

面向嵌入式容器应用的负载预测方法研究

常祎雯¹, 李伟刚¹, 武君胜¹, 张生华¹, 李毅²

(1.西北工业大学 软件学院, 陕西 西安 710072; 2.中国航空工业集团公司第一飞机设计研究院, 陕西 西安 710089)

摘要:当前随着虚拟容器架构在嵌入式计算环境中的广泛应用,调度资源优化得以实现对负载动态变化的自适应,但其效果高度依赖容器负载预测的准确性,目前相关研究面临数据集匮乏和已有预测方法不适应嵌入式应用特征等问题。面向航空机载嵌入式应用场景,构建了符合嵌入式容器化环境的应用数据集;针对负载预测精度和计算效率较低的问题,提出结合了 CEEMDAN 算法和 Informer 模型的轻量级负载预测模型。CEEMDAN 算法通过分解时间序列数据,提升了建模的精度,而 Informer 模型利用稀疏自注意力机制,有效降低了计算复杂度和内存消耗。实验结果表明,所提出的模型与主流时序预测方法对比,各项误差指标平均下降约 10%,适合嵌入式应用场景。

关键词:嵌入式虚拟容器;容器云;负载预测;航空机载软件

中图分类号:TP3 **文献标志码:**A **文章编号:**1000-2758(2026)01-0125-09

目前,国内学术界和工业界正在积极推动云原生理念与嵌入式系统的深度融合^[1],通过技术创新和实践探索,丰富嵌入式系统的技术体系^[2]。中航工业及下属研究机构对航空机载虚拟容器技术的探索也取得了积极进展。例如,中航工业计算所推出的容器版天脉3操作系统^[3]支持应用软件的服务化架构和分布式部署,为航空机载应用提供了灵活与高效的管理能力。

在嵌入式环境中,基于容器的应用部署方式因其轻量化、可移植性和易于管理的特性,逐渐成为主流。容器应用^[4]是指运行在容器中的各类嵌入式服务或任务,具有生命周期短、负载动态变化明显的特点。容器部署则是需要通过容器技术(如 Docker)^[5]将应用及其依赖打包并在嵌入式系统中运行。由于嵌入式设备的计算资源受限,容器应用的资源需求(如 CPU、内存等)随着应用行为的变化而波动。因此,如何高效动态分配资源以适应不同需求,并提高资源利用率,成为关键问题。动态资源配置^[6]是应对这一挑战的有效手段,而实现动态分配依赖于精准的资源负载预测^[3]。通过预测未来负

载趋势,可提前调整资源分配,确保负载和资源的匹配。容器资源负载预测本质上属于时序预测^[7],但由于嵌入式环境中负载数据复杂性以及云计算领域内负载预测研究起步较晚,缺乏专用负载数据集,且嵌入式应用通常具备实时性、高安全性和资源受限等特征,已有的预测算法难以满足嵌入式环境下动态资源负载预测的特定需求。现有的研究多聚焦于迁移和改进通用时序预测算法,以及机器学习模型融合,旨在提高预测精度。

传统时间序列预测模型方面,如 Roy 等^[8]提出的自回归平均移动模型(autoregressive moving average, ARMA),虽适用于平稳负载数据,但实际中负载受多种因素影响,往往呈现非平稳性。为此,Calheiros 等^[9]提出了差分整合移动平均自回归模型(autoregressive integrated moving average, ARIMA)以适应非平稳负载数据。混合模型方面, Sudhakar 等^[10]结合 ARIMA 与自回归神经网络进行实时资源负载预测,取得了较好的精度。Chen 等^[11]将长短期记忆网络(long short-term memory, LSTM)与极端梯度提升模型(extreme gradient boosting, XGBoost)相结合进行预测,而 Nie 等^[12]则采用多目标灰狼算法(multi-objective grey wolf optimizer, MOGWO)对多个神经网络模型权重进行优化。Chen 等^[13]将时间卷积网络(temporal convolutional network, TCN)与全

收稿日期:2025-05-16

基金项目:航空科学基金(2022Z067003001)资助

作者简介:常祎雯(2001—),硕士研究生

通信作者:李伟刚(1972—),副教授 e-mail:liweigang@nwpu.edu.cn

连接层和注意力机制结合。

这些研究多为迁移通用时序预测算法,当前许多基于深度学习的复杂模型虽然在处理多维度、高非线性数据时表现出了较高的预测精度,但其较高的计算复杂度和资源需求使其难以直接应用于嵌入式场景。而单一或简单组合模型对于复杂数据的预测效果有限,且未充分考虑实时性和资源受限等实际因素,导致预测方法难以适用于复杂的嵌入式云原生环境。

目前公开的嵌入式容器应用数据集较少,如基于边缘计算环境的 EdgeBenchDatasetD 与用于物联网设备的 RIOTBenchmarkDataset,在样本规模与场景多样性方面均存在不足。为此,本文基于机载嵌入式环境构建了符合嵌入式云原生特征的容器应用数据集 Acore-Data,并结合 Alibaba-Cluster-Trace-Microservices-v2022 的特征进行数据对比与优化,以支持负载建模与性能分析。

嵌入式云原生环境具有低延迟、高可靠性与强实时性要求,典型机载平台的 CPU 配额不超过 1 C,内存容量约为 512 MB~2 GB,预测响应需在 200 ms 以内。实时操作系统(real-time operating system, RTOS)因此成为关键支撑,以天脉 3 为代表的高性能 RTOS 通过任务调度、隔离机制与容器化支持,保证了多任务并发下的实时性与安全性,为嵌入式云原生研究提供了稳定实验平台。

因此,本文引入数据降维^[14]和嵌入式环境特性,在减少计算开销的同时提高预测精度,提出基于完全集合经验模态分解与稀疏自注意力机制的轻量级负载预测模型(CEEMDAN-Informer),在有限算力条件下实现高精度、低延迟的预测,为嵌入式容器资源管理与调度提供有效支撑。

1 嵌入式云原生容器应用数据集构建

本文构建的嵌入式云原生容器应用数据集,充分考虑了嵌入式环境的特殊需求,特别是机载系统在资源和实时性方面的严格要求。数据集以云原生应用的数据属性和机载环境下的应用信息为构建原则,参考阿里巴巴、谷歌和华为等公司的公开容器应用数据集架构特征,设定合理的构建原则以确保数据的准确性、可靠性和完整性。参考 Alibaba-

Cluster-Trace-Microservices-v2022 数据集中的数据信息,分析得到本文数据集所需的数据属性。以云原生应用的数据属性和机载环境下的应用信息为构建数据集的基本原则,着重考虑嵌入式系统的可靠性、安全性及资源受限等特征,为后续构建嵌入式云原生容器应用数据集提供了全面的指导基础和实际参考。嵌入式云原生容器应用数据集所需的数据属性如表 1 所示。

表 1 嵌入式云原生应用数据属性

属性	描述
应用类型	在嵌入式环境中不同类型的应用具有不同优先级
运行时长	一次任务运行的时间长度
时间戳	数据采集的时刻
容器信息	容器应用的名称及所占用的 CPU、内存等资源的数据
节点信息	嵌入式板卡机器的信息

深入分析嵌入式云原生环境与通用云原生环境的异同。机载嵌入式系统常应用于飞行控制、导航、通信等关键任务中,对系统的可靠性、安全性和资源消耗有着较高要求。因此通过调研分析将任务关键型、数据处理型和通信传输型这 3 类应用的资源需求信息作为数据集的构建特征,应用的具体信息如表 2 所示。

表 2 机载环境中的应用信息

应用类型	实例	资源需求
任务关键型	飞行控制、导航系统	计算密集型,需要较强的计算资源来支持复杂的算法处理,以便快速计算并调整飞行状态,对存储的需求较低
数据处理型	监控、故障诊断系统	存储密集型,需要较大的存储资源,用于保存历史数据、诊断日志等信息,对计算资源需求较低
通信传输型	通信导航、信息传输系统	通信应用,需要较高的通信带宽,对计算和存储资源的需求相对较低

在数据集构建过程中,依托飞腾 D2000 开发板搭载的天脉 3 操作系统,设计符合嵌入式环境的数据采集方式,并建立高效的存储结构,确保数据来源的真实性与可靠性。飞腾 D2000 开发板作为国产高性能嵌入式平台,具备强大的计算能力和丰富的接口扩展,而天脉 3 操作系统提供完善的安全机制、

实时响应能力和全面的系统服务,确保了嵌入式应用的安全性、稳定性和高效性。容器通过天脉容器编排系统进行部署,在 3 类典型机载云原生应用中,每类运行 3 个实例,每个实例运行 3 个副本,采集间隔设置为 1 s,调用编排系统的 API。基于前文所述的数据原则,获取机器信息、容器运行时间、容器名称以及容器 CPU、内存等资源的实时使用情况,并实时获取负载信息,最终将这些数据本地存储。

在数据清洗环节,针对数据缺失和冗余问题进行了有效处理。原始数据集包含约 50 万条容器负载信息,由于容器数据冗杂,分析特定容器负载变化存在困难。为便于分析特定容器的负载变化,按 Pod 名称将数据分为多个子数据集。由于同一时间戳下可能存在多条重复记录,仅保留每个时间戳的第一条数据。对于容器负载数据中的缺失值,采用了三次指数平滑法进行填补。三次指数平滑法的基本思想是通过历史数据的加权平均计算缺失值,其原理是将时间序列分解为多个部分,根据历史数据与当前数据的距离分配权重,权重随着距离增大呈指数级衰减,并最终趋近零。由于容器资源负载数据的变化幅度较大,平滑系数取值为 0.7。

该方法具有较强的适应性,能够根据数据变化自动调整,因此对负载时序数据有较好的填补效果。此外,三次指数平滑法计算简便,其递归特性使得数据填补无需维护极少量的状态参数,有效节省存储空间和处理时间。基于上文中的数据原则,app1-c2-flightsim-ujq47x52qx-ujifo 数据集包括机器信息、容器的运行时间、容器名称以及容器 CPU、内存等资源的实时使用情况。以其子数据集中的 CPU 利用率数据为例,基于指数平滑法的填补效果如图 1 所示,红色点表示填补计算的结果。可见,提出的缺失数据处理方法能够较好地完成对容器资源负载数据的平滑处理。

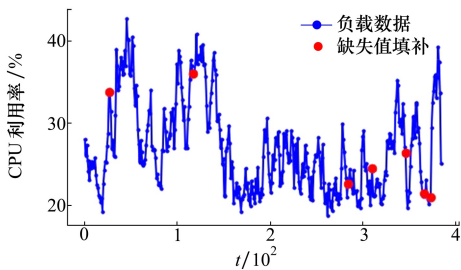


图 1 基于指数平滑法的缺失值处理

机载云原生应用数据,共包含 9 个应用实例和 27 个实时负载数据集,平均每个数据集约有 1.8 万条样本点,覆盖高、中、低负载阶段的典型特征。实验结果表明,当训练样本减少至 30% 时,预测误差仅上升约 4%,说明该数据规模足以支撑模型训练与验证。

为评估 Acore-Data 数据集的质量与代表性,本文从完整性、一致性和多样性 3 个方面进行了分析。完整性方面,采用 1 s 采样周期与多副本冗余机制,数据缺失率低于 0.5%,时序连续性达 99.8%;经三次指数平滑填补后,噪声占比降低约 15%。一致性方面,利用重复检测与 3σ 异常剔除法清洗数据,重复率控制在 1% 以内,保证样本可靠性。多样性方面,数据覆盖任务关键型、数据处理型和通信传输型 3 类负载,分别代表计算密集、存储密集与通信密集任务,其 CPU 与内存利用率方差显著(标准差 > 0.3),体现良好的负载差异性。

综上,Acore-Data 数据集具备较高的真实性、稳定性和代表性,可为嵌入式容器负载预测研究提供可靠支撑。后续将进一步扩展至工业控制、车载计算与边缘物联网等场景,以增强数据集的普适性。

2 负载预测模型

在嵌入式容器化应用中,受限的计算资源、实时性要求和复杂多变的负载特性使得精准的负载预测成为挑战。为了解决这一问题,本文在 Acore-Data 数据集基础上提出了基于信号分解的轻量级负载预测模型,模型架构如图 2 所示。该架构设计充分结合了 CEEMDAN 与 Informer 模型各自的优势。

CEEMDAN 作为一种优化的信号分解算法,能够将非线性、高复杂度的原始负载数据分解为若干相对平稳、复杂度更低的分量,使得数据中的潜在模式和局部趋势更加清晰,从而为后续的预测提供更具信息量和代表性的输入特征。Informer 模型则针对传统 Transformer 模型在长序列时序预测中的性能瓶颈,提出了基于概率的多头稀疏自注意力机制、蒸馏自注意力机制和并行生成预测序列的优化设计,显著降低了计算复杂度和内存消耗,并有效避免误差积累问题。这使得 Informer 模型能够在资源受限的嵌入式环境中快速生成高精度预测序列。

CEEMDAN-Informer 负载预测模型不仅能够充分挖掘负载数据的深层特征,还能在资源受限的嵌

入式场景中实现高效、低延迟的预测,为容器资源调度和系统优化提供了可靠支持,有效提升了嵌入式应用的资源利用率和系统稳定性。

提供更加清晰且有规律的输入。

CEEMDAN 算法是对经验模态分解 (EMD) 的改进,通过引入自适应噪声和迭代过程,解决了模态混叠和端点效应问题。CEEMDAN 算法的详细实现步骤如下:

1) 添加高斯白噪声。在初始信号序列 $x(t)$ 中加入高斯白噪声 $\sigma_0\omega_i(t)$,形成新的信号序列 $x_i(t)$, $i = (1, 2, \dots, n)$,如(1)式所示。

$$x_i(t) = x(t) + \sigma_0\omega_i(t) \quad (1)$$

式中: i 表示加入高斯白噪声的次数; σ 为噪声标准差,与噪声相乘以调整噪声的强度; $\omega_i(t)$ 为第 i 次添加的高斯白噪声信号。

2) 多次重构。对带有噪声的新信号序列 $x_i(t)$ 进行多次随机重构,每次重构都可能得到不同的信号结果,如进行 N 次得到 N 个重构信号序列。

3) EMD 分解。对 N 次重构得到的信号序列进行 EMD 分解,每次分解都可能得到不同的 IMF,至此共得到了 N 个 IMF 分量。

4) 取平均。对这 N 个分量进行取平均操作,得到最终的第一个模态分量 $I_{MF_1}(t)$,如(2)式所示。

$$I_{MF_1}(t) = \frac{1}{N} \sum_{i=1}^N I_{MF_1^i}(t) \quad (2)$$

5) 残差处理。将最终的 $I_{MF_1}(t)$ 与初始信号序列 $x_1(t)$ 相减,得到第一个残差 $R_1(t)$,如(3)式所示。

$$R_1(t) = x_1(t) - I_{MF_1}(t) \quad (3)$$

6) 重复迭代。将残差作为新的初始信号序列,不断重复步骤 1)~5),持续得到 IMF_s 分量和残差余量。设经过 n 次迭代后,残差不可以再被分解,至此完成了对初始信号序列的分解,如(4)式所示。

$$x(t) = \sum_{i=1}^m x_i(t) = \sum_{i=1}^m I_{MF_i}(t) + R_n(t) \quad (4)$$

在负载预测阶段,CEEMDAN 分解得到的各个本征模态函数 (IMFs) 和残差项 (Res) 将作为输入特征提供给 Informer 预测模型。Informer 是一种基于自注意力机制的高效时序预测模型,特别适用于处理长序列数据,并能够捕捉数据中的长期依赖关系。与传统的时序预测模型相比,Informer 通过其稀疏自注意力机制有效降低计算复杂度,在处理大规模数据时仍能保持高效性。Informer 模型首先使用基于概率的多头稀疏自注意力机制 (probSparse self-attention),然后在不同自注意力层之间融入蒸馏自注意力机制 (self-attention distilling),最后采用解码器

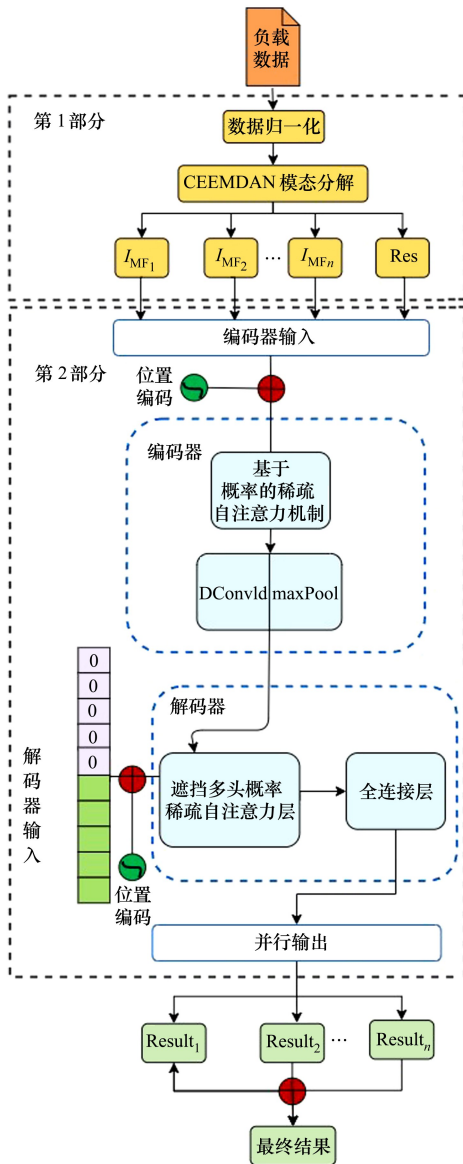


图 2 CEEMDAN-Informer 模型架构图

在数据预处理阶段,首先对原始数据进行归一化处理,接着应用 CEEMDAN 算法对数据进行模态分解。CEEMDAN 是一种有效的信号分解方法,它将原始信号分解为多个本征模态函数 (IMFs) 和一个残差项 (Res),从而提取信号中的不同频率成分并去除噪声干扰。通过这种方式,CEEMDAN 将复杂信号分解为多个简单成分,使每个 IMF 分量代表数据中不同的周期性特征,残差项则表示低频趋势。经过分解后,噪声成分被有效去除,为后续预测任务

(decoder) 架构并行生成预测序列 (one forward operation)。

具体实现流程如图 3 所示,该图清晰展示了数据预处理、模态分解、预测输入以及最终预测结果生成的全过程。

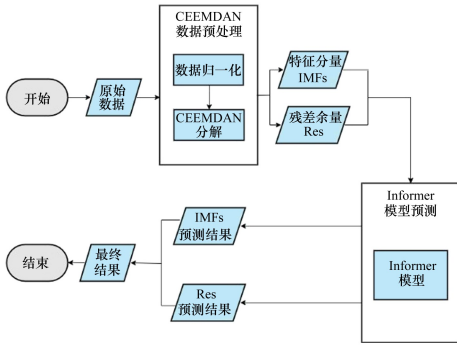


图 3 CEEMDAN-Informer 模型的算法流程

在本模型中,IMFs 分量和残差项作为输入,Informer 模型通过学习这些输入,能够捕捉负载数据中的周期性波动和潜在趋势,从而进行准确的负载预测。最终,通过累加操作,将各个 IMFs 分量和残差项的预测结果结合起来,得到最终的负载预测值。此累加过程将预测结果从各个频率成分重建为原始数据的全局预测值。整个预测过程将分解与重建相结合,能够更加精确地捕捉负载数据的变化趋势,并有效减少噪声对预测结果的影响。

3 嵌入式云原生容器负载预测实验

对于嵌入式云原生容器负载预测实验,为衡量模型的预测精确度,本文选择了 3 种在预测实验中最常被使用的评估指标,即平均绝对误差 (mean absolute error, MAE)、均方误差 (mean squared error, MSE) 和均方根误差 (root mean squared error, RMSE),用于评价负载预测效果。由于 Acore-Data 数据集中的容器数量庞大,为了突出实验的核心和重点,本次实验结果仅对容器 app1-c2-flightsim-ujq47x52qx-ujifo 的相关数据进行介绍和分析。在实际环境中,集群的性能通常受 CPU 资源、内存资源、磁盘吞吐率和网络带宽的影响,而在嵌入式云原生环境中磁盘吞吐率和网络带宽对集群的影响较低,为提高核心因素的效果,本文的负载预测仅基于 CPU 和内存资源。同时,使用文献 [13] 中层次分析

法的 CPU 与内存比例,计算得到权重向量 $\mathbf{W} = (0.666\ 6, 0.333\ 4)^T$, 以此对数据进行降维,根据 (5) 式计算出负载评价度量值 L_t , 代表 t 时刻的负载值。

$$L_t = \mathbf{W} \times \begin{pmatrix} L_{\text{cpu}} \\ L_{\text{mem}} \end{pmatrix} \quad (5)$$

式中, L_{cpu} 和 L_{mem} 分别代表 t 时刻 CPU 利用率和内存利用率进行归一化处理之后的值。

为验证该指标的合理性与有效性,进行了对比实验。结果表明, L_t 相较于 CPU 和内存单项指标的变化更加平滑,能更准确地反映系统整体资源利用率。采用 L_t 作为统一输入后,CEEMDAN-Informer 模型的 MAE、MSE 及 RMSE 分别降低约 1.3%, 2.8%, 2.0%, 且与 CPU、内存利用率的 Pearson 相关系数分别为 0.93 和 0.87。说明该负载度量方法在提高预测精度的同时降低了特征维度,具有良好的代表性与实用性。

本次实验按 4 : 1 比例将数据集划分为训练集和测试集。使用 (5) 式对容器 app1-c2-flightsim-ujq47x52qx-ujifo 的测试集进行负载计算,计算所得结果如图 4 所示。

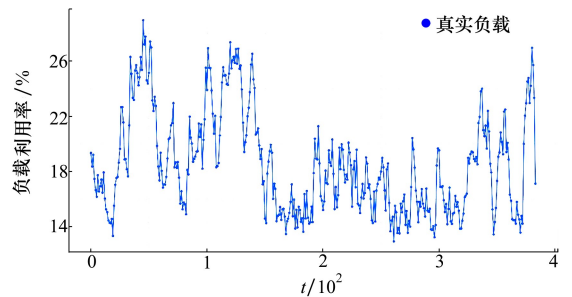


图 4 app1-c2-flightsim-ujq47x52qx-ujifo 负载数据图

3.1 有效性实验

为了验证 CEEMDAN-Informer 模型的有效性,本文进行了有效性实验。对所有分量的预测结果进行重构,从而得到 CEEMDAN-Informer 的最终预测结果,如图 5 所示。

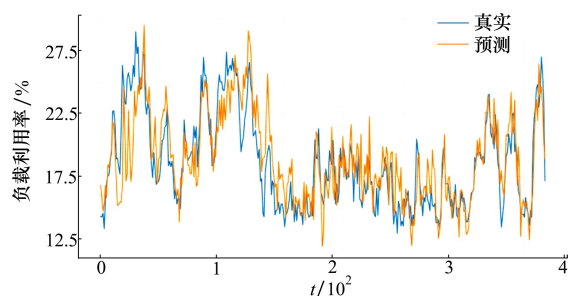


图 5 CEEMDAN-Informer 预测结果

观察图 5 可见,CEEMDAN-Informer 的拟合曲线与初始曲线十分贴近,表明 CEEMDAN-Informer 在嵌入式云原生数据集上有良好的预测效果,在损失率方面也有良好表现,具体如表 3 所示。表 3 中记录了 6 个 IMF 特征分量和残差分量 Res 的损失率合并后 IMF_s 的损失率。

表 3 损失率结果

分量	指标		
	MAE	MSE	RMSE
I_{MF_1}	2.127 1	11.812 3	3.436 9
I_{MF_2}	0.694 3	0.321 9	0.567 4
I_{MF_3}	0.486 6	0.311 8	0.558 4
I_{MF_4}	1.764 4	6.683 3	2.585 2
I_{MF_5}	0.492 8	0.273 5	0.523 0
I_{MF_6}	0.322 6	0.127 7	0.357 4
残值	0.107 9	0.048 3	0.219 8
最终结果	3.509 2	20.009 5	4.473 2

3.2 对比实验

为验证 CEEMDAN-Informer 模型在预测精度方面的优越性,本文设计对比实验,引入 TCN、Transformer、LSTM-XGBoost^[11] 和 MOGWO^[15] 4 种典型模型进行比较。其中,TCN 采用因果卷积结构以保持时间依赖顺序; Transformer 利用自注意力机制高效

捕获长距离依赖;LSTM-XGBoost 结合了 LSTM 的时序特征提取与 XGBoost 的回归优势;MOGWO 通过多目标灰狼优化在全局与局部搜索间取得平衡。

在进行容器资源负载预测时,预测步长对后续部署优化方案的设计具有重要影响。因此,本文针对不同步长进行了实验,将步长分别设置为 4,8,16 和 32,以验证模型在不同步长条件下的预测效果。同时,为了确保实验的严谨性,对比模型尽可能采用了与 CEEMDAN-Informer 一致的全局变量和参数设置,若网络结构无法完全实现,则根据经验法进行调优,以找到最佳的参数组合。为了证明本文提出的预测方法在嵌入式云原生容器负载预测领域具有较高的准确度和适用性,本节设计了多种对比实验。同时,为了保证实验结果的有效性和可靠性,所有预测模型的输入都是 Acore-Data 数据集。如图 6 所示,可以看出当预测步长分别为 4,8,16,32 时,CEEMDAN-Informer 的预测性能指标均明显优于其他模型,说明该模型在多步预测方面具有更高的精度。

如表 4 所示,分析对比实验的实验结果后发现本文提出的 CEEMDAN-Informer 负载预测模型的预测效果最好。

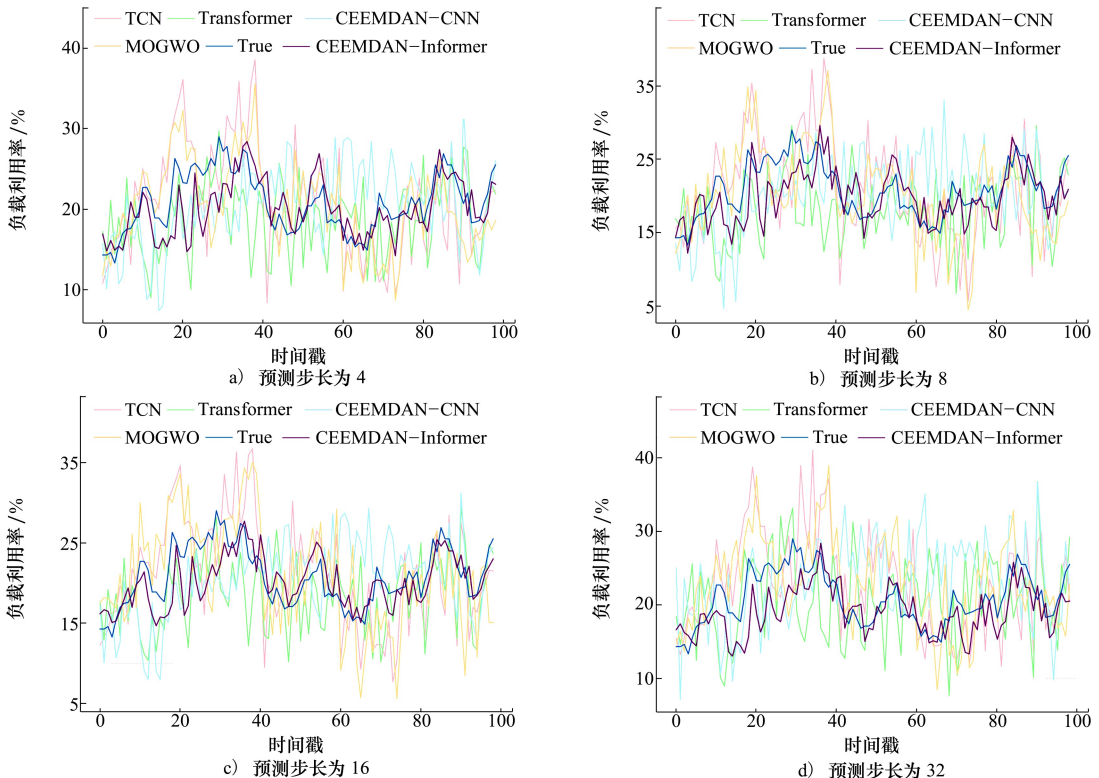


图 6 对比试验预测结果

表 4 对比实验结果

模型	指标	预测步长 4	预测步长 8	预测步长 16	预测步长 32
TCN	MAE	4.085 4	4.210 1	4.604 1	5.043 2
	MSE	28.642 2	30.944 4	36.501 2	45.893 8
	RMSE	5.351 8	5.562 7	6.041 6	6.774 5
Transformer	MAE	4.219 2	4.302 3	4.418 4	5.183 3
	MSE	27.042 1	27.945 1	32.044 5	39.217 7
	RMSE	5.200 2	5.286 3	5.660 7	6.262 4
CEEMDAN-CNN	MAE	4.352 2	4.731 5	4.768 5	5.344 6
	MSE	35.348 2	34.983 2	37.054 1	41.440 4
	RMSE	5.192 6	5.698 6	6.035 5	6.434 6
MOGWO	MAE	4.153 1	4.382 9	4.850 8	4.977 8
	MSE	28.163 5	31.049 4	35.071 2	44.012 6
	RMSE	5.306 6	5.572 2	5.922 1	6.634 2
CEEMDAN-Informer	MAE	3.509 2	4.039 8	4.305 6	4.570 7
	MSE	20.009 5	23.466 2	28.993 9	35.272 9
	RMSE	4.473 2	4.844 2	5.384 6	5.939 1

如表 4 所示,分析对比实验结果后发现本文提出的 CEEMDAN-Informer 负载预测模型预测效果最好。与 TCN 模型相比,CEEMDAN-Informer 的预测精度更高,当预测步长分别为 4,8,16,32 时,MAE 分别降低了 4.3%,4.1%,6.4%,9.3%,MSE 分别降低了 30.1%,24.1%,20.5%,23.1%,RMSE 分别降低了 16.4%,12.9%,10.8%,12.3%。与 Transformer 模型相比,CEEMDAN-Informer 的预测精度更高,当预测步长分别为 4,8,16,32 时,MAE 分别降低了 7.3%,6.1%,2.6%,11.8%,MSE 分别降低了 26.0%,16.0%,9.5%,10.1%,RMSE 分别降低了 14.0%,8.4%,4.9%,5.2%。与 CEEMDAN-CNN 模型相比,CEEMDAN-Informer 的预测精度更高,当预测步长分别为 4,8,16,32 时,MAE 分别降低了 19.4%,14.6%,10.8%,14.5%,MSE 分别降低了 43.4%,32.9%,21.7%,14.8%,RMSE 分别降低了 13.9%,14.9%,10.8%,7.7%。与 MOGWO 模型相比,CEEMDAN-Informer 的预测精度更高,当预测步长分别为 4,8,16,32 时,MAE 分别降低了 15.5%,7.8%,11.2%,8.2%,MSE 分别降低了 29.0%,24.4%,17.3%,19.9%,RMSE 分别降低了 15.7%,13.1%,9.1%,10.5%。

具体来看,在时序数据预测中,网络模型结构是否适配数据集的特点对预测精度具有重要影响。在大多数情况下,常见的时序预测网络模型如 TCN、CEEMDAN-CNN 和 MOGWO,通常是较为合适的选择,尽管可能存在一定的精度偏差,但整体上仍能表现出良好的预测性能。但是 TCN、CEEMDAN-CNN

和 MOGWO 对较长的时间序列的预测能力较弱,表现出一定的局限性,尤其是随着序列跨度的增大,其预测能力显著下降。相比之下,对于 Transformer 模型而言,其与 Informer 模型有着相似的架构,对于长序列的预测能力较强,然而,随着序列长度的进一步增加,Transformer 的预测能力也会逐渐减弱。不论是 TCN、CEEMDAN-CNN、MOGWO 还是 Transformer,它们的预测效果都不如通过 CEEMDAN 信号分解算法对数据进行降维处理后再预测的效果好。这表明,先利用 CEEMDAN 对数据进行分解以提取关键分量,然后结合 Informer 模型进行预测的策略,可以有效提升预测精度。本文提出的 CEEMDAN-Informer 模型验证了这一点,其预测结果显著优于单一模型方法,进一步证明了该方法在提高长序列预测精度方面的可行性与有效性。

通过对不同预测步长(4,8,16,32)下的结果比较可知(见表 4 和图 6),随预测步长增加,所有模型 MAE、MSE 和 RMSE 均呈上升趋势,说明负载序列在长时间跨度下的不确定性更高。相比之下,CEEMDAN-Informer 模型在全尺度表现优异:步长为 32 的 MAE(4.5707)优于部分模型步长为 4 的表现。特别是在 16,32 步长区间,其 MAE 增幅(6.1%)远低于 Transformer(17.3%),展现出更强的长序预测稳定性。其优势来源于 CEEMDAN 的信号去噪能力与 Informer 稀疏自注意力对远程依赖的高效建模。

此外,为评估模型在嵌入式平台中的适应性,本文进一步统计了其计算资源占用与推理延迟。实验

结果显示, CEEMDAN-Informer 在飞腾 D2000 平台上运行时平均 CPU 占用率低于 28%, 单步预测延迟约 180 ms, 能耗较 Transformer 降低 22%。由此可见, 该模型在保证预测精度的同时具备良好的实时性与能效, 适用于资源受限的嵌入式环境。

3.3 极端资源受限条件下的性能实验

为验证 CEEMDAN-Informer 模型在极端资源受限环境下的适应性, 本文在飞腾 D2000 平台上利用 cgroups 限制 CPU 配额和内存上限, 设置 R0 (1.0 C/2048 MB)、R1 (0.5 C/1024 MB)、R2 (0.25 C/512 MB) 和 R3 (0.125 C/256 MB) 4 种资源配置。在各配置下, 对 3 类典型嵌入式负载 (任务关键型、数据处理型、通信传输型) 进行测试, 并与 Transformer、TCN 和 CEEMDAN-CNN 模型对比。

实验结果表明, 随着资源限制增强, 所有模型的预测精度均有所下降, 但 CEEMDAN-Informer 的 MAE 与 RMSE 增长最小。在 R3 极端条件下, 模型 MAE 仍保持 4.71、RMSE 为 6.08, 较 Transformer 分别降低约 11% 和 9%, 单步预测延迟 178 ms, 截止期违约率 (DMR) 仅 0.7%, 满足 200 ms 实时性要求。资源消耗方面, 平均 CPU 占用 27%, 峰值内存 480 MB, 比 Transformer 低约 25%, 单次预测能耗减

少 22%。在预测-调度闭环实验中, 系统 CPU 与内存利用率分别提升 12% 和 15%, 服务等级协议 (SLA) 违约率下降 18%。

综上, CEEMDAN-Informer 模型在负载预测精度、实时性及能效方面均表现优异, 能在极端资源受限的嵌入式环境中稳定运行, 具备良好的工程可部署性。

4 结 论

现有通用容器部署方案难以应对嵌入式云原生环境中负载的复杂性与实时性需求。针对当前缺乏高质量嵌入式容器应用数据集的现状, 本文以航空机载为典型场景, 基于天脉 3 操作系统与飞腾 D2000 平台, 构建了机载嵌入式基础环境, 并完成了 Acore-Data 数据集的采集与清洗, 为相关研究提供了真实、可用的数据支撑。

在此基础上, 本文提出了一种结合 CEEMDAN 与 Informer 的轻量级负载预测模型, 兼顾预测精度、实时性与资源约束, 在嵌入式云原生环境中表现出良好的适应性和实用性, 为后续的资源管理与容器调度提供了有效支撑。

参考文献:

- [1] Li S, Xu L D, Zhao S. The internet of things: a survey[J]. Information Systems Frontiers, 2015, 17: 243-259.
- [2] Marwedel P. Embedded system design: embedded systems foundations of cyber-physical systems, and the internet of things[M]. New York: Kluwer Academic Publishing, 2021: 21-22.
- [3] 张锋, 杨粤涛, 高伟林, 等. 小型化低功耗机载显示器图形系统设计与实现[J]. 电讯技术, 2022, 62(6): 813-819.
Zhang Feng, Yang Yuetao, Gao Weilin, et al. Design and implementation of miniaturized low-power airborne display graphic system[J]. Telecommunication Engineering, 2022, 62(6): 813-819. (in Chinese)
- [4] 李阳, 苗张旺. 数字经济时代云计算产业发展与挑战[J]. 数字经济, 2023(增刊 2): 78-83.
Li Yang, Miao Zhangwang. Development and challenges of the cloud computing industry in the digital economy era[J]. Digital Economy, 2023(S2): 78-83. (in Chinese)
- [5] Gannon D, Barga R, Sundaresan N. Cloud-native applications[J]. IEEE Cloud Computing, 2017, 4(5): 16-21.
- [6] 徐胜超, 熊茂华. 基于遗传算法的容器云资源配置优化[J]. 计算机与现代化, 2022, 1(1): 108-112.
Xu Shengchao, Xiong Maohua. Resource allocation optimization of container cloud based on genetic algorithm[J]. Computer and Modernization, 2022, 1(1): 108-112. (in Chinese)
- [7] 梁荣华. 基于资源负载预测的容器云弹性伸缩策略研究[D]. 桂林: 桂林理工大学, 2022.
Liang Ronghua. Research on elastic scaling strategy of container cloud based on resource load prediction[D]. Guilin: Guilin University of Technology, 2022. (in Chinese)
- [8] Roy N, Dubey A, Gokhale A. Efficient autoscaling in the cloud using predictive models for workload forecasting[C]//2011 IEEE 4th International Conference on Cloud Computing, New York, 2011: 500-507.
- [9] Calheiros R N, Masoumi E, Ranjan R, et al. Workload prediction using ARIMA model and its impact on cloud applications' QoS[J]. IEEE Trans on Cloud Computing, 2014, 3(4): 449-458.
- [10] Sudhakar C, Kumar A R, Siddhartha N, et al. Workload prediction using ARIMA statistical model and long short-term memory recurrent neural networks[C]//2018 International Conference on Computing, Power and Communication Technologies, New

York, 2018: 600-604.

- [11] Chen Z, Liu J, Li C, et al. Ultra short-term power load forecasting based on combined LSTM-XGBoost model[J]. Power System Technology, 2020, 44(2): 614-620.
- [12] Nie Y, Jiang P, Zhang H. A novel hybrid model based on combined preprocessing method and advanced optimization algorithm for power load forecasting[J]. Applied Soft Computing, 2020, 97: 1068089-106811.
- [13] Chen W, Lu C, Ye K, et al. RPTCN: resource prediction for high-dynamic workloads in clouds based on deep learning[C]// 2021 IEEE International Conference on Cluster Computing, Portland, 2021: 59-69.
- [14] Jia W, Sun M, Lian J, et al. Feature dimensionality reduction: a review[J]. Complex & Intelligent Systems, 2022, 8(3): 2663-2693.
- [15] 梁强, 徐永航, 李永亮, 等. 基于 MOGWO 的 45# 钢表面激光抛光工艺参数多目标优化[J]. 表面技术, 2024, 53(10): 173-182.
- Liang Qiang, Xu Yonghang, Li Yongliang, et al. Multi-objective optimization of laser polishing process parameters for 45# steel surface based on MOGWO[J]. Surface Technology, 2024, 53(10): 173-182. (in Chinese)

Study on load prediction methods of embedded container-based applications

Chang Yiwen¹, Li Weigang¹, Wu Junsheng¹, Zhang Shenghua¹, Li Yi²

(1.School of Software, Northwestern Polytechnical University, Xi'an 710072, China;)
(2.AVIC the First Aircraft Institute, Xi'an 710089, China)

Abstract: Currently, the use of virtual container architectures in embedded computing environments is becoming increasingly popular, offering new possibilities for resource scheduling to balance load fluctuations. However, effective scheduling relies heavily on accurate load prediction, and existing research in this area suffers from a lack of dedicated datasets and inadequate adaptation of existing prediction methods to the characteristics of embedded applications. Focusing on an avionics embedded application scenario, a dataset tailored to containerized embedded environments is constructed. To address the issues of low prediction accuracy and computational inefficiency, a light-weight load prediction model is proposed by integrating the CEEMDAN algorithm with the Informer model. The CEEMDAN algorithm enhances the modeling accuracy by decomposing the time series data, while the Informer model reduces the computational complexity and memory consumption through a sparse self-attention mechanism. Experimental results demonstrate that, comparing with the mainstream time series prediction methods, the present model achieves an average reduction of about 10% in prediction errors and is well-suited for embedded application scenarios.

Keywords: embedded virtual containers; container cloud; load prediction; avionics software

引用格式: 常祎雯, 李伟刚, 武君胜, 等. 面向嵌入式容器应用的负载预测方法研究[J]. 西北工业大学学报, 2026, 44(1): 125-133.

Chang Yiwen, Li Weigang, Wu Junsheng, et al. Study on load prediction methods of embedded container-based applications[J]. Journal of Northwestern Polytechnical University, 2026, 44(1): 125-133. (in Chinese)