威胁行为者的目标是在受害者网络中维持持久化。因此，他们必须不断与病毒通信并对其进行更新。通过 DNS 协议，一个称为 DNS 隧道的过程 [133] 在恶意软件与控制器之间传输信息。此外，电子邮件和云服务极大地拓展了现代通信的范围，为勒索软件犯罪分子打开了广阔的可乘之机。

图7.3。 勒索软件从开始到结束的结构。勒索软件客户端通过恶意电子邮件或其他外部媒介进入 IT 平台。随后，客户端与黑客的命令与控制服务器通信，以下载加密密钥。用户数据由勒索软件客户端加密，最后发送勒索通知。



过去，黑客会利用垃圾邮件僵尸网络尽可能多地入侵系统并传播勒索软件。reference mandal2020digital。尽管勒索软件一直是所有拥有数字文件的人面临的重大问题，但随着犯罪分子开始专门针对企业发动攻击，并可能对运营造成毁灭性影响，这一问题变得更加严重。以下是一些最引人注目的勒索软件攻击案例：

BitPaymer19： BitPaymer19 [134] 是一种尤其致命的勒索软件，最近攻击了一家提供油井钻探服务的美国公司。BitPaymer 攻击者通常会利用钓鱼邮件，先用恶意软件感染受害者，之后再横向渗透网络，以窃取其他敏感数据。当 IT人员无法工作时（例如， 在周末和节假日），勒索软件攻击会

APT33： 某个广为人知的攻击者组织主要将目标集中在石油行业及其供应网络。能源行业中，与石化制造业和航空业有关联的组织均遭到攻击；APT33在这些领域既参与军事活动，也参与商业活动。APT33还攻击了欧洲和亚洲的能源公司。2018年10月至2018年12月，并延续至2019年期间，一台 Powerton 命令与控制服务器托管在命令与控制网站 timesync.com 上，并与少量属于石油公司的IP地址通信。2019年11月底至12月初的三周内，一家欧洲石油公司在印度运营的数据库服务器与一台由APT33使用的 Powerton 命令与控制服务器通信。2018年秋季，有人发现，一家

总部位于英国、为炼油厂和石化厂提供专业服务的大型公司可能已遭到APT33入侵。

电子邮件网络钓鱼是 APT33 最常用的渗透方法。多年来，这种骗局一直依赖同一种诱饵：一封看似合法、实则以提供工作为诱饵的鱼叉式网络钓鱼电子邮件。其他攻击活动则针对航空业和石油行业的招聘流程 [135]。此外，电子邮件中还提供了指向恶意“.hta”文件的链路。为了进一步扎根于目标网络，APT33 可能会利用随“.hta”文件下载的 PowerShell 脚本，下载更多恶意软件。

7.5 不兼容的物联网设备

智能石油与天然气行业最常见的漏洞之一，是使用彼此不兼容的物联网（IoT）设备，如图7.1所示。事实上，智能工业（例如石油和天然气行业）背后的真正引擎，是自动化系统：它们从各种各样的物联网（IoT）设备和传感器获取数据，使用机器学习或统计模型处理数据，再通过各种执行操作落实决策。实际上，传感器和其他相互连接的物联网设备由众多公司制造和采购，因此本质上具有异构性。这种多样性可能造成不兼容，可能被网络攻击者利用，也可能在危机期间导致服务中断。为了有效传输数据，并在紧急情况下（例如检测有毒气体）提供实时通信，配置相互连接且兼容的物联网设备至关重要，这些设备应能实现无缝交互。

由于采购统一且完全兼容的物联网设备十分困难，甚至不可能，研究人员正在探索其他解决方案，例如开发标准协议，以实现所有工业物联网设备之间的有效通信。为此，领先的 IT 公司正携手合作，创建一种单一协议（称为 Matter 协议 [136]），使其兼容所有物联网设备。该协议是一种用于智能家居之间网络互联通信的新协议，由连接标准联盟推动，该联盟成员包括苹果、谷歌、Amazon 等科技巨头。问题在于，缺少一种基于成熟可靠技术的标准基于 IP 的通信协议，来构建安全可靠的物联网生态系统。

在智能油气及其他智能工业场景中，我们可能会考察三种不同类型和程度的不兼容。接下来的章节将讨论存在的不兼容问题：硬件层面、软件层面以及数据管道层面的不兼容。

7.5.1硬件层面不兼容

石油和天然气行业越来越多地使用市面上常见的产品来替代专用设备。由于适应性强，这些产品比传统的过程控制系统更容易受到安全问题的影响。由于它们被广泛使用和部署，攻击面显著扩大。攻击油田的方法有很多。例如，可以使用智能实时摄像机监控

可能存在危险的区域，以进行异常检测。不过，未授权用户可能会利用控制系统打开阀门，使有毒或爆炸性气体泄漏。如果传感器、执行器、摄像头及其配套硬件都使用同一种协议来检查漏洞并标记异常活动，就可能免受此类攻击。

7.5.2软件级不兼容

软件层面的兼容性问题可能源于使用过时或不受支持的软件，从而导致系统故障。此外，恶意病毒可能通过第三方软件进入内部系统。过时的软件必须更新，才能与现代硬件和应用程序兼容。如果系统没有及时更新，或使用了并非为其操作系统设计的增强功能，就更容易遭到攻击。

石油和天然气行业的公司通常会假设所购买的数字产品是安全的，并且能够集成到更庞大的系统中。然而，他们通常会确认系统中的其他部分是否与新组件兼容。此外，油气公司可能并不总是有足够的资源来核查软件层面的不兼容问题。网络犯罪分子正是通过“智能”技术中的漏洞，侵入油气公司的专用网络。

由于物联网（IoT）的智能化，石油供应链中的企业领导者必须提出防范

网络安全问题的新方法。近期发生的黑客攻击利用了软件中的漏洞。2017年，一种名为 NotPetya 的网络攻击袭击了众多机构，包括一家电力供应商、银行、公共交通网络以及一家大型国际集装箱航运公司。有趣的是，该病毒通过乌克兰公司的更新版会计软件传播。当感染扩散至其他计算机时，便导致系统崩溃；在此案例中，网络犯罪分子用一种名为“SolarWinds”（2021年）的间谍软件感染了面向客户的认证软件。在这两个案例中，黑客都利用软件漏洞进入互联供应商系统。此外，他们还设置了可能被用于窃取知识产权财务数据或在用户设备之间传播恶意软件的漏洞。

7.6 利用区块链应对智能油气中的网络威胁

区块链技术近来因成为比特币和以太坊等加密货币的基础而备受瞩目，也是一种有效的安全方法。数据以区块链的形式存储，各区块依次相连，并在每个区块中保存前一区块的哈希值。由于遭到篡改的数据区块的哈希值将不再与后续区块中的哈希值一致，因此可以追踪攻击行为。如今，工业4.0和智能油气领域的多个子系统都在使用区块链技术。

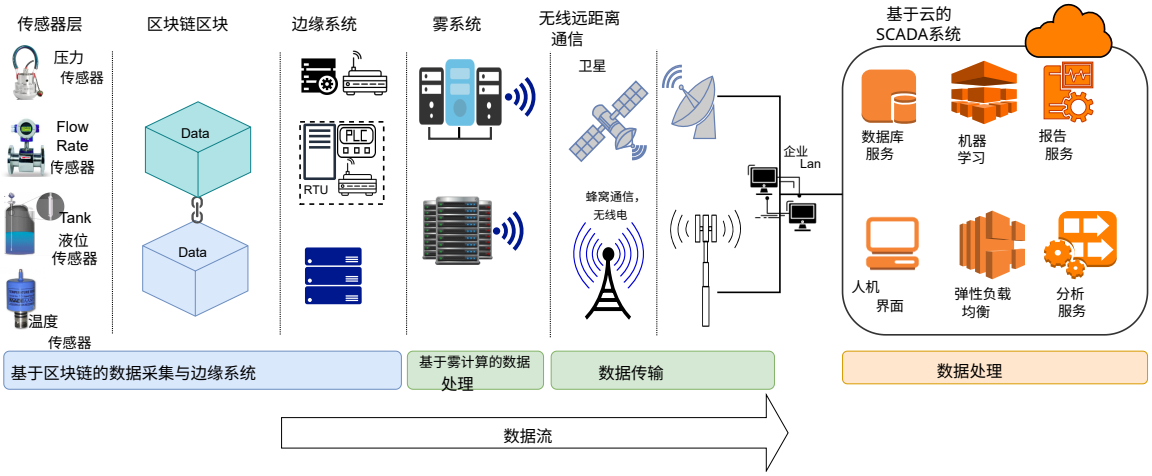
7.6.1基于区块链的控制系统（*SCADA*）

工业4.0运动改变了现代工业中信息技术和运营技术的作用。SCADA系统从智能物联网设备收集信息，并将其发送到服务器进行分析，构成了大多数

OT平台的支柱。然而，这种数据采集方式本质上既不安全也不可靠，为黑客提供了可乘之机。因此，人们提出了基于边缘计算与雾计算的区块链安全程序，以保护SCADA系统的数据采集交易。收集到的传感器数据会在被基于云的SCADA系统处理之前加密并存入数据块中，这种方法的总体概览如图7.4所示。每个数据块中都存储着前一个区块和当前区块的数据哈希。随后，数据聚合器（DA）和所有中继服务器共同参与区块验证过程。服务器会多次响应，以供验证。在就该区块的合法性达成共识后，DA会将请求转发给所有参与的服务器。DA将新区块添加到区块链中，然后成功地将更新后的区块链发送到指挥中心。近期一篇论文提出了挖矿节点选择技术，以及一种与工业4.0兼容且更安全的共识流程 [137], ，以下各节将对此进行讨论。

7.6.1.1 共识机制。简单来说，共识机制是指验证者/矿工在将新发布的区块添加到区块链之前，验证其真实性的过程，从而维护网络的完整性。各种区块链网络投入了大量时间和能源来开发各种共识技术。公有链和私有区块链都使用某些共识机制。工作量证明（PoW）、权益证明（PoS）、委托权益证明（DPoS）、实用拜占庭容错（PBFT）、

图7.4。 石油和天然气公司端到端SCADA系统中的基于区块链的数据传输。区块链使数据在传输过程中能够加密，以便进行处理，从而提高数据安全性，即使数据在传输过程中被劫持也是如此。



权威证明（PoA）和 RAFT [138]都是常见共识技术的示例。每种共识技术都有各自的优点和缺点。例如，PoW 对新参与者并不公平，因为其计算成本高昂，而且有利于最富有的验证者。另一方面，DPoS 的稳健性和去中心化程度较低。由于缺乏匿名性，PBFT 仅限于许可型（非公有）区块链 [139]。

7.6.1.2 挖矿节点选择。通过运行区块链软件并促进数据传输来参与区块链网络的机器称为“节点”。网络中的节点可以是笔记本电脑、手机或路由器等。“挖矿节点”是参与区块链交易处理和验证的节点。区块链网络中的任何参与者都可以选择参与挖矿过程。术语“挖矿”指的是向区块链添加新交易的活动。图7.4展示了数据

聚合器（DA）边缘服务器负责收集数据、处理数据，并协调挖矿节点的选择与验证。如果您希望尽可能节省时间和处理电力，请将 DA 部署在与中继服务器相同的专用网络中。因此，图7.4所示数据预处理阶段使用的雾服务器托管在 DA 自己的专用网络中

在 [137], 中，作者提供了一种专门用于选择挖矿节点的方法。首先，DA 服务器向中继服务器发起数据请求。DA 收集来自各个中继的所有读数后，会生成一个随机数并将其广播到整个网络。为了统计随机数出现的次数，中继服务器会对各自的数据进行哈希处理，并比较结果。此时，DA 服务器的统计数据与其他所有服务器的统计数据完全相同。最后，每台服务器都会投票选出在这一过程中随机数出现次数最多的服务器。如果所有其他中继服务器都同意，DA 服务器便会选择计数最高的中继作为当前周期的挖矿节点。另一方面，假设有大量中继主机的计数都达到最高，或者所有计数均为 0。此时，DA 会通过密码学安全流程随机选择挖矿节点 [140]。

7.6.2区块链助力工业物联网建立信任

信任问题是工业物联网（IIoT）安全面临的障碍之一。传统的公钥基础设施（PKI）设计建立在单一信任根之上，在这种异构、分散的物联网环境中运行效果不佳，而该环境可能受到多个管理

域的管辖。因此，对于此类场景，有必要构建一种分布式信任模型，在现有信任域之上运行，并在物联网设备之间实现端到端信任，而不依赖单一信任根。因此，建立一种集成信誉系统的基于信用的区块链可能会有所帮助 [141]。

区块链在石油和天然气行业的另一种潜在用途，是存储操作安全关键型工业机械所需的资质凭证。例如，员工和承包商的资质，如硫化氢培训、急救和焊接资质，可以安全地记录并保存在公司的区块链网络上。将此类信息存储在区块链网络中 [142]，所有成员都可以随时验证资质凭证和标准操作程序。

7.7 工业物联网智能解决方案的风险

随着技术不断进步，越来越多的行业和产品接入互联网，了解工业物联网部署的风险至关重要。任何希望在制造业或工业领域使用物联网，或连接现有技术以实现自动化远程监控的企业，都应权衡其优缺点。下一节将讨论智能解决方案可能存在的性能不足问题。

7.7.1人机交互问题

工业物联网已经取得了长足进步；如今，机器能够自动处理来自联网设备的数据。此外，各种自动化传感器和执行器（例如摄像机、智能眼镜以及带音频

输入/输出的自动阀门）已投入使用，以协助或取代人工工作者，确保生产顺利运行，并确保工作节点在使用不同机器完成作业时的安全。

图 7.5. 从感知到控制操作的人机交互工作流程。

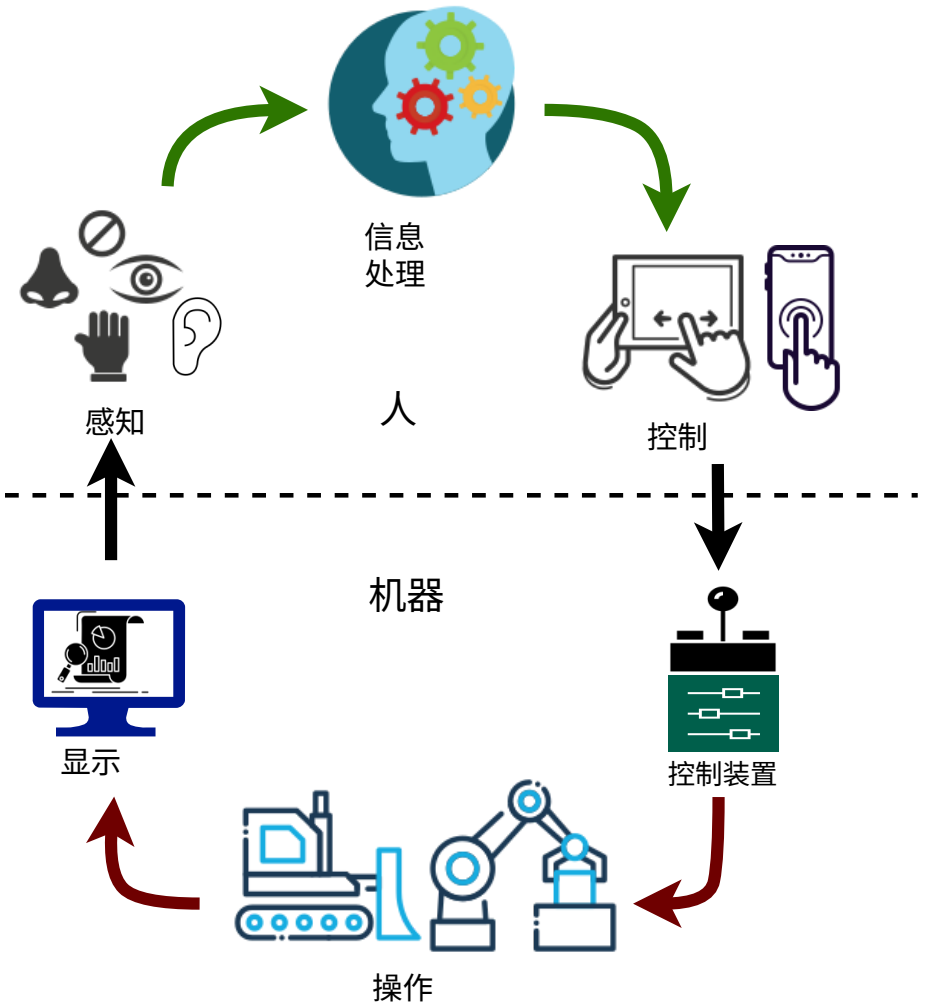


图 7.5 展示了人机交互中人和机器两侧如何

交互协作。如图所示，人们利用不同的感官（如视觉、嗅觉和听觉）查看机器的结果。因此，人工工作者通过信息处理来运行或控制机器。接着，机器执行它们的

作业，并将结果呈现给人们，让他们理解结果的含义。人与机器协作的整个过程称为人机交互过程。

如果相互依赖的机器无法按预期运行，或不便于用户使用，工作现场的生产和安全可能会受到威胁。为了完成重要的工业任务，或者更重要的是，为了“强调凌驾于智能系统选择之上的决定权”，人们往往需要通过输入/输出设备与计算接口进行交互之外的其他人机交互。以派往政治敏感地区（如边境地区）执行自主油气勘测的无人机或遥控无人潜水器为例。然而，其算法效率低下或出现故障，可能导致它勘测指定区域以外的地区，并使操作员无法控制勘测路线。这类人机交互故障可能会在政治或军事层面造成意想不到的后果。

7.7.2机器对机器交互问题

机器在联网自主系统中相互通信，以完成各种活动。在这些系统中，利用多个设备执行自动化动作序列；如果出现问题，可能是由于（A）设备产生了误导性输出（例如，异常检测错误导致自动阀门关闭，或误报警导致自动关门困住现场工作人员），或者（B）设备之间存在不兼容。在采用不可容错操作的生产环境中，机器对机器接口问题可能导致事故或灾难事件。

图7.6。 压缩机组内一处封闭的石油生产区发生小型火灾事故。火灾报警器启动，洒水喷头开始喷水，导致发电机发生电力故障，电动门因此锁闭。不幸的是，工作节点当时正在进行管道维护，因门关闭而被困在设施内。在此，机器间交互造成了现场工作人员的安全问题。

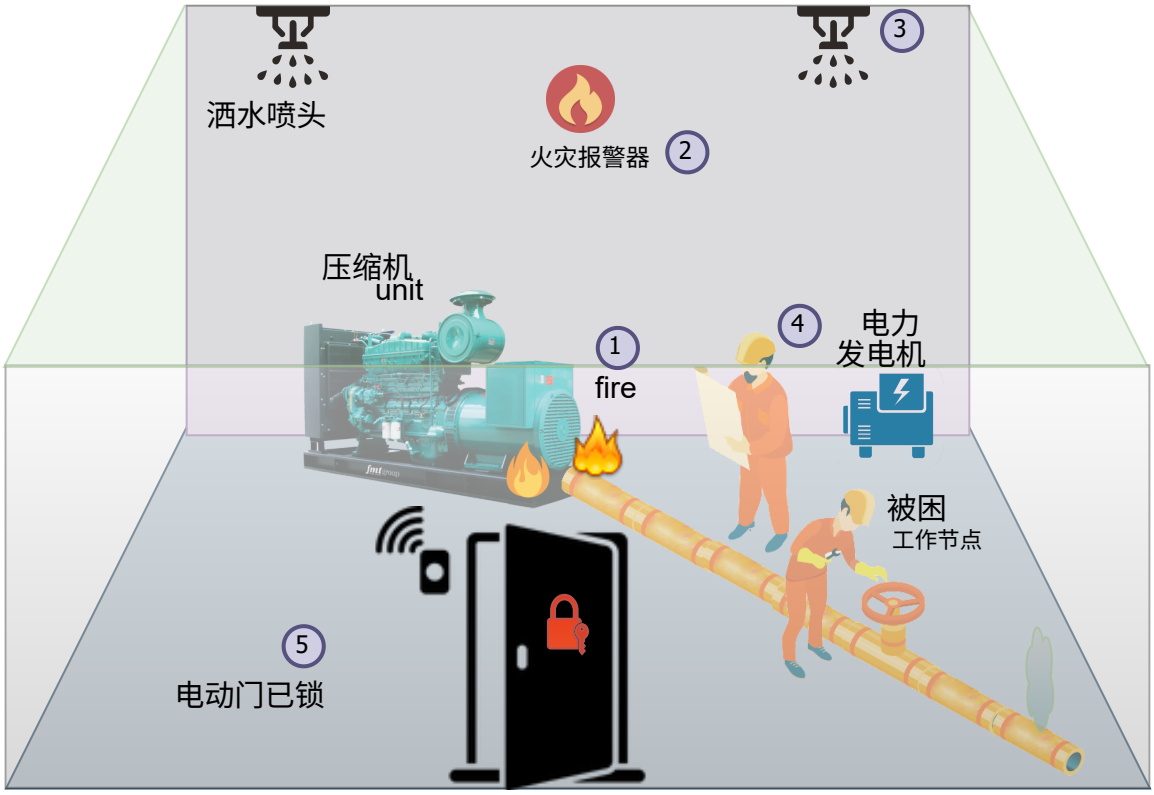


图7.6展示了一个情景，说明了.....的影响

机器对机器交互问题。考虑管道维护以及在生产场景中与众多设备进行通信的情况。设想一条管道连接到一台机器（例如蒸馏装置），该管道位于需要维修的封闭空间内。因此，当发生火灾危险时，维护人员在封闭空间内工作。作为预防措施，气体传感器检测到烟雾（图7.6中的步骤2），启动洒水喷头（步骤3），并向控制器发送警报。发电机因洒水喷头的水而关闭

（步骤 4）。为保障外部员工的安全，控制器立即发出指令，让自动门关闭（步骤 5），同时忽略区域内的工作节点。在此情境中，控制器无法识别设施内的人员，因此针对外部工作节点执行安全操作，却使内部工作节点陷入危险。由此可见，机器对机器交互方面的问题有时会导致此类情境，必须通过评估适用于油气行业的智能解决方案来解决。

7.8 智能工业中的偏见

软件开发不同阶段的人为干预可能引入各种偏见，从而损害原本智能的软件解决方案的质量 [143]。过去，偏见曾导致事故和潜在的危险情况。偏见有多种类型，包括基于年龄、种族、性取向、残疾及其他人口统计特征的偏见。此外，许多现场团队成员（例如，工作节点、工程师和协调员）依赖容易出现上述缺陷的软件工具和仿真。本文将讨论不同类型的偏见，以及它们可能对智能油气业务乃至更广泛领域造成的损害。

7.8.1人工智能（*AI*）解决方案导致的偏见

尽管人工智能系统已在多个领域展现出革命性的作用，但它们容易产生以下两种偏见：

- 训练数据集中存在一些缺口，导致模型在某些输入上的表现不佳。

- 模型的偏见与用于训练的原始数据集中的偏见相同。

训练数据集缺失是人工智能偏见的一个来源，商业人脸识别系统就体现了这一点。研究人员发现，训练数据集中缺少深色皮肤女性 [144] ，这是导致该人脸识别系统对其白人男性的准确性达到 99.9%，而对有色人种女性的准确性仅为 35.0% 的根本原因。然而，问题在于，“大数据”并不一定能提供有效且可信的模型。例如，社交媒体是一个成熟的大型数据集来源。据 [145]报告，互联网青少年中只有 24% 使用 Twitter，因此应谨慎看待从 Twitter 数据中得出的结论。

不公平的模型不一定会在人口群体子集上的表现较差。即使模型给出的结果正确，它仍然是不公正的。在这种情况下，数据集存在偏差，而模型会重复或放大继承偏见。例如，自然语言处理（NLP）模型通常使用大量由人撰写的文本语料库进行训练（例如，新闻文章）。然而，人们发现，使用 Google News 文章训练的词嵌入反映了女性／男性的性别刻板印象。例如，模型会回答说，父亲是医生，母亲是护士；或者“男人”是“计算机程序员”，而“女人”是“家庭主妇”。当模型因不公正的流程或结构而使用有偏差的数据进行训练时，就会产生这种偏见 [146]。

A人工智能偏见的另一个例子是 Yelp 的餐厅评论系统。餐厅 s

可以付费请 Yelp 在其平台上推广自己的门店，但这必然会影响有多少人看到某家餐厅的广告，进而影响谁决定去那里用餐。因此，Yelp 的评价可能会不公正地偏向知名度更高的餐厅。

7.8.2工业*4.0*智能解决方案中的自动化偏见

该领域最受尊敬的心理学家之一、伊利诺伊大学芝加哥分校的 Linda J. Skitka 将自动化偏见定义为：“在高度自动化的决策场景中，当许多决策由自动化辅助工具（如计算机、物联网设备和智能手机）处理，而人主要负责监控正在进行的任务时，个体往往会犯的一类特定错误。”

倾向于使用自动化辅助和决策支持系统，通常称为“自动化偏见”。2010年7月26日，Enbridge输油管道破裂 [147] ，大量原油流入卡拉马祖河和塔尔马奇溪，自动化偏见是造成这一事件的重要因素。事实证明，自满和倾向于依赖自动化都对Enbridge输油管道灾难产生了重大影响。因此，企业、政府和监管机构在设计系统以降低因疏忽而犯错的可能性时，必须考虑自动化偏见。“自动化偏见”指人倾向于选择所需脑力最少的行动。类似的思路也适用于人工智能和自动化的基本原理：从海量数据中学习。这类计算意味着未来状况大体上会保持不变。还需考虑的另一个因素是，错误的训练数据可能会导致错误的

学习 [148] ，而这与人工智能偏见有着隐含的关联。

7.8.3油气行业中的性别偏见

根据一份研究报告 [149]，石油和天然气行业正面临熟练人才短缺的问题，而性别偏见使问题雪上加霜，因为女性工人被排除在招聘之外。研究通过访谈来自世界各地的男女工作节点，并分析他们的评论，得出了相关结论。事实上，石油和天然气行业素有由男性主导的名声。不过，一些石油和天然气企业正努力实现性别平等和员工多样性，另一些企业却任由性别差距扩大。尽管许多企业努力将性别平等纳入政策、行动和程序中，但仍面临性别失衡和各种隐性偏见等挑战。

7.8.4智能油气解决方案中的认知偏差

认知偏差这一新近提出的概念，指的是人类思维和信息处理中的心理缺陷，可能导致不准确或不理性的评估或决策。阿莫斯·特沃斯基和丹尼尔·卡尼曼在1974年发表于《科学》杂志的一篇文章中首次提出了这一概念（Tversky & Kahneman， 1974 [150]）。此后，大量文献探讨了认知偏差及其对人类思想和行为的影响。

根据对认知偏见的普遍理解，它是一种心理缺陷，会导致对外部数据的错误解读，并损害选择和判断的逻辑性与精度。偏见是在无意识中产生的自动化

过程，能够加快并提升决策效能。可能导致认知偏见的因素有多种，包括公众影响和情绪。过去几年，人们越来越意识到认知偏见可能对作业安全造成的威胁。例如，偏离常规、常态化和群体思维等偏见如今已得到广泛认可。此外，2010年深水地平线 [151]调查使公众开始关注认知偏见，至少在海上钻井行业从业者中是如此。因此，国际油气生产者协会（IOGP）强调了这些认知损伤对于安全的重要性。因此，在构建需要人类针对复杂工业运营作出决策的智能自动化解决方案时，现在正是应对认知偏见的最佳时机。

7.9 总结

随着数字化工业4.0革命的到来，油气作业发生了巨大变化，这场革命广泛应用了尖端计算机硬件和软件。然而，这些发展也为网络犯罪分子创造了机会，使其能够更高效地查找信息技术或运营技术系统中的漏洞，或查找两者混合系统中的漏洞。智能技术的异构性与不兼容性，以及它们之间连接困难，也可能成为网络犯罪分子的另一个入侵途径。人机交互和机器间交互的问题，以及随着时间推移而采用的技术之间的不兼容，是其中最显著的

障碍，阻碍了智能技术在传统和智能石油与天然气行业中的广泛应用。尽管工业4.0为石油和天然气行业带来了巨大助益，但该行业的企业高管和专业人士应警惕其智能技术遭到误用。过去几年里，我们痛切地认识到，盲目安装或采用智能技术可能会引发各种各样的风险。研究人员和从业者在决定是否采用智能技术时，必须牢记这些缺点。在这方面，作为本学位论文的一部分，我们出版了一本书[152]，探讨物联网技术的范围及其在工业4.0革命兴起背景下的发展，并对智能油气行业中的智能解决方案及其缺点进行了详细分析。

不断增长的物联网和智能设备（例如，智能网关、传感器、控制器、执行器）会产生大量原始数据，这些数据需要存储、预处理和分析，以挖掘潜在洞见，从而提高工业系统的效率。因此，各种工业4.0延迟敏感型应用基于机器学习（ML）运行，并利用生成的传感器数据实现自动化及其他工业活动。因此，云计算平台一直在提供服务 [153]，以对工业领域不断增长的数据执行各种操作。隐私、中心性和费用一直是有效利用云数据中心的重要制约因素。因此，边缘计算与雾计算将计算服务 [154, 155, 156]带到更靠近终端用户和数据源的位置。然而，受资源限制，边缘设备可能无法满足较大的计算需求。相比之下，雾系统拥有更多计算资源，且其中间件比边缘系统更强大，因此可能更适合满足计算需求。由于雾系统具有异构性，而异构性是导致执行时间出现随机性的因素之一，这会影响工业4.0应用的完成时间。为了针对工业4.0开发稳健的解决方案，有必要研究异构雾系统中各种基于机器学习的应用的执行时间行为。因此，我们对基于机器学习的工业4.0应用进行统计分析，以了解这些应用的执行时间模式。此外，我们还介绍

雾计算系统中的真实世界工业4.0智能应用执行轨迹，这些轨迹可为未来研究工作提供帮助。

尽管雾系统拥有的计算资源多于边缘系统，但灾害发生时计算需求激增仍可能降低性能。因此，在本学位论文中，我们提出将附近站点（由私营公司拥有）的雾计算系统联邦化，以支持此类场景。此外，系统管理员采用与负载均衡方法相关研究中提出的高效资源分配机制，能够使雾联邦概念切实可行。真实世界的工业4.0应用程序在雾计算平台上的执行轨迹，对于制定有效的资源分配方法至关重要。因此，我们利用此前的工作负载轨迹，为工业4.0中的延迟敏感型应用制定了一种适用于联邦雾系统的统计资源分配方法。

此外，工业4.0应用所采用的异构软件方法（例如，单体式、微服务）可能因其多样化的延迟约束而影响雾联邦的执行计划，导致系统性能下降。因此，在超额订阅情形（例如，事故和灾害）下，采用有效的资源分配方法对微服务应用进行分解，可以维持系统性能。相应地，工业计算平台（即，联邦雾系统）应充分考虑微服务工作流应用的随机执行行为、软件结构和延迟要求。我们提出了一种基于概率估计的资源分配方法，将微服务工作流划分到联邦

雾计算系统中，以满足其延迟要求。此外，雾资源联邦化这一概念会使参与联邦的私有雾系统面临数据安全和隐私问题，因为这些雾系统中存储或处理着敏感公司数据。因此，我们提出了一种基于联邦学习方法的数据隐私保护方案，用于在联邦雾计算系统中训练基于机器学习的工业4.0应用。

8.1 讨论

本学位论文的主要目标是研究并开发有效的资源分配方案，利用现代分布式计算系统服务于远程工业4.0。为此，我们首先探讨并识别偏远海上产业中的各种智能计算方面（例如，石油和天然气、矿产、可持续能源），这些领域的计算需求极高，而传统计算系统因延迟约束而效率低下。因此，偏远地区的私有雾系统能够满足工业计算需求。我们进而识别出在异构雾系统中执行的时延敏感型任务所具有的随机执行时间行为。为此，我们探究了异构执行平台（例如，亚马逊云服务、Chameleon）上各种基于机器学习的应用程序的执行时间行为。因此，我们提出了异构计算资源中执行时间的真实工作负载。此外，在偏远工业中，计算需求激增可能会在灾害时期降低雾系统的性能，导致无法按时完成时延敏感型任务请求。为此，我们提出在偏远工业中联邦化邻近的雾系统

并组建雾联邦，以满足激增的计算需求。因此，我们采用联邦概念，并针对超额订阅情形，利用先前合成的真实应用工作负载，开发了一种统计资源分配方法。随后，我们针对工业4.0中广泛使用的单体应用评估了所提出的方案。之后，我们研究智能微服务应用的内部结构，以了解分解对应用工作流完成情况的影响。我们提出了一种概率工作流划分方法，并将其与先前提出的资源分配方法结合，从而提升雾联邦的性能，并确保远程工业4.0中的安全。最后，我们探讨了在共享私有雾系统以开发准确的工业4.0机器学习模型时所面临的数据隐私问题。因此，我们研究了雾联邦中的联邦学习技术，以确保私有雾系统的数据隐私。在此背景下，我们探讨了联邦学习设置中的类别不平衡问题，该问题可能降低全局模型的鲁棒性。因此，我们提出了一种能够抵御类别不平衡问题的联邦学习方法。

在第3章中，我们分析并估算了异构云和雾资源（例如，Amazon、Chameleon）上基于深度神经网络的应用的性能。在此，我们识别了各种工业4.0应用的随机执行行为。因此，我们从两个不同的视角，即以应用为中心和以资源为中心，运用不同的统计工具，探索并建模各种工业4.0智能应用的推理执行行为。

此外，我们介绍了四种不同的工业4.0基于深度神经网络的应用的执行时间工作负载，旨在开发适用于联邦雾系统的稳健资源分配方法。

在第4章中，我们探讨了雾联邦的可用性及其优势。雾联邦可用于支持灾害等紧急情况（例如，火灾、爆炸和石油泄漏）。作为智能工业的一个实例，我们考虑由多个彼此邻近的石油开采场组成的远程智能油田，每个开采场都配备雾计算系统，以满足本地计算需求。尽管如此，在发生石油泄漏等紧急情况时，为了协调多项活动（例如，对石油泄漏进行无人机巡检、处理摄像机图像和传感器数据），计算需求可能会增加。因此，我们提出了一种针对单体式时延敏感应用的概率资源分配方法，该方法利用我们此前的工作负载数据，从联邦中有效选择相关的雾系统。由于该资源分配方法能够根据我们先前的工作识别接收应用的随机执行行为，因此可确保雾联邦的鲁棒性，按时完成大多数接收的工作负载。

在第5章中，我们探讨工业4.0应用的现代软件架构（即，微服务），以在雾联邦上制定高效的执行策略。相比之下，我们发现，采用单体式软件架构的遗留应用仍然存在于各个工业领域。因此，为了满足远程工业的计算需求，执行平台（联邦雾系统）应了解工业4.0应用的软件架构。

因此，对于微服务，我们考虑采用应用拆分策略，以提高按时完成执行的可能性。此外，对于单体应用和单个微服务，我们利用先前掌握的随机执行行为知识，在联邦雾系统之间高效地分配雾资源。基于此，我们提出了一种微服务统计分区和资源分配方法，该方法考虑了底层软件架构以及工业4.0应用的随机执行延迟。

在第6章中，我们探讨了在远程工业4.0中训练基于机器学习的应用程序时，雾联邦的数据安全与隐私方面。在此情境下，我们研究了利用雾联邦训练基于机器学习的石油泄漏检测应用程序的联邦学习技术，以保护该联邦中私有雾系统的数据安全。因此，我们发现，训练数据中的低发生率事件（即类别不平衡）会降低机器学习模型的准确性，并可能对紧急情况造成不利影响。在此，我们提出一种定制的联邦学习技术，针对雾联邦中的类别不平衡问题进行处理，以提升远程工业4.0的安全措施。

在第7章中，我们探讨了工业领域将各种应用程序集成后开发的智能解决方案所带来的缺点和副作用。因此，我们针对油气行业中的智能解决方案，引入了网络威胁和副作用分类体系，从结构上分析这些智能解决方案中的不安全区域。相应地，多个脆弱区域，包括软件和硬件组件、人机交互问题，以及不同

形式的智能解决方案偏见，都将通过高效的韧性方法加以应对，以帮助系统架构师或工业研究人员开发适用于工业4.0的稳健智能解决方案。

总之，我们探讨并研究了各种工业4.0应用的随机执行行为，并介绍了一种现实世界中的执行工作负载，该工作负载已用于我们的资源分配研究。随后，我们探讨了利用私有雾系统满足工业4.0各种计算需求的联邦概念。尤其是在灾害等超额订阅情形下，如果负载得到适当平衡，联邦就能更加高效。因此，我们开发了一种负载均衡方法，以增强联邦在紧急情况中的鲁棒性；这属于我们研究方向中的系统管理层面。接着，我们深入应用层，研究工业4.0应用的各种软件架构（例如，单体式、微服务）。由此，我们发现微服务工作流应用可以通过分解来提升应用工作流按时完成的能力。相应地，我们提出了一种概率式微服务划分与资源分配方法，以提升雾联邦的性能。随后，我们探讨了在训练基于机器学习的工业4.0应用时，联邦雾系统的数据安全与隐私方面。最后，我们指出了智能解决方案中的各种潜在问题；要为工业4.0应用开发高效且鲁棒的智能解决方案，就必须妥善解决这些问题。

8.2 未来研究方向

基于我们在开发资源分配、微服务工作流划分和安全资源共享方案过程中所得的发现，我们确定了一些可进一步拓展的领域，这些领域有助于提升工业4.0的鲁棒性与安全性。还有几个方面可以继续深入研究。

8.2.1面向工业*4.0*应用的联邦雾系统强化学习资源分配

在本学位论文中，我们提出了一种面向雾联邦系统的应用完成时间统计估算方法，以便将工业4.0应用分配到合适的雾系统中。我们对任务完成成功率的估算可能较为粗略，有时会导致接收任务错失截止期限。我们认为，基于强化学习技术运行的资源分配方法 [157, 158] 能够提升联邦雾系统的服务质量。强化学习（RL）作为机器学习的一个重要分支应运而生，因为它能够使自主智能体针对环境中不断变化的条件做出合理决策。因此，可以使用强化学习技术有效应对不确定的执行行为。在这种情况下，可利用强化学习为联邦雾计算系统中的任务卸载和执行分配资源 [159] ，从而提升整体性能。

8.2.2联邦雾系统中的数据局部性感知资源分配

物联网设备的激增伴随着网络流量的大幅增长

这可能会超出当前网络的承载能力。此外，当这些设备分析传感器信息或用户信息时，数据隐私和时延是重要的考量因素。因此，云计算等传统方法并不适用。不过，雾计算等先进计算平台可以满足这一需求。了解初始输入数据的本地化如何影响雾平台的性能，对于制定合理的负载均衡和资源分配方案至关重要 [160, 161]。因此，当多个受截止期限限制的数据密集型应用动态到达时，有必要对异构联邦雾环境进行性能评估。例如，这些应用可能需要来自物联网层或本地雾资源的数据（例如，已传输到雾层的传感器数据，或由先前应用处理的数据）。在这种情况下，研究输入数据本地化对数据放置概率各异的联邦雾系统性能的影响，可能会影响联邦资源分配的效率。因此，我们考虑探索在远程工业4.0的超额订阅情形下，数据本地化对联邦雾系统中资源分配方法的影响。

8.2.3面向工业*4.0*操作的动态容错联邦雾系统

雾计算系统靠近数据源，为终端用户提供低时延。相比之下，雾的分布式特性有助于处理工业4.0产生的大量传感器生成的数据。因此，联合雾计算资源可以支持延迟敏感型任务和数据处理

服务。然而，雾系统存在各种不确定性（例如，瞬时故障[162], 网络和停电），在满足激增的计算需求时必须考虑这些因素。尤其是在紧急情况下，这些不确定性可能导致任务无法成功完成，从而对环境甚至人类生命造成重大损害。因此，雾联邦的资源管理系统应能够感知这些不确定性，并提供高效的容错 [163, 164, 165]解决方案，以确保接收到的延迟敏感型任务按时成功完成。相应地，联邦管理系统应考虑提供一项服务，持续监控雾资源，并向所有参与雾节点发送邻近雾节点状态的信号。此外，由于雾系统的内部问题（例如，软件、硬件），已经开始的服务执行也可能受到干扰或失败。在这种情况下，每个雾系统都应具备相应的方法（例如，重新执行，或将失败的任务卸载到另一个雾节点），以确保接收到的任务成功执行。因此，容错联邦雾系统对于在紧急情况或灾害情况下满足激增的计算需求至关重要，能够增强系统鲁棒性，并推动实现安全的工业4.0。

与超级云类似， [166], 雾系统提供各种虚拟服务， [167]例如跨雾联邦的应用部署、多租户、互操作性和服务迁移。因此，要构建能够提供各种雾虚拟服务的雾联邦，就需要支持容错特性，以便高效利用雾服务。此外，在超额订阅情形下，容错服务需要解决雾联邦的可扩展性问题，确保系统鲁棒性。

在动态条件下。因此，为了满足所有这些要求，我们考虑在未来开展研究，开发一种雾系统（超级雾），为联邦雾系统提供容错雾服务，从而提高联邦的效率。

具备异构资源的雾系统以及动态雾联邦 [168]概念日益普及，因此需要开发雾友好型应用，并对应用程序栈和雾资源进行适当研究。然而，构建这类应用既耗时，又需要克服若干重大障碍。首先是支持雾网络的动态特性；其次是管理应用逻辑依赖上下文的特性；第三是应对系统的庞大规模。因此，我们必须考虑如何利用现有软件组件，将应用程序分解并部署到地理位置分散的雾联邦中；这些组件可以经过修改和复用，从而参与雾应用程序 [169]。因此，将应用层与执行层抽象分离，可以成为解决雾系统异构性挑战的首要目标。由此，我们希望研究如何为动态联邦雾系统开发雾友好型应用，使其能够识别超级雾系统的特性。

8.2.4智能工业*4.0*解决方案中的人机交互认知方面

工业4.0是一场工业技术范式转变，要求人和机器（例如，机器人、无人机）采用新的协作方式。

不断升级的先进传感器和协作机器的引入，引发了人们对安全问题的关注。在工业4.0这一高度技术化且富于创新的场景中，一系列赋能技术不断涌现，人和机器之间的资源高度互联，这些因素都对安全产生影响。一方面，先进的软件工具和机器有助于人类工作（人机协作）。另一方面，它们必须与其他智能设备进行通信并共享数据（人机交互） [170]。自“智能”技术问世以来，这些创新技术的部署环境以及一线人类工作人员的职责都发生了变化。在复杂的工业运营或灾害时期，人机交互可能颇具挑战，这与人类工人的认知方面密切相关 [171] 。当某些任务需要专门的人类能力时，人与机器之间便会形成真正的“协作”。在当今现代工业中，工作节点与“智能机器”的交互可以让工作更轻松，因为任务实现了更高程度的自动化，也更不容易出现人为错误。相反，这也会使工作场所变得更加复杂，因为不同系统之间的信息与通信流增加了。例如，借助传感器和尖端技术，我们可以收集所需信息，从而准确预测工业机械的健康状况，并实施精准处理。由于人类必须处理需要收集、分析和理解的海量数据（大数据），机器操作员的认知交互负荷也从技能水平提升至知识水平 [172]。因此，我们希望探讨各种

工业4.0人机交互问题的认知方面，并为人类工人在复杂的工业场景中提供帮助，开发智能解决方案。

8.2.5工业领域人机交互的雾计算与高级分析

随着工业革命的发展，计算技术的进步借助自动化、机器人技术、人工智能及其他现代智能解决方案，改变了工业运营。尽管如此，各种复杂的工业作业（例如，机器维护、油井钻探作业、制造机器）仍需要人为干预和交互 [173, 174] ，以确保作业的精度和准确性。因此，与机器通信的人类操作员有时需要处理机器生成的数据，以便与机器高效通信 [175, 176]。在这种情况下，人类操作员可以使用随身携带的移动设备来处理数据，或查看由附近计算机系统处理的数据。因此，雾计算可能是支持人机交互计算需求的理想候选方案 [177, 178, 179]。此外，雾计算利用多种高级分析技术（例如，机器学习、深度神经网络、强化学习）处理机器生成的数据，可以为操作人员提供有用的洞见，从而改善人机交互。

参考文献

[1] M. Singh、S. Tuli、R. J. Butcher、R. Kaur 和 S. S. Gill，“从云计算向工业4.0的动态转型：环保之选还是气候变化威胁”，载于面向环境与生态工程的物联网智能建模，2021年，第275–293页。[2] 思科，“油气行业新现实：数据管理与分析”，思科及其关联公司，白皮书，2017年。[3] M. Tortonesi、M. Govoni、A. Morelli、G. Riberto、C. Stefanelli 和 N. Suri，“驾驭物联网数据洪流：面向雾计算应用的创新型以信息为中心的服务模型”，《未来一代计算机系统》期刊，第93卷，第888–902页，2019年。[4] J. H. Park、S. Lee、S. Yun、H. Kim 和 W.-T. Kim，“采用多功能人工智能框架的可靠火灾探测系统”，《传感器》期刊，第19卷，第9期，第2025页，2019年。[5] R. Combier，“油藏模拟迁移至云端 - Rescale”，https://rescale.com/blog/reservoir-simulation-moves-to-the-cloud/，2014年3月，[在线；访问日期：2022-11-01]。[6] S. Trinks 和 C. Felden，“支持实时分析应用的边缘计算架构：智能工厂和工业4.0应用领域的最新进展”，载于*IEEE*国际大数据会议论文集（*Big Data*），2018年，第2930–2939页。[7] M. Hartmann、U. S. Hashmi 和 A. Imran，“智能医疗保健系统中的边缘计算：综述、挑战与研究方向”，《新兴电信技术汇刊》期刊，第33卷，第3期，第e3710页，2022年。[8] X. Chen、Q. Shi、L. Yang 和 J. Xu，“ThriftyEdge：面向智能物联网应用的资源高效边缘计算”，《*IEEE Network*》期刊，第32卷，第1期，第61–65页，2018年。[9] S. Rani、A. Kataria 和 M. Chauhan，“工业4.0中的雾计算：应用与挑战——研究路线图”，雾边缘计算范式的节能解决方案，第173–190页，2022年。

[10] A. Jamwal、R. Agrawal 和 M. Sharma，《深度学习与制造业可持续性：模型、工业4.0应用及其影响》，《信息管理数据洞察》期刊，第2卷第2期，页码100107，2022年。

[11] D. Batistakis、A. Xenakis、G. Papastergiou、P. Chatzimisios 和 V. C. Gerogiannis，《基于人工智能的预测即服务模型：估算工业4.0 5G应用中的机器轴承健康状态》，载于第*12*届国际信息、智能、系统与应用会议（*IISA*）论文集，2021年，第1–8页。[12] H. Webert、T. D¨oß、L. Kaupp 和 S. Simons，《工业4.0中的故障处理：定义、流程与应用》，《传感器》期刊，第22卷第6期，页码2205，2022年。[13] A. E. Eshratifar、M. S. Abrishami 和 M. Pedram，《JointDNN：面向智能移动云计算服务的高效训练与推理引擎》，《*IEEE*移动计算汇刊》期刊，2019年。[14] R. F. Hussain、A. Pakravan 和 M. A. Salehi，《云计算系统中智能工业4.0应用的性能分析》，载于第*22*届*IEEE*高性能计算与通信国际会议、*IEEE*第*18*届智慧城市国际会议及*IEEE*第*6*届数据科学与系统国际会议（*HPCC/SmartCity/DSS*）论文集，2020年，第11–18页。[15] T.-Y. Kan、Y. Chiang 和 H.-Y. Wei，《移动边缘计算系统中的任务卸载与资源分配》，载于第*27*届无线与光通信会议（*WOCC*）论文集，2018年，第1–4页。[16] J.-S. Jwo、C.-H. Lee 和 C.-S. Lin，《数据孪生驱动的智能制造信息物理工厂》，《传感器》期刊，第22卷第8期，页码2821，2022年。[17] N. Dragoni、S. Giallorenzo、A. L. Lafuente、M. Mazzara、F. Montesi、R. Mustafin 和 L. Safina，《微服务：过去、现在与未来》，《当前与未来的软件工程》，第195–216页，2017年。[18] C. Sonmez、A. Ozgovde 和 C. Ersoy，《EdgeCloudSim：用于边缘计算系统性能评估的环境》，载于第*2*届雾计算与移动边缘计算国际会议论文集，系列FMEC '17，2017年5月，第39–44页。[19] A. Srinivasan、M. LeHenaff、V. H. Kourafalou、W. C. Thacker、N. F. Tsinoremas、J. Helgers、C. B. Paris、H. Kang、M. Iskandarani、J. P. Zysman等，《用于模拟深水地平线油井喷发所泄漏石油归趋的多任务计算》，载于第*3*届网格与超级计算机多任务计算研讨会论文集，2010年，第1–7页。[20] C. Hanlon，《Usumacinta灾难》，*Proto-Type*，第1卷，2013年。

[21] J. Daley，《孟买高北平台灾难》，*Proto-Type*，第1卷，2013年。

[22] 《Wired——什么是云？》https://www.accenture.com/ acnmedia/PDF-145/Accenture-What-is-Cloud-WIRED-Brand-Lab-Video-Transcript.pdf，访问ed 2021年4月29日。[23] “石油和天然气行业的云计算 | COPAS”，2021年8月10日，[在线；访问日期：2022-05-01]。[24] R. Buyya、A. Beloglazov 和 J. Abawajy，“面向云计算的数据中心资源节能管理：愿景、架构要素与开放挑战”，《*arXiv*预印本》*arXiv:1006.0308*，2010年。[25] “什么是边缘计算？你需要了解的一切”，https://searchdatacenter.techtarget.com/definition/edge-computing，访问d 2021年7月20日。[26] S. P. Singh、A. Nayyar、R. Kumar 和 A. Sharma，“雾计算：从架构到边缘计算和大数据处理”，*Journal of Supercomputing*期刊，第75卷，第4期，第2070–2105页，2019年。[27] D. Balouek-Thomert、E. G. Renart、A. R. Zamani、A. Simonet 和 M. Parashar，“迈向计算连续体：实现数据驱动工作流的边缘到云集成”，《高性能计算应用期刊》，第33卷，第6期，第1159–1174页，2019年。[28] A. Luckow、K. Rattan 和 S. Jha，“Pilot-edge：沿边缘到云连续体的分布式资源管理”，载于*IEEE*国际并行与分布式处理研讨会研讨班（*IPDPSW*）论文集，2021年，第874–878页。[29] K. Rao、G. Coviello、W. Hsiung 和 S. Chakradhar，“Eco：5G应用的边缘云优化”，载于第*21*届*IEEE*集群、云和互联网计算国际研讨会（*CCGrid*）论文集，2021年5月，第649–659页。[30] M. C. Lucas-Esta˜n 和 J. Gozalvez，“自组织工业物联网网络的负载均衡”，*IEEE* 工业信息学汇刊，第15卷，第9期，第5052–5063页，2019年。[31] H. Pei Breivold，“迈向未来工厂：在云计算和物联网背景下迁移工业遗留自动化系统”，《企业信息系统》期刊，第14卷，第4期，第542–562页，2020年。[32] M. S. Mahmoud，“基于云的工业自动化架构”，载于第三届信息与通信技术国际大会*gy*, 2019年，第51–62页。

[33] T. Lewandowski、D. Henze、M. Sauer、J. Nickles 和 B. Bruegge，“一种支持自组织协作式物联网资源网络的软件架构”，载于*2020*年第五届雾计算与移动边缘计算国际会议（*FMEC*），2020年，第70–77页。[34] L. D. Xu 和 L. Duan，“工业4.0信息物理系统的大数据：综述”，*Enterprise Information Systems*，第13卷，第2期，第148–169页，2019年。[35] Y. Lu，“工业4.0：技术、应用及开放研究问题综述”，*Journal of Industrial Information Integration*，第6卷，第1–10页，2017年。[36] L. Zhou、D. Wu、J. Chen 和 Z. Dong，“当计算拥抱智能：面向工业物联网的内容感知数据处理”，*Journal of IEEE Internet ofThings Journal*，第5卷，第3期，第1657–1666页，2018年。[37] J. Nilsson 和 F. Sandin，“工业4.0中的语义互操作性：最新进展综述与展望”，载于第*16*届*IEEE*工业信息学国际会议论文集（*INDIN*），2018年，第127–132页。[38] R. Hussain、M. Amini、A. Kovalenko、Y. Feng 和 O. Semiari，“面向远程智能油田灾害管理的联邦边缘计算”，载于第*21*届*IEEE*高性能计算与通信国际会议论文集，HPCC’19系列，2019年8月，第929–936页。[39] A. Parrott 和 L. Warshaw，“工业4.0与数字孪生”，载于 *DeloitteUniversity Press*，2017年，第1–17页。[40] F. Tao、J. Cheng、Q. Qi、M. Zhang、H. Zhang 和 F. Sui，“大数据驱动的数字孪生产品设计、制造与服务”，*Journal ofAdvanced Manufacturing Technology*，第94卷，第9期，第3563–3576页，2018年。[41] “以智能边缘和物联网变革石油和天然气行业”，https://www.ciosummits.com/SB Oil Gas IoT-1.pdf，访问日期：2021年4月25日。[42] B. Eakins 等，“提高NOAA NCEI深层档案库中科学海洋钻探数据的可访问性”，2015年。[43] J. Xu、B. Palanisamy、H. Ludwig 和 Q. Wang，“Zenith：面向边缘计算的效用感知资源分配”，载于 *IEEE*边缘计算国际会议论文集（*EDGE*），2017年，第47–54页。[44] B. Lorenzo、J. Garcia-Rois、X. Li、J. Gonzalez-Castano 和 Y. Fang，“面向物联网的稳健动态边缘网络架构”，*Journalof IEEE Network*，第32卷，第8–15页，2017年1月。

[45] C. Chang、A. Hadachi 和 S. Srirama，“异构云-雾-边缘计算环境中面向物联网的自适应边缘进程迁移”，*arXiv* 预印本 *arXiv:1811.10939*，2018年11月。[46] M. A. Salehi、J. Smith、A. A. Maciejewski、H. J. Siegel、E. K. Chong、J. Apodaca、L. D. Brice˜no、T. Renner、V. Shestak、J. Ladd 等，“异构计算系统中独立任务的基于随机性的鲁棒动态资源分配”，并行与分布式计算杂志 *(JPDC)*，第97卷，第96–111页，2016年11月。[47] W. Z. Khan、M. Y. Aalsalem、M. K. Khan、M. S. Hossain 和 M. Atiquzzaman，“面向石油和天然气行业的可靠物联网架构”，载于第*19*届先进通信技术国际会议论文集，2017年2月，第705–710页。[48] S. Wunderlich、J. A. Cabrera、F. H. Fitzek 和 M. Reisslein，“采用并行矩阵分块运算DAG调度的异构多核物联网节点中的网络编码”，*IEEE*物联网杂志，第4卷，第4期，第917–933页，2017年5月。[49] Y. Wang、X. Lin 和 M. Pedram，“移动云计算系统中的嵌套式两阶段博弈优化框架”，载于第*7*届*IEEE*服务导向系统工程国际研讨会论文集 *(SOSE)*，2013年3月，第494–502页。[50] Z. Hu、Y. Wei、X. Wang 和 M. Song，“蜂窝网络中绿色中继站辅助小区缩放方案”，载于第*12*届自然计算、模糊系统与知识发现国际会议论文集，2016年8月，第2030–2035页。[51] M. M. K. Tareq、O. Semiari、M. A. Salehi 和 W. Saad，“采用边缘计算的超可靠、低时延车联网无线通信”，*arXiv*预印本 *arXiv:1808.06015*，2018年。[52] X. Xu、H. Cao、Q. Geng、X. Liu、F. Dai 和 C. Wang，“边缘计算环境中不确定条件下工作流调度的动态资源配置”，并发与计算：实践与经验，第34卷，第14期，文章编号e5674，2022年。[53] L. Li、Q. Guan、L. Jin 和 M. Guo，“雾排队系统中具有不确定持续时间的异构实时任务的资源分配与任务卸载”，*IEEE Access*，第7卷，第9912–9925页，2019年。[54] S. Pejoski、Z. Hadzi-Velkov 和 T. Shuminoski，“射频能量采集IRS辅助无线网络中基于Lyapunov漂移加惩罚的资源分配”，*Radioengineering*，第31卷，第3期，第383页，2022年。

[55] G. Somashekar 和 A. Gandhi，“面向微服务的最优配置”，载于第*1*届机器学习与系统研讨会论文集，2021年，第7–14页。[56] H. Vural 和 M. Koyuncu，“领域驱动设计是否有助于找到微服务的最优模块化程度？”*IEEE Access*，第9卷，第32 721–32 733页，2021年。[57] G. Kecskemeti、A. C. Marosi 和 A. Kertesz，“将单体服务分解为微服务的 Entice 方法”，载于高性能计算与仿真国际会议论文集 *(HPCS)*，2016年，第591–596页。[58] T. Alam，“基于云的物联网应用及其在智慧城市中的作用”，*Smart Cities*，第4卷，第3期，第1196–1219页，2021年。[59] A. L. Shastry、D. S. Nair、B. Prathima、C. P. Ramya 和 P. Hallymysore，“将非云原生应用迁移到云端的方法”，载于第*12*届*IEEE*年度计算与通信研讨会暨会议论文集 *(CCWC)*，2022年，第0632–0638页。[60] K. Rao、G. Coviello、W.-P. Hsiung 和 S. Chakradhar，“Eco：5G应用的边缘云优化”，载于第*21*届*IEEE/ACM*集群、云与互联网计算国际研讨会论文集 *(CCGrid)*，2021年，第649–659页。[61] F. Faticanti、F. De Pellegrini、D. Siracusa、D. Santoro 和 S. Cretti，“雾计算中吞吐量感知的应用分区与部署”，*IEEE* 网络与服务管理汇刊，第17卷，第4期，第2436–2450页，2020年。[62] S. Gupta、S. Gupta 和 D. Goyal，“物联网中的无线传感器网络与性能优化”，计算机科学与通信最新进展（原名：计算机科学最新专利），第15卷，第1期，第14–22页，2022年。[63] A. A. Sadri、A. M. Rahmani、M. Saberikamarposhti 和 M. Hosseinzadeh，“雾计算与物联网中的数据缩减：系统性文献综述”，物联网，第100629页，2022年。[64] S. Yi、C. Li 和 Q. Li，“雾计算综述：概念、应用及

d

“问题”，载于移动大数据研讨会论文集，2015年，第37–42页。[65] K. Ha、Z. Chen、W. Hu、W.Richter、P. Pillai 和 M. Satyanarayanan，“迈向可穿戴认知辅助”，载于第*12*届移动系统、应用与服务年度国际会议，2014年，第68–81页。

[66] B. McMahan、E. Moore、D. Ramage、S. Hampson 和 B. A. y Arcas，“基于去中心化数据的深度网络通信高效学习”，载于人工智能与统计学，2017年，第1273–1282页。[67] K. Bonawitz、H. Eichner、W. Grieskamp、D. Huba、A. Ingerman、V. Ivanov、C. Kiddon、J. Koneˇcn`y、S. Mazzocchi、H. B. McMahan 等，“迈向大规模联邦学习：系统设计”，*arXiv*预印本 *arXiv:1902.01046*，2019年。[68] Q. Xia、W. Ye、Z. Tao、J. Wu 和 Q. Li，“边缘计算联邦学习综述：研究问题与解决方案”，《高可信计算期刊》，第100008页，2021年。[69] Y. Lu、X. Xu 和 L. Wang，“智能制造流程与系统自动化——标准及预期场景的批判性综述”，《制造系统期刊》，第56卷，第312–325页，2020年。[70] N. Sakib、N. U. I. Hossain、F. Nur、S. Talluri、R. Jaradat 和 J. M. Lawrence

，

“利用贝叶斯信念网络评估石油和天然气供应链中的概率性灾害”，《生产经济学期刊》，第235卷，108107页，2021年。[71] H. Lu、L. Guo、M. Azimi和K. Huang，“石油和天然气4.0时代：系统综述与展望”，《工业计算机期刊》，第111卷，第68–90页，2019年。[72] A. Hahn，“工业控制系统中的运营技术与信息技术”，载于《*SCADA*及其他工业控制系统的网络安全》，2016年，第51–68页。[73] T. Nguyen、R. G. Gosine和P. Warrian，“面向石油和天然气行业4.0的大数据分析系统综述”，《*IEEE Access*》，第8卷，第61 183–61 201页，2020年。[74] A. J. Dunnings和T. P. Breckon，“针对非时序实时火灾检测的实验定义卷积神经网络架构变体”，载于第*25*届*IEEE*国际图像处理会议（*ICIP*）论文集。IEEE，2018年，第1558–1562页。[75] M. Z. Alom、T. M. Taha、C. Yakopcic、S. Westberg、P. Sidike、M. S. Nasrin、B. C. Van Esesn、A. A. S. Awwal和V. K. Asari，“从AlexNet开始的历史：深度学习方法综合综述”，*arXiv*预印本 *arXiv:1803.01164*，2018年。[76] D. Anguita、A. Ghio、L. Oneto、X. Parra和J. L. Reyes-Ortiz，“用于基于智能手机的人体活动识别的公共领域数据集。”载于*Esann*，2013年。

[77] M. Krestenitis、G. Orfanidis、K. Ioannidis、K. Avgerinakis、S. Vrochidis和I. Kompatsiaris，“利用深度神经网络从卫星图像中识别溢油”，《遥感学报》，第11卷，第15期，1762页，2019年。[78] J. Long、E. Shelhamer和T. Darrell，“用于语义分割的全卷积网络”，载于*IEEE*计算机视觉与模式识别会议论文集，2015年，第3431–3440页。[79] Z. Huang、C. O. Dumitru、Z. Pan、B. Lei和M. Datcu，“利用深度迁移学习对大规模高分辨率SAR图像进行分类”，《地球科学与遥感快报》，2020年。[80] A. Mustafa、M. Alfarraj和G. AlRegib，“利用时间卷积网络从地震数据估算声阻抗”，*arXiv*预印本 *arXiv:1906.02684*，2019年。[81] R. Versteeg，“Marmousi经验：基于合成复杂数据集确定速度模型”，《*The Leading Edge*》，第13卷，第9期，第927–936页，1994年9月。[82] J. Varia、S. Mathew 等，“亚马逊云服务概述”，亚马逊云科技，第1–22页，2014年。[83] V. Veillon、C. Denninnart和M. A. Salehi，“F-FDN：用于低延迟视频流的雾计算系统联邦”，载于第*3*届*IEEE*雾计算与边缘计算国际会议论文集，2019年，第1–9页。[84] K. Keahey、P. Riteau、D. Stanzione、T. Cockerill、J. Mambretti、P. Rad和P. Ruth，“Chameleon：面向计算机科学研究的可扩展生产级测试平台”，载于《当代高性能计算：从拍字节级迈向艾字节级》，2019年5月，第3卷，第5章，第123–148页。[85] S. S. Ogden和T. Guo，“移动应用程序深度神经网络推理性能的特征分析”，*arXiv*预印本 *arXiv:1909.04783*，2019年。[86] H. Moradi、W. Wang和D. Zhu，“多租户云中应用程序的自适应性能建模与预测”，载于第*21*届高性能计算与通信国际会议（*HPCC*）论文集，2019年，第638–645页。[87] X. Li、M. A. Salehi、Y. Joshi、M. K. Darwich、B. Landreneau和M. Bayoumi，“利用异构云服务进行视频转码的性能分析与建模”，《*IEEE*并行与分布式系统汇刊（*TPDS*）》，第30卷，第4期，第910–922页，2019年。[88] Z. Hanusz、J. Tarasinska和W. Zielinski，“均值已知时的夏皮罗-威尔克检验”，《*REVSTAT-Statistical*》，第14卷，第1期，2016年。

[89] I. M. Chakravarti、R. G. Laha 和 J. Roy，《应用统计方法手册》，威利概率与数理统计丛书（美国）英文版，1967年。[90] D. J. Lilja，《计算机性能测量：从业者指南》。剑桥大学出版社，2005年。[91] S. Patil 和 D. J. Lilja，“使用重抽样技术计算基于速率的性能指标调和平均数的置信区间”，*IEEE*《计算机体系结构快报》，第9卷，第1期，2010年。[92] B. Charyyev、A. Alhussen、H. Sapkota、E. Pouyoul、M. H. Gunes 和 E. Arslan，“保障数据传输免受静默数据损坏”，载于第*19*届*IEEE/ACM*集群、云和网格计算国际研讨会（*CCGRID*）论文集，2019年，第262–271页。[93] S. Mukherjee，“异构蜂窝网络中的下行链路信干噪比分布”，《通信领域精选》期刊，第30卷，第3期，第575–585页，2012年4月。[94] J. Crompton，“数字油田炒作曲线：石油和天然气行业数字油田计划的现状评估”，载于*SPE*数字能源会议暨展览会论文集。OnePetro，2015年。[95] X. Li、M. A. Salehi、M. Bayoumi 和 R. Buyya，“CVSS：利用云服务实现成本高效且服务质量感知的视频流”，载于第*16*届*IEEE/ACM*集群、云和网格计算国际研讨会（*CCGrid*）论文集，CCGRID ’16系列，2016年5月，第106–115页。[96] C. Denninnart、J. Gentry 和 M. A. Salehi，“通过概率任务剪枝提升异构无服务器计算系统的鲁棒性”，载于第*33*届*IEEE*国际并行与分布式处理研讨会专题研讨会论文集，2019年5月。[97] C. O. Diaz、M. Guzek、J. E. Pecero、G. Danoy、P. Bouvry 和 S. U. Khan，“异构计算系统中的能量感知快速调度启发式算法”，载于高性能计算与仿真国际会议论文集，2011年7月，第478–484页。[98] M. Skedsmo、R. Ayasse、N. Soleng、M. Indregard 等，“利用卫星图像进行溢油检测与响应：技术洞察及监管背景”，载于石油工程师学会（*SPE*）健康、安全、安保、环境与社会责任国际会议暨展览会论文集，2016年4月。[99] R. F. Hussain、M. A. Salehi、A. Kovalenko、S. Salehi 和 O. Semiari，“鲁棒资源分配使

g edg电子计算 g 面向智能油田，”见论文集 *gs*

第*24*届并行与分布式处理技术及应用国际会议，2018年8月。[100] H. A. Bramantyo、B. S. Utomo 和 E. M. Khusna，《油气炼厂中的物联网数据处理》，*Jaringan Telekomunikasi*期刊，第12卷，第1期，第48–54页，2022年。[101] A. M. Titu 和 A. Stanciu，《将运营技术与信息技术相融合》，载于第*12*届电子、计算机与人工智能国际会议（*ECAI*）论文集，2020年，第1–6页。[102] J. Daily 和 J. Peterson，《预测性维护：大数据分析如何改善维护工作》，载于《商用航空航天供应链整合挑战》论文集，2017年，第267–278页。[103] 思科，《油气行业新现实：数据管理与分析》，https://www.cisco.com/c/dam/en us/solutions/industries/energy/docs/OilGasDigitalTransfo

rm

sep 17 2017, [在线；访问日期：2022-08-01].[104] E. Kristiani、C.-T. Yang、C.-Y. Huang、Y.-T. Wang 和 P.-C. Ko，“使用 OpenStack 和 Kubernetes 实现用于空气质量监测应用程序的云边计算架构”，《移动网络与应用》期刊，第26卷，第3期，第1070–1092页，2021年。[105] C. D. Heising 和 W. S. Grenzebach，“Ocean Ranger钻井平台灾难：风险分析”，《风险分析》期刊，第9卷，第1期，第55–62页，1989年。[106] X. Deng、J. Yin、P. Guan、N. N. Xiong、L. Zhang 和 S. Mumtaz，“通过边缘计算为多用户工业物联网实现智能、时延感知的部分计算任务卸载”，《*IEEE*物联网期刊》，2021年。[107] L. Cricelli、R. Mauriello 和 S. Strazzullo，“ 创新服务的市场分析：卫星通信行业的案例研究 ”，载于工业工程与工业管理第三届国际会议论文集，2022年，第92–100页。[108] B. Aberle，“面向智能制造设备的异常检测算法之基于微服务的机器学习生产部署”，硕士论文，阿尔托大学理学院，2022年。[109] A. J. Calder´on Godoy 和 I. Gonz´alez P´erez，“整合传感器与执行器网络及 SCADA 系统，推动传统柔性制造系统向工业4.0概念转型”，《传感器与执行器网络》期刊，第7卷，第2期，第23页，2018年。

[110] N. Verba、K.-M. Chao、J. Lewandowski、N. Shah、A. James 和 F. Tian，“面向应用分析与部署的工业4.0雾计算环境建模”，《未来一代计算机系统》期刊，第91卷，第48–60页，2019年。[111] A. Lakhan、Q.-U.-A. Mastoi、M. Elhoseny、M. S. Memon 和 M. A. Mohammed，“利用物联网辅助的移动雾云，为医院和医疗企业实现基于深度神经网络的应用划分与调度”，《企业信息系统》期刊，第16卷，第7期，第1883122页，2022年。[112] A. Lochbihler，“可数网络最大流最小割定理及其概率论应用的机械化证明”，《自动推理》期刊，第1–26页，2022年。[113] Z. M. Cınar、A. Abdussalam Nuhu、Q. Zeeshan、O. Korhan、M. Asmael 和 B. Safaei，“面向工业4.0可持续智能制造的预测性维护机器学习”，《可持续发展》期刊，第12卷，第19期，第8211页，2020年。[114] M. Duan、D. Liu、X. Chen、R. Liu、Y. Tan 和 L. Liang，“移动系统中具有全局不平衡数据的自平衡联邦学习”，《*IEEE*并行与分布式系统汇刊》，第32卷，第1期，第59–71页，2020年。[115] M. Duan、D. Liu、X. Chen、Y. Tan、J. Ren、L. Qiao 和 L. Liang，“Astraea：用于提高移动深度学习应用程序分类准确率的自平衡联邦学习”，载于第*37*届*IEEE*国际计算机设计会议（*ICCD*）论文集，2019年，第246–254页。[116] N. Mhaisen、A. Awad、A. Mohamed、A. Erbad 和 M. Guizani，“非独立同分布数据上的分层联邦学习分析与最优边缘分配”，*arXiv*预印本 *arXiv:2012.05622*，2020年。[117] O. Ronneberger、P. Fischer 和 T. Brox，“U-Net：用于生物医学图像分割的卷积网络”，载于医学图像计算与计算机辅助干预国际会议论文集，2015年，第234–241页。[118] N. Abraham 和 N. M. Khan，“采用改进型注意力U-Net进行病灶分割的新型焦点Tversky损失函数”，载于第*16*届*IEEE*生物医学成像国际研讨会（*ISBI 2019*）论文集，2019年，第683–687页。[119] S. S. M. Salehi、D. Erdogmus 和 A. Gholipour，“使用三维全卷积深度网络进行图像分割的Tversky损失函数”，载于

医学影像中的机器学习国际研讨会。施普林格，2017年，第379–387页。[120] R. R. Outeiral、P. Bos、H. J. van der Hulst、A. Al-Mamgani、B. Jasperse、R. Sim˜oes 和 U. A. van der Heide，《磁共振成像中口咽部原发肿瘤分割类别不平衡问题的应对策略》，发表于《放射肿瘤学物理与影像学杂志》，第23卷，第144–149页，2022年。[121] S. Jadon，《语义分割损失函数综述》，载于*IEEE*生物信息学与计算生物学计算智能会议（*CIBCB*）论文集，2020年，第1–7页。[122] P. C. Sau，《使用注意力模块和 Tversky损失函数进行视网膜血管分割》，载于国际通信与人工智能会议论文集，2022年，第503–513页。

[123] A. Ziller, A. Trask, A. Lopardo, B. Szymkow, B. Wagner, E. Bluemke, J.-M. Nounahon, J. Passerat-Palmbach, K. Prakash, N. Rose等，“Pysyft：一款便于开展联邦学习的库”，载于《联邦学习系统》，2021年，第111–139页。

[124] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard 等，“Tensorflow：一种用于大规模机器学习的系统”，载于第*12*届 {*USENIX*} 操作系统设计与实现研讨会（{*OSDI*} *16*），2016年，第265–283页。[125] E. Bisong，“Google Colaboratory”，载于《在*Google*云平台上构建机器学习和深度学习模型》，2019年，第59–64页。[126] I. Kilovaty，“保障管道网络安全”，《休斯敦法律评论》，第60卷，2023年。[127] “网络攻击导致欧洲主要石油终端陷入混乱 | 标普全球商品洞察”，2月 3 2022, [在线；访问日期：2022-04-24]。[128] “近年来能源基础设施遭受的几起最大规模网络攻击——Prospero Events Group”，3月 24 2022, [在线；访问日期：2022-04-24]。[129] “石油和天然气行业的信息技术与运营技术融合：主要策略与优势”， [在线；访问日期：2022-04-24]。[130] Q. Zhang, Z. Qian, S. Wang, L. Yuan, 和 B. Gong，“生产力下降还是提升？新技术采用对油田市场的影响”，《能源经济学杂志》，第108卷，文章编号105930，2022年。[131] “了解信息技术与运营技术融合在石油和天然气行业中的作用”， [在线；访问日期：2022-04-26

].

[132] E. D. Knapp 和 J. Langill，《工业网络安全：保障智能电网、*SCADA*及其他工业控制系统的关键基础设施网络》。Syngress，2014年。[133] N. Jan, A. Nasir, M. S. Alhilal, S. U. Khan, D. Pamucar, 和 A. Alothaim，“利用复直觉模糊关系的新型结构，研究石油和天然气行业的网络安全与网络犯罪”，《*Entropy*》期刊，第23卷，第9期，文章编号1112，2021年。[134] S. Mandal 和 A. Chirputkar，“石油和天然气行业的数字化转型与网络中断”，*PalArch*《埃及考古学／埃及学期刊》，第17卷，第6期，第4402–4416页，2020年。[135] Hacquebord 和 C. Pernet，“深入钻探：审视针对石油和天然气行业的网络攻击”，2019, [在线]可访问：https://www.trendmicro.com/vinfo/it/security/news/internet-of-things/drilling-deep-a-look-at-cyberattacks-on-the-oil-and-gas-industry。[136] B. Conde Gallego 和 J. Drexl，“物联网连接标准：当前标准必要专利监管框架的适应性如何？”《*IIC*——知识产权与竞争法国际评论》，第50卷，第1期，第135–156页，2019年。[137] M. T. Hossain, S. Badsha, 和 H. Shen，“Porch：一种面向智能电网和工业4.0中区块链赋能的未来SCADA系统的新型共识机制”，载于*IEEE*国际物联网、电子与机电一体化会议（*IEMTRONICS*）论文集，2020年，第1–7页。[138] D. Ongaro 和 J. Ousterhout，“探索易于理解的共识算法”，载于*USENIX*年度技术会议论文集（*Usenix ATC 14*），2014年，第305–319页。[139] M. B. Mollah, J. Zhao, D. Niyato, K.-Y. Lam, X. Zhang, A. M. Ghias, L. H. Koh, 和 L. Yang，“面向未来智能电网的区块链：全面综述”，《*IEEE*物联网期刊》，第8卷，第1期，第18–43页，2020年。[140] R. Zhang, R. Xue, 和 L. Liu，“区块链上的安全与隐私”，《*ACM*计算调查》（*CSUR*），第52卷，第3期，第1–34页，2019年。[141] R. Di Pietro, X. Salleras, M. Signorini, 和 E. Waisbard，“基于区块链的物联网信任系统”，载于第*23*届*ACM*访问控制模型与技术研讨会论文集，2018年，第77–83页。[142] S. Abdul-Jabbar，“理事会专栏：区块链技术如何颠覆石油和天然气行业”，9月 27 2021, [在线；访问日期：2022-04-29]。

[143] J. Grundy，《面向下一代基于云和边缘的智能生活应用的以人为中心的软件工程》，载于第*20*届*IEEE*集群、云和互联网计算国际研讨会论文集（*CCGRID*），2020年，第1–10页。[144] S. Lohr，《如果你是白人男性，人脸识别就很准确》，载于数据与分析伦理。Auerbach Publications，2018年，第143–147页。[145] M. Madden、A. Lenhart、S. Cortesi、U. Gasser、M. Duggan、A. Smith 和 M. Beaton，《青少年、社交媒体与隐私》，皮尤研究中心期刊，第21卷，第1055期，第2–86页，2013年。[146] T. Bolukbasi、K.-W. Chang、J. Y. Zou、V. Saligrama 和 A. T. Kalai，《男人之于计算机程序员，是否如女人之于家庭主妇? 词嵌入去偏》，神经信息处理系统进展期刊，第29卷，2016年。[147] D. Wesley 和 L. A. Dau，《恩桥输油管道灾难中的自满与自动化偏见》，设计工效学期刊，第25卷，第1期，第17–22页，2017年。[148]《自动化偏见》，https://databricks.com/glossary/automation-bias。[149] K. Rick、I. Mart´en 和 U. Von Lonski，《未开发储量：促进石油和天然气行业的性别平衡》，世界石油理事会、波士顿咨询集团，2017年。[150] A. Tversky 和 D. Kahneman，《不确定性下的判断：启发式与偏差：判断中的偏差揭示了不确定情境下的一些思维启发式》，《科学》期刊，第185卷，第4157期，第1124–1131页，1974年。[151] M. Prause 和 J. Weigand，《钻井平台：用于训练去偏技术的领导力实践游戏》，载于*2017*年冬季仿真会议（*WSC*），2017年，第4312–4323页。[152] R. F. Hussain、A. Mokhtari、A. Ghalambor 和 M. A. Salehi，面向石油和天然气行业智能运营的物联网：从上游到下游，第1版。爱思唯尔，2022年。[153] M. L. Rathod、A. Shivaputra、H. Umadevi、K. Nagamani 和 S. Periyasamy，《面向智慧农场农业科技的云计算与网络》，《纳米材料学报》，第2022卷，2022年。[154] A. Elhadad、F. Alanazi、A. I. Taloba 和 A. Abozeid，《用于管理实时通知的医疗监测系统雾计算服务》，《医疗保健工程期刊》，第2022卷，2022年。

[155] Z. Cheng、J. Zhang、T. Song 和 J. Hu，《云辅助停车车辆边缘计算中计算资源分配的基于契约的方案》，*Physical Communication*期刊，第101916页，2022年。[156] I. Sarkar 和 S. Kumar，《异构雾计算网络中的基于深度学习的节能计算卸载策略》，*Journal of Supercomputing*期刊，第1–18页，2022年。[157] H. Ye、G. Y. Li 和 B.-H. F. Juang，《面向车对车通信的基于深度强化学习的资源分配》，*IEEE*车辆技术汇刊，第68卷，第4期，第3163–3173页，2019年。[158] H. Zhang、H. Zhou 和 M. Erol-Kantarci，《开放无线接入网切片中的资源分配联邦深度强化学习》，*arXiv*预印本 *arXiv:2208.01736*，2022年。[159] A. Lakhan、M. A. Mohammed、O. I. Obaid、C. Chakraborty、K. H. Abdulkareem 和 S. Kadry，《软件定义网络赋能的雾计算范式中感知资源分配的高效深度强化学习》，*Automated Software Engineering*期刊，第29卷，第1期，第1–25页，2022年。[160] G. L. Stavrinides 和 H. D. Karatza，《异构雾环境中输入数据局部性各异的实时工作流编排》，载于第*5*届雾计算与移动边缘计算国际会议（*FMEC*）论文集，2020年，第202–209页。[161] ——，《分布式环境中线性流水线应用的数据感知资源分配》，载于第*13*届信息与通信系统国际会议（*ICICS*）论文集，2022年，第121–126页。[162] M. Al Ahmad、S. Alnazer 和 S. S. Patra，《灾难事件下雾节点的瞬态分析》，载于第*3*届新兴技术国际会议（*INCET*）论文集，2022年，第1–5页。[163] W. Saeed、Z. Ahmad、A. I. Jehangiri、N. Mohamed、A. I. Umar 和 J. Ahmad，《雾计算中医疗物联网的容错数据管理方案》，*KSII*互联网与信息系统汇刊（*TIIS*），第15卷，第1期，第35–57页，2021年。[164] K. Wang、Y. Shao、L. Xie、J. Wu 和 S. Guo，《基于雾计算的医疗物联网自适应容错数据处理》，*IEEE*网络科学与工程汇刊，第7卷，第1期，第263–273页，2018年。[165] P. Zhang、Y. Chen、M. Zhou、G. Xu、W. Huang、Y. Al-Turki 和 A. Abusorrah，《用于雾计算系统性能优化的容错模型》，*IEEE*物联网期刊，第9卷，第3期

，

第1725–1736页，2021年。

[166] Q. Jia、Z. Shen、W. Song、R. Van Renesse 和 H. Weatherspoon，《超级云：机遇与挑战》，*ACM SIGOPS* 操作系统评论，第49卷第1期，第137–141页，2015年。[167] J. Li、J. Jin、D. Yuan 和 H. Zhang，《虚拟雾：一种面向物联网的虚拟化雾计算框架》，*IEEE*物联网期刊，第5卷第1期，第121–131页，2017年。[168] A. Hammoud、M. Kantardjian、A. Najjar、A. Mourad、H. Otrok、Z. Dziong 和 N. Guizani，《面向车联网的动态雾联邦方案》，*IEEE* 网络与服务管理汇刊，2022年。[169] N. K. Giang、R. Lea 和 V. C. Leung，《开发大规模动态雾计算应用程序：案例研究》，《软件：实践与经验》，第50卷第5期，第519–532页，2020年。[170] F. Ansari、S. Erol 和 W. Sihn，《重新思考工业4.0中的人机学习：范式转变如何看待人类学习的作用？》《制造业 *Procedia*》，第23卷，第117–122页，2018年。[171] A. Chac´on、C. Angulo 和 P. Ponsa，《为工业4.0操作人员开发认知顾问代理》，《人工智能在工业中的应用新趋势》，第4卷，第127页，2020年。[172] M. Madonna、L. Monica、S. Anastasi 和 M. Di Nardo，《与工业4.0技术集成的人机交互中的认知需求演变》，*Wit* 建成环境汇刊，第189卷，第13–19页，2019年。[173] D. Gorecky、M. Schmitt、M. Loskyll 和 D. Z¨uhlke，《工业4.0时代的人机交互》，载于第*12*届 *IEEE* 工业信息学国际会议论文集（*INDIN*），2014年，第289–294页。[174] N. Kumar 和 S. C. Lee，《智能工厂中的人机界面：系统性文献综述》，《技术预测与社会变迁》，第174卷，第121284页，2022年。[175] A. Nawaz、A. S. Arora、W. Ali、N. Saxena、M. S. Khan、C. M. Yun 和 M. Lee，《智能人机界面：为中试规模污水处理厂厌氧氨氧化序批式反应器系统提供敏捷操作与决策支持》，*IEEE* 工业信息学汇刊，第18卷第9期，第6224–6232页，2022年。[176] H. Shang、C. Sivaparthipan 等，《运用人机交互开展高等教育系统的交互式教学》，*Computers and Electrical Engineering*期刊，第100卷，第107811页，2022年。

[177] Z. Wang、X. Lou、Z. Yu、B. Guo 和 X. Zhou，“基于无线感知和雾计算实现非侵入式实时人机交互”，*Personal and Ubiquitous Computing*期刊，第 23 卷，第 1 期，第 29–41 页，2019年。[178] W. Na、Y. Feng、B. Muthu 和 A. Shanthini，“面向农村儿童的人机交互辅助智能教育系统”，*Computers and Electrical Engineering*期刊，第 99 卷，文章编号 107812，2022年。[179] M. M. Ahmed、G. Yang 和 S. Gaweesh，“对网联车辆人机界面在恶劣天气条件下驾驶时的驾驶员感知进行评估：来自怀俄明州的初步发现”，*Frontier*期刊

s

心理学，第 11 卷，第 1889 页，2020年。

个人简介

Razin Farhan Hussain 于2011年秋季在孟加拉国军事科学与技术学院获得计算机科学与工程理学学士学位。他于2012年2月立即开启职业生涯，在名为“ERA Infotech Ltd”的一家软件公司担任程序员。在第一份工作任职九个月后，他加入了顶尖跨国软件公司之一“三星研发院”，担任软件工程师。Razin Farhan Hussain 在三星工作了约4年5个月，随后决定通过接受高等教育来充实自己的学术知识。因此，Razin Farhan Hussain 于2017年秋季在路易斯安那大学拉斐特分校开启了计算机科学博士之旅。在攻读博士学位期间，Razin Farhan Hussain 于2019年春季获得了计算机科学理学硕士学位。他的研究兴趣包括：云计算、雾联邦中的资源分配、任务调度、机器学习、面向工业4.0的基于深度神经网络的应用，以及联邦学习。目前，Razin Farhan Hussain 在名为“TryCycle Data Systems”的一家软件公司担任软件工程师，为加拿大和美国的医疗保健行业提供软件解决方案。