

• 机械工程 •

DOI:10.12454/j.jsuese.202300136



本刊网刊

基于策略梯度Informer模型的滚动轴承剩余寿命预测

熊佳豪¹, 李 锋^{1*}, 汤宝平², 汪永超¹, 罗 玲³

(1.四川大学 机械工程学院, 四川 成都 610065; 2.重庆大学 机械传动国家重点实验室, 重庆 400044;
3.中国测试技术研究院, 四川 成都 610021)

摘要: 典型的编码器-解码器——Transformer存在二次时间复杂度、高内存使用及模型结构复杂等固有限制, 造成Transformer用于滚动轴承剩余寿命(RUL)预测会表现出较低预测精度和较低计算效率的问题。为此, 提出一种新型编解码器——策略梯度Informer(PG-Informer)模型, 并将其应用于滚动轴承RUL预测。首先, 在PG-Informer的新型编解码器体系结构——Informer中设计了概率稀疏自注意力(PSSA)机制替代Transformer中原有的自注意力机制, 以提高非线性逼近能力并减少时间和空间复杂度; 然后, PG-Informer采用自注意力蒸馏(SAD)操作减少参数维度和参数量, 并提高了对时间序列的预测鲁棒性; 此外, PG-Informer的生成式解码器对解码输入进行一步解码输出预测结果, 无需动态多步解码, 提升了对时间序列的预测速度; 最后, 构造了策略梯度学习算法来提高对PG-Informer参数的训练速度。PG-Informer的以上优势使所提出的基于PG-Informer模型的滚动轴承RUL预测方法可以获得较高的预测精度、较好的鲁棒性和较高的计算效率。对辛辛那提大学智能维护系统中心的1号滚动轴承的RUL预测实验结果表明, 所提出方法预测得到的RUL值为963 min, 其RUL预测误差仅为6.50%, 比基于Transformer的RUL预测方法预测精度更高、预测误差更小、鲁棒性更好; 所提出方法所耗费的RUL预测时间仅为132.37 s, 比基于Transformer的RUL预测方法的预测时间更短。以上实验结果验证了所提出方法的有效性。

关键词: Informer模型; 概率稀疏自注意力机制; 策略梯度; 滚动轴承; 剩余寿命预测

中图分类号: TH17; TP391.6

文献标志码: A

文章编号: 2096-3246(2024)04-0273-14

滚动轴承被广泛应用于航空航天、轨道交通、风力发电、装备制造等工业领域, 是航空发动机、高速铁路机车、风电机组、数控机床等大型机械装备的关键旋转部件。但滚动轴承往往处在高速、重载的连续运行状态下, 一旦发生故障会引发连锁反应, 轻则导致设备故障、经济损失; 重则造成严重的安全事故, 危害人身安全。因而, 如果能准确地提早预测出滚动轴承的剩余寿命(RUL), 将有助于避免机械装备发生事故, 减少经济损失, 避免人员伤亡, 具有重要的经济价值和工程意义^[1-3]。

目前, 滚动轴承RUL预测方法可以分为3类: 基于物理模型的方法、基于先验知识的方法和基于人工智能的方法^[4-6]。然而, 对非线性、随机、动态变化

的滚动轴承退化过程的模拟会给物理模型引入大量不确定参数, 从而导致物理模型的参数难以估计。因此, 确定的滚动轴承退化过程仿真物理模型难以建立。而基于先验知识的RUL预测方法依赖专家知识和经验进行逻辑推理, 受限于各种规则, 因而也难以应用于滚动轴承RUL预测。不同于前两种方法, 以机器学习、深度学习为代表的人工智能RUL预测方法, 从滚动轴承的历史监测数据中提取特征, 进而建立基于特征数据和特征退化过程的模型来预测RUL。这类方法因具有较强大的特征提取能力而成为目前滚动轴承RUL预测的研究热点。如: 王久健等^[7]采用卷积长短时记忆神经网络(ConvLSTM)预测滚动轴承的RUL; 姚德臣等^[8]采用了一种融合注意力机制的

收稿日期:2023-02-27 修回日期:2023-08-09 网络出版日期:2024-06-03

基金项目:四川省中国制造2025四川行动资金项目计划(2019CDZG-22);机械传动国家重点实验室开放基金资助项目(SKLM-T-KFKT-201718);中央高校基本科研业务费(2022CDZG-12)

作者简介:熊佳豪(1999—),男,硕士生。研究方向:机械设备状态监测、故障诊断与寿命预测。E-mail:2838732973@qq.com

*通信作者:李 锋,副教授, E-mail: lifeng19820501@163.com

门控循环单元(AGRU)模型预测了滚动轴承的RUL; Qin等^[9]使用门控双注意单元神经网络(GDAU)进行滚动轴承RUL预测。然而,上述基于人工智能的滚动轴承RUL预测方法均存在一定缺陷。具体而言,ConvLSTM、AGRU和GDAU在长序列的条件下捕捉时序数据的空间局部信息与时间的互相依赖关系的能力会迅速衰减,即出现记忆力退化现象,因此,它们的泛化能力皆有不足,将它们应用于滚动轴承RUL预测均存在一定局限性。

Transformer是一种开始流行的编解码器,主要以自注意力(SA)机制为基础,可以捕捉输入序列中任意向量之间的依赖关系,实现了输入并行计算,相比于只能串行计算^[7]的LSTM、门控循环单元(GRU)等循环神经网络(RNNs)^[10],可明显提升对网络参数的训练速度,因此,Transformer在滚动轴承RUL预测中表现出一定应用前景^[11-13]。然而,Transformer仍然存在二次时间复杂度、高内存使用及编码器-解码器体系结构复杂(即参数维度和参数量较大)等缺陷,会引起RUL预测精度和计算效率降低等问题。针对Transformer的不足,提出一种新型编解码器——策略梯度Informer(PG-Informer)模型,并首次将其引入滚

动轴承RUL预测这一应用领域。相比Transformer,PG-Informer改善了非线性逼近能力,减少了时间和空间复杂度,减少了网络参数维度和参数量,提升了对时间序列的预测鲁棒性和预测速度,还进一步提高了对网络参数的训练速度。因此,基于PG-Informer模型的滚动轴承RUL预测方法可以获得比Transformer等机器学习方法更高的RUL预测精度、更好的鲁棒性和更高的计算效率。

1 策略梯度Informer模型

1.1 策略梯度Informer理论模型

在PG-Informer中,首先,设计了新型编解码器体系结构——Informer(主要由概率稀疏自注意力(PSSA)机制、自注意力蒸馏(SAD)操作、生成式解码器3部分组成)来提高PG-Informer的非线性逼近能力,减少其时间和空间复杂度,提高其对时间序列的预测鲁棒性和预测速度;接着,构造了策略梯度学习算法加快对PG-Informer参数的训练速度。PG-Informer的整体模型结构如图1所示。图1中, X_{token} 和 X_0 分别为输入序列 X_{feed} 的开始标记和预测值序列 y 的占位符, θ_t 为 t 时刻的参数集 θ 。

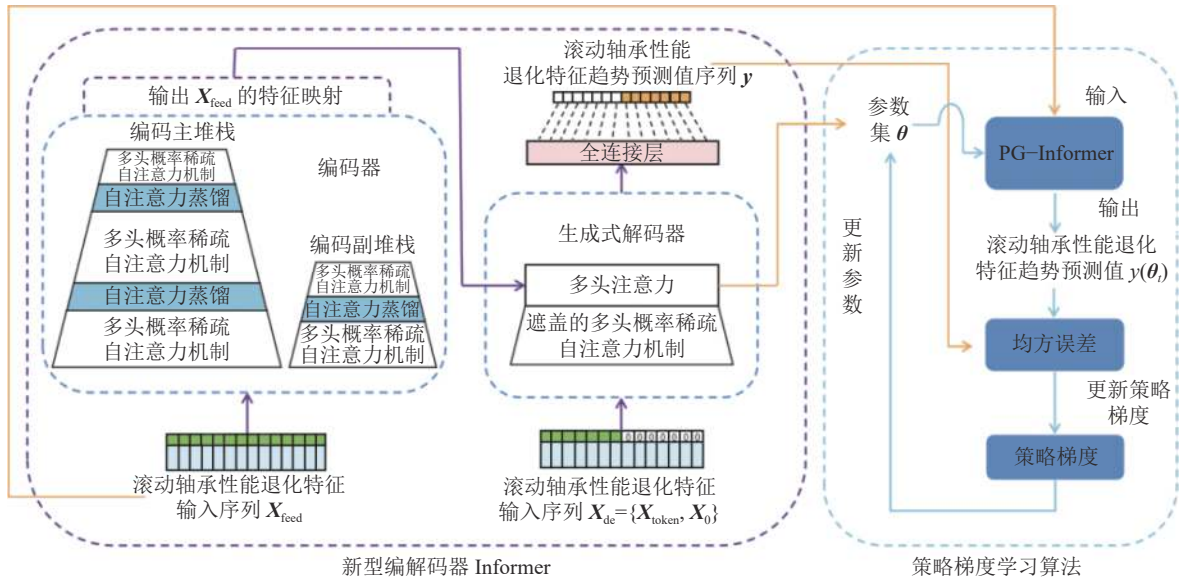


图 1 PG-Informer的模型结构

Fig. 1 Model structure of PG-Informer

假设输入PG-Informer的滚动轴承性能退化特征序列为 \mathbf{x} , $\mathbf{x} = \{x_1, x_2, \dots, x_{L_x}\}$, L_x 为 \mathbf{x} 的序列长度;同时,假设PG-Informer的输出,即滚动轴承性能退化特征趋势预测值序列为 \mathbf{y} , $\mathbf{y} = \{y_1, y_2, \dots, y_{L_y}\}$, L_y 为 \mathbf{y} 的序列长度。PG-Informer的非线性逼近能力和鲁棒性好、并行计算能力强、内存占用小等一系列优点得益于其概率稀疏自注意力(PSSA)机制的构造,但该机

制使其失去了学习序列位置信息(即滚动轴承性能退化特征时序信息)的能力。为此,在 \mathbf{x} 送入PG-Informer之前需要经过统一的输入表示。图2为PG-Informer的输入表示。假设经过输入表示的 \mathbf{x} 的特征维数变为 d_{model} ,则该统一的输入表示实现过程如下:

首先,使用固定位置嵌入来保存局部上下文(即保存滚动轴承性能退化特征数据的位置信息),固定

位置嵌入正弦项 $P_{(p,2j)}^E$ 和余弦项 $P_{(p,2j+1)}^E$ 为:

$$P_{(p,2j)}^E = \sin(p/(2L_x)^{2j/d_{\text{model}}}) \quad (1)$$

$$P_{(p,2j+1)}^E = \cos(p/(2L_x)^{2j/d_{\text{model}}}) \quad (2)$$

式中: p 为 x 的元素索引, $p = 1, 2, \dots, L_x$; j 为固定位置嵌入的元素下标, $j \in \{1, 2, \dots, \lfloor d_{\text{model}}/2 \rfloor\}$, $\lfloor d_{\text{model}}/2 \rfloor$ 为 d_{model} 的一半向下取整。

接着,为了调整滚动轴承性能退化特征数据的维度,用1维卷积滤波器将标量上下文 x_p (即滚动轴承性能退化特征)投影到 d_{model} 维向量 u_p 中,该过程表达如下:

$$u_p = x_p * f \quad (3)$$

式中, $*$ 为卷积运算, f 为1维卷积核。

于是,通过求和运算,得到输入矢量 x_{feed}^p :

$$x_{\text{feed}}^p = au_p + P_{(p,2j)}^E + P_{(p,2j+1)}^E \quad (4)$$

式中, a 为平衡标量投影和固定位置嵌入之间权重大小的因素, $P_{(p,2j)}^E$ 、 $P_{(p,2j+1)}^E$ 分别为元素值都为 $P_{(p,2j)}^E$ 、 $P_{(p,2j+1)}^E$ 的向量。通过以上统一的输入表示可以将原始滚动轴承性能退化特征序列的数据结构转化为适应PG-Informer输入的形式,这样就可以恢复PG-Informer对序列位置信息的学习能力。

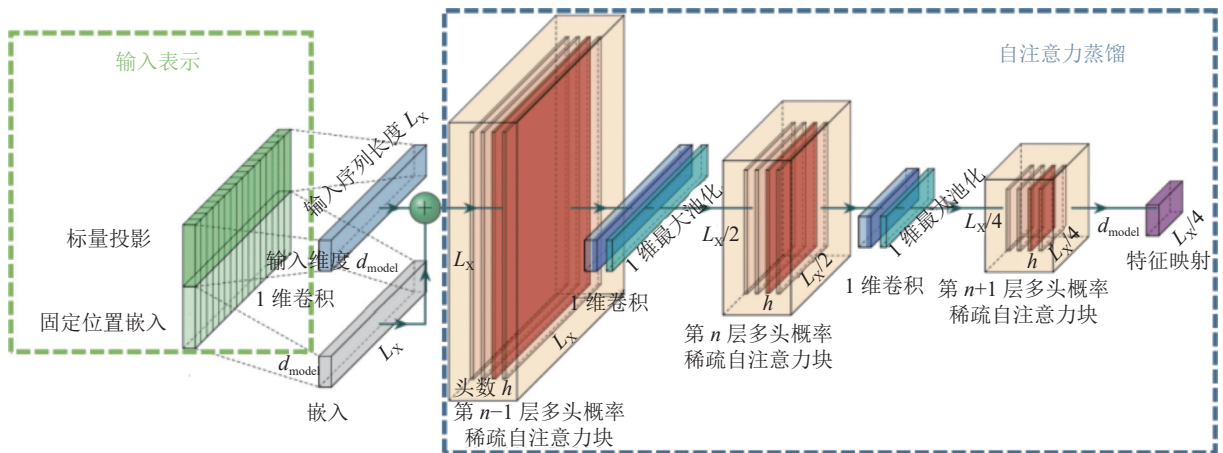


图3 PG-Informer 编码器中的单个编码堆栈示意图

Fig. 3 Schematic diagram of a single encoding stack in the PG-Informer encoder

1) PSSA机制

滚动轴承往往在变工况(变载荷、变转速)条件下运行,同时,其监测数据经常受到强环境噪声干扰,这导致提取的性能退化特征信噪比低、非平稳、非线性。为此,在PG-Informer的编码器中设计PSSA机制用于提取滚动轴承性能退化特征数据之间的长期依赖关系,以降低环境噪声干扰并增强PG-Informer对非平稳、非线性的性能退化特征的非线性逼近能力。

如上所述,经过输入表示之后,滚动轴承性能退化

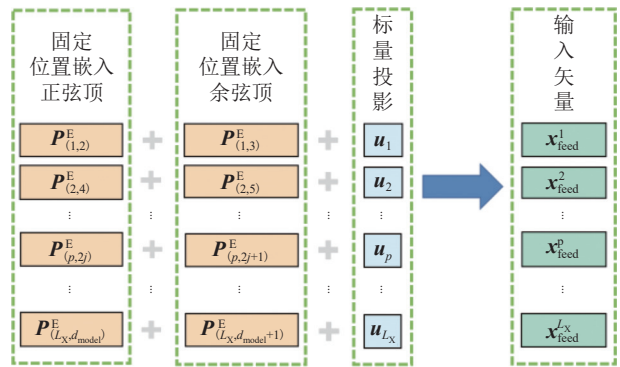


图2 PG-Informer的输入表示

Fig. 2 Input representation of PG-Informer

1.1.1 PG-Informer的编码器

图3为PG-Informer编码器中的单个编码堆栈示意图。如图3所示:首先,在PG-Informer编码器的多头自注意力块中采用PSSA机制提高其自身的非线性逼近能力并减少其时间和空间复杂度;其次,在编码器中采用自注意力蒸馏(SAD)操作来降低PG-Informer的网络维度和网络参数量;另外,还构建了如图1所示的将输入减半的编码主堆栈副本对齐编码主堆栈和编码副堆栈的输出维度,并将主、副堆栈的特征映射结果合并作为编码器的输出以提高PG-Informer对时间序列的预测鲁棒性。

特征序列 x 的形状变成一个输入矩阵 $X_{\text{feed}} \in \mathbb{R}^{L_x \times d_{\text{model}}}$ 。 X_{feed} 首先进入PSSA块中,由该块中的多头自注意力随机生成3个权值矩阵 W^q 、 W^k 、 W^v ,即PG-Informer需要学习的参数。将 X_{feed} 与这3个权值矩阵分别相乘,分别得到查询矩阵 $Q \in \mathbb{R}^{L_Q \times d}$ 、键矩阵 $K \in \mathbb{R}^{L_K \times d}$ 、值矩阵 $V \in \mathbb{R}^{L_V \times d}$, d 为 X_{feed} 的维度, $d = d_{\text{model}}$,该过程表达如下:

$$Q = [q_1, \dots, q_i, \dots, q_{L_Q}] = W^q \cdot [x_{\text{feed}}^1, x_{\text{feed}}^2, \dots, x_{\text{feed}}^{L_x}] \quad (5)$$

$$\mathbf{K} = [\mathbf{k}_1, \dots, \mathbf{k}_i, \dots, \mathbf{k}_{L_K}] = \mathbf{W}^k \cdot [\mathbf{x}_{\text{feed}}^1, \mathbf{x}_{\text{feed}}^2, \dots, \mathbf{x}_{\text{feed}}^{L_x}] \quad (6)$$

$$\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_i, \dots, \mathbf{v}_{L_V}] = \mathbf{W}^v \cdot [\mathbf{x}_{\text{feed}}^1, \mathbf{x}_{\text{feed}}^2, \dots, \mathbf{x}_{\text{feed}}^{L_x}] \quad (7)$$

接下来, 将第 i 个查询向量 \mathbf{q}_i 的自注意力定义为概率形式的核平滑器, 即:

$$\mathcal{A}(\mathbf{q}_i, \mathbf{K}, \mathbf{V}) = \sum_{u=1}^{L_K} \frac{\text{kernel}(\mathbf{q}_i, \mathbf{k}_u)}{\sum_{l=1}^{L_K} \text{kernel}(\mathbf{q}_i, \mathbf{k}_l)} \mathbf{v}_u = E_{p(\mathbf{k}_u|\mathbf{q}_i)}(\mathbf{v}_u) \quad (8)$$

式中: $\mathcal{A}(\mathbf{q}_i, \mathbf{K}, \mathbf{V})$ 为单个查询向量 \mathbf{q}_i 的自注意力机制表达式; $p(\mathbf{k}_u|\mathbf{q}_i)$ 为键向量 \mathbf{k}_u 相对于查询向量 \mathbf{q}_i 的概率分布, $p(\mathbf{k}_u|\mathbf{q}_i) = \text{kernel}(\mathbf{q}_i, \mathbf{k}_u) / \sum_{l=1}^{L_K} \text{kernel}(\mathbf{q}_i, \mathbf{k}_l)$; $\text{kernel}(\mathbf{q}_i, \mathbf{k}_l) = \exp(\mathbf{q}_i \mathbf{k}_l^T / \sqrt{d})$; $E_{p(\mathbf{k}_u|\mathbf{q}_i)}(\mathbf{v}_u)$ 为值向量 \mathbf{v}_u 关于概率分布 $p(\mathbf{k}_u|\mathbf{q}_i)$ 的期望。

于是, 键向量 \mathbf{k}_u 相对于查询向量 \mathbf{q}_i 的概率分布 $p(\mathbf{k}_u|\mathbf{q}_i)$ 和均匀分布 $q(\mathbf{k}_u|\mathbf{q}_i)$ ($q(\mathbf{k}_u|\mathbf{q}_i) = 1/L_K$, L_K 为键矩阵 \mathbf{K} 的长度) 之间的“相似性”可用来区分权重“重要”的查询向量 \mathbf{q}_i 。这里通过 KL (Kullback-Leibler) 散度 S_{KL} 衡量键向量 \mathbf{k}_u 相对于查询向量 \mathbf{q}_i 的概率分布 (p) 和均匀分布 (q) 之间的“相似性”, 即:

$$S_{\text{KL}}(q||p) = \ln \sum_{l=1}^{L_K} e^{\mathbf{q}_i \mathbf{k}_l^T / \sqrt{d}} - \frac{1}{L_K} \sum_{u=1}^{L_K} \mathbf{q}_i \mathbf{k}_u^T / \sqrt{d} - \ln L_K \quad (9)$$

去掉式 (9) 中的常数项 $\ln L_K$, 则可将第 i 个查询向量 \mathbf{q}_i 的稀疏性度量 $M(\mathbf{q}_i, \mathbf{K})$ 定义为:

$$M(\mathbf{q}_i, \mathbf{K}) = \ln \sum_{u=1}^{L_K} e^{\mathbf{q}_i \mathbf{k}_u^T / \sqrt{d}} - \frac{1}{L_K} \sum_{u=1}^{L_K} \frac{\mathbf{q}_i \mathbf{k}_u^T}{\sqrt{d}} \quad (10)$$

这样, 允许每个键向量 \mathbf{k} 只关注 m 个主要查询向量 \mathbf{q}_i ($i = 1, 2, \dots, m$), m 个主要查询向量 \mathbf{q}_i 的 (单头) 自注意力机制如下:

$$\mathcal{A}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\overline{\mathbf{Q}} \mathbf{K}^T}{\sqrt{d}} \right) \mathbf{V} \quad (11)$$

式中, $\overline{\mathbf{Q}}$ 为一个和查询矩阵 \mathbf{Q} 形状大小相同的稀疏矩阵, 只包含稀疏度量 $M(\mathbf{q}_i, \mathbf{K})$ 下的前 m 个主要查询向量 \mathbf{q}_i , 而 m 由一个恒定的采样因子 c 控制, $m = c \ln L_Q$, L_Q 为查询矩阵 \mathbf{Q} 的长度。

根据式 (10)~(11), 每一次计算查询向量 \mathbf{q}_i 与键向量 \mathbf{k} 的稀疏性度量时, 只需要进行 $O(\ln L_Q)$ 次点积, $O(\cdot)$ 为时间 (空间) 复杂度的符号表示, 并且内存使用量仅为 $O(L_K \ln L_Q)$, 减少了并行计算的时间开销和高内存占用; 此外, 多头的 PSSA 机制可为每个自注意力头生成不同的、稀疏的查询向量 \mathbf{q} 与键向量 \mathbf{k} 之间的稀疏性度量值, 从而避免了滚动轴承性能退

化特征数据严重的信息损失并提取了性能退化特征之间的长期依赖关系, 因而提高了对环境噪声的抗干扰能力和对性能退化特征的非线性逼近能力。

2) SAD 操作

滚动轴承性能退化特征序列 \mathbf{x} 在时间跨度过大的情况下, 会造成 PG-Informer 的网络维度和网络参数量太大; 同时, 作为 PSSA 机制的自然结果, 多头概率稀疏自注意力 (MHPSSA) 块的特征映射是值矩阵 \mathbf{V} 的冗余组合, 该冗余组合包含对滚动轴承性能退化优势特征的干扰信息, 这些干扰信息也会带来网络维度和网络参数量过大的问题, 从而减缓对 PG-Informer 参数的训练速度。为此, 在 PG-Informer 的编码器中采用 SAD 操作, 即对 MHPSSA 块的输出减半来突出主导注意力, 对滚动轴承性能退化特征中的优势特征逐层进行突出, 从而有效处理过长的输入序列, 并生成一个稠密的自注意力特征映射。经过 SAD 操作后, MHPSSA 块的权值矩阵的参数维度将大幅削减 (即图 3 的重叠的红色方块的尺寸将减小)。SAD 操作是从编码堆栈第 n 层 MHPSSA 块向后延伸到编码堆栈的第 $(n+1)$ 层 MHPSSA 块的过程, 该过程表达如下:

$$\chi_{n+1}^T = \text{MaxPool}(\text{ELU}(\text{Conv1d}([\chi_n]_{\text{AB}}))) \quad (12)$$

式中: MaxPool 为最大池化操作; χ_n 为第 n 层 MHPSSA 块的参数矩阵; χ_{n+1} 为 1 维最大池化层的输出, 即第 $n+1$ 层 MHPSSA 块的参数矩阵; $[\cdot]_{\text{AB}}$ 为 MHPSSA 块, 它包含了多头注意和基本操作 (即包括计算值向量 \mathbf{v}_u 关于概率分布 $p(\mathbf{k}_u|\mathbf{q}_i)$ 的期望、运用 KL 散度计算“相似性”来区分重要的查询向量 \mathbf{q}_i 、计算 PSSA 等操作); Conv1d(\cdot) 将 MHPSSA 块的输出输入 1 维卷积滤波器 (带有 ELU(\cdot) 激活函数) 中执行 1 维卷积操作。

再将 1 维卷积后的结果加载到一个最大池化层中从而实现从 χ_n 的降采样, 得到第 $n+1$ 层 MHPSSA 块的参数矩阵 χ_{n+1} 。由于 χ_{n+1} 相对于 χ_n 的参数维度和参数量减半, 因此 SAD 操作减少了 PG-Informer 的参数维度和参数量, 同时将整个内存使用量 (空间复杂度) 减少到 $O((2-\varepsilon)L_x \lg L_x)$, 其中 ε 为一个很小的正值。

此外, PG-Informer 编码器中的单个编码堆栈是接收整个输入序列 (滚动轴承性能退化特征序列) 的主堆栈, 为了增强 SAD 操作的鲁棒性, 还构建了将输入减半的编码主堆栈副本, 它通过逐步减少编码副堆栈的自注意力蒸馏层的数量来对齐编码主堆栈和编码副堆栈的输出维度。

最后, 将主、副堆栈的特征映射结果合并作为 PG-Informer 编码器的输出矩阵 \mathbf{C} , 以减少滚动轴承性能退化特征数据输入的扰动 (包含的噪声) 对预测

性能的影响,提高PG-Informer对时间序列的预测鲁棒性。

1.1.2 PG-Informer的生成式解码器

Transformer等典型的编解码器使用动态多步解码,即当前时刻的解码结果依赖于前一时刻,这导致解码推理过程需要进行多步迭代式的推理,这会降低解码器的推理速度,为此,PG-Informer解码器采用生成式解码的方式,通过一次向前计算,输出所有的预测序列 \mathbf{y} 。

首先,PG-Informer解码器的输入矩阵 $\mathbf{X}_{de} \in \mathbb{R}^{(L_{token} + L_y) \times d_{model}}$ 可推导如下:

$$\mathbf{X}_{de} = \text{Concat}(\mathbf{X}_{token}, \mathbf{X}_0) \quad (13)$$

式中:Concat(\cdot)为合并矢量操作; \mathbf{X}_{token} 为开始标记, $\mathbf{X}_{token} \in \mathbb{R}^{L_{token} \times d_{model}}$,即所有输入矢量 \mathbf{x}_{feed}^1 、 \mathbf{x}_{feed}^2 、 \dots 、 \mathbf{x}_{feed}^p 、 \dots 、 $\mathbf{x}_{feed}^{L_x}$ 中的后 L_{token} 个矢量; \mathbf{X}_0 为预测值序列 \mathbf{y} 的占位符, $\mathbf{X}_0 \in \mathbb{R}^{L_y \times d_{model}}$,初始化为零矩阵。

然后,将输入矩阵 \mathbf{X}_{de} 送入解码器中,解码器由两个解码层构成,每个解码层的内部依次使用遮盖的多头概率稀疏自注意力(MMHPSSA)机制和多头注意力(MHA)机制^[14]来生成解码输出。具体而言,输入矩阵 \mathbf{X}_{de} 由解码器的MHPSSA机制计算得到 $\bar{\mathbf{Q}}$ 、 \mathbf{K} 、 \mathbf{V} 矩阵,再计算 $\bar{\mathbf{Q}}$ 和 \mathbf{K}^T 的乘积 $\bar{\mathbf{Q}}\mathbf{K}^T$,在得到 $\bar{\mathbf{Q}}\mathbf{K}^T$ 之后使用遮盖操作,即矩阵 $\bar{\mathbf{Q}}\mathbf{K}^T$ 与遮盖矩阵 \mathbf{S}_{Mask} 的遮盖部分(元素设为负无穷)点乘,得到被遮盖后的矩阵 \mathbf{M} ,该遮盖操作表达如下:

$$\mathbf{M} = \bar{\mathbf{Q}}\mathbf{K}^T \cdot \mathbf{S}_{Mask} \quad (14)$$

矩阵 \mathbf{M} 再通过MHPSSA机制得到MMHPSSA的输出矩阵 \mathbf{Z}_k ,即:

$$\mathbf{Z}_k = \text{softmax}\left(\frac{\mathbf{M}}{\sqrt{d}}\right)\mathbf{V} \quad (15)$$

式中:softmax(\cdot)为激活函数; k 为下标索引, $1 \leq k \leq h$, h 为MHPSSA机制和MHA机制的头数。

接下来,通过拼接多个输出 \mathbf{Z}_k 得到MMHPSSA的整体输出矩阵 \mathbf{Z} (\mathbf{Z} 与输入矩阵 \mathbf{X}_{de} 的维度一样),该过程表达如下:

$$\mathbf{Z} = \text{Concat}(\mathbf{Z}_1, \dots, \mathbf{Z}_k, \dots, \mathbf{Z}_h) \quad (16)$$

之后,用编码器输出矩阵 \mathbf{C} 和MMHPSSA的整体输出矩阵 \mathbf{Z} 来训练MHA机制得到 \mathbf{Q} 、 \mathbf{K} 、 \mathbf{V} 矩阵(对于第1个解码层,则改用输入矩阵 \mathbf{X}_{de} 代替 \mathbf{Z} 来训练MHA得到 \mathbf{Q} 矩阵),再将得到的 \mathbf{Q} 、 \mathbf{K} 、 \mathbf{V} 矩阵输入已训练好的MHA机制得到MHA的输出矩阵 \mathbf{Z}'_k ,即:

$$\mathbf{Z}'_k = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V} \quad (17)$$

于是,通过合并矢量操作拼接多个输出 \mathbf{Z}'_k 得到解码输出 \mathbf{X}_{de_out} ,即:

$$\mathbf{X}_{de_out} = \text{Concat}(\mathbf{Z}'_1, \dots, \mathbf{Z}'_k, \dots, \mathbf{Z}'_h) \quad (18)$$

最后,将 \mathbf{X}_{de_out} 输入到一个全连接层中,获得最终输出 $\mathbf{y} = \{y_1, y_2, \dots, y_{L_y}\}$,该过程表达如下:

$$\mathbf{y} = \mathbf{A}\mathbf{X}_{de_out} + \mathbf{b} \quad (19)$$

式中, \mathbf{A} 为权重矩阵, \mathbf{b} 为偏置向量。

如式(14)所示,在PG-Informer解码器中使用MMHPSSA机制可以防止输入矩阵 \mathbf{X}_{de} 关注未来的位置信息,即遮住 \mathbf{X}_{de} 的未来位置信息,从而避免了Transformer中出现的自回归现象,保证了PG-Informer训练阶段和推理阶段的一致性;又如式(14)~(19)所示,PG-Informer采用的是生成式一步解码方式,即由输入矢量通过MMHPSSA机制和MHA机制随机生成得到解码输出 \mathbf{X}_{de_out} ,且只需将 \mathbf{X}_{de_out} 输入一个全连接层就能直接一步得到最终输出(即滚动轴承性能退化特征趋势预测值序列 \mathbf{y})。以上生成式一步解码方式将预测的时间复杂度由 $O(L_y)$ 降到 $O(1)$,因而提高了PG-Informer解码推理的速度,进而提高了PG-Informer对滚动轴承性能退化特征趋势序列的预测速度。

1.2 策略梯度学习用于PG-Informer参数训练

设计策略梯度强化学习算法用于训练PG-Informer编码器权值、解码器权值、全连接层的权重和偏置等参数。

设PG-Informer参数集为 $\theta = [\mathbf{W}_1^q, \mathbf{W}_1^k, \mathbf{W}_1^v, \mathbf{W}_2^q, \mathbf{W}_2^k, \mathbf{W}_2^v, \mathbf{A}, \mathbf{b}]$ (\mathbf{W}_1^q 、 \mathbf{W}_1^k 、 \mathbf{W}_1^v 分别为编码器中的 \mathbf{Q} 、 \mathbf{K} 、 \mathbf{V} 矩阵对应的权值矩阵, \mathbf{W}_2^q 、 \mathbf{W}_2^k 、 \mathbf{W}_2^v 分别为生成式解码器中的 \mathbf{Q} 、 \mathbf{K} 、 \mathbf{V} 矩阵对应的权值矩阵),并且定义PG-Informer的损失函数为预测值和真实值的均方误差 \mathcal{L}_θ 。

为更新该参数集 θ ,将PG-Informer作为可与环境交互的智能体,将参数集 θ 作为环境的状态值。在智能体更新参数集 θ 的过程中,需要找到最优策略 ω_θ^* ,使得从初始状态到终止状态的累计回报最大^[15-17]。这里,使用每一个时间步下的平均回报(即在一个确定的时间步长内,智能体处于所有状态的概率与每一种状态下采取所有动作能够得到的立即回报之积)作为累计回报,该累计回报 $J(\theta)$ 表达式如下:

$$J(\theta) = \sum_{t=1}^T \mu_{\omega_\theta}(s) \sum_{a \in \bar{a}} \omega_\theta(a|s, \theta) R_t^a \quad (20)$$

式中: T 为终止时刻; $\mu_{\omega_\theta}(s)$ 为基于策略 ω_θ 生成的马尔可夫链关于状态 s 的分布; \bar{a} 为所有可能动作 a 的集合;

$\omega_\theta(a|s, \theta)$ 为在给定的状态 s 和参数集 θ 下, 采取任何可能动作 a 的概率; R_t^a 为 t 时刻时采取动作 a 得到的立即回报。根据策略梯度定理, 式 (20) 可以转换成一种不涉及状态分布 $\mu_{\omega_\theta}(s)$ 对策略参数 θ 求导的表达形式, 即:

$$\nabla_\theta J(\theta) \propto \sum_{s \in \bar{s}} \mu(s) \sum_{a \in \bar{a}} Q_\omega(s, a) \nabla_\theta \omega_\theta(a|s, \theta) \quad (21)$$

式中, $\nabla_\theta J(\theta)$ 为 $J(\theta)$ 对 θ 的偏微分向量, \bar{s} 为所有可能状态 s 的集合, $\mu(s)$ 为策略 ω 下稳定状态 s 的分布, $Q_\omega(s, a)$ 为状态 s 在策略 ω 下采取动作 a 的价值, $\nabla_\theta \omega_\theta(a|s, \theta)$ 为 $\omega_\theta(a|s, \theta)$ 对 θ 的偏微分列向量, \propto 为“正比于”。

在遵循策略 ω_θ 而产生完整轨迹 $(s_0, a_0, r_1, \dots, s_{T-1}, a_{T-1}, r_T, \dots, s_{T-1}, a_{T-1}, r_T)$ 的情况下 (r_t 为 t 时刻智能体反馈的回报), 目标函数 $J(\theta)$ 的梯度 $\nabla J(\theta)$ 相对于式 (21) 右边的比例常数 (轨迹长度) 在更新参数 θ 时可以被步长 β 吸收, 因此式 (21) 可进一步推导如下:

$$\begin{aligned} \nabla_\theta J(\theta) &\propto \sum_{s \in \bar{s}} \mu_{\omega_\theta}(s) \sum_{a \in \bar{a}} Q_\omega(s, a) \nabla_\theta \omega_\theta(a|s, \theta) = \\ &\sum_{s \in \bar{s}} \mu_{\omega_\theta}(s) \sum_{a \in \bar{a}} \omega_\theta(a|s, \theta) Q_\omega(s, a) \frac{\nabla_\theta \omega_\theta(a|s, \theta)}{\omega_\theta(a|s, \theta)} = \\ &\sum_{s \in \bar{s}} \mu_{\omega_\theta}(s) \sum_{a \in \bar{a}} \nabla_\theta \ln \omega_\theta(a|s, \theta) Q_\omega(s, a) \end{aligned} \quad (22)$$

为了在不改变策略梯度 $\nabla_\theta J(\theta)$ 的情况下进一步减小策略梯度的方差, 需要从策略梯度 $\nabla_\theta J(\theta)$ 里减去一个基线函数 $B(s)$ 。为了保证基线函数仅与状态 s 有关, 而与动作 a 无关, 设置 $B(s) = \mathcal{V}_{\omega_\theta}(s)$, $\mathcal{V}_{\omega_\theta}(s)$ 为策略 ω_θ 下当前状态 s 的值, 则 $\nabla_\theta J(\theta)$ 可以进一步表达为:

$$\begin{aligned} \nabla_\theta J(\theta) &= \sum_{s \in \bar{s}} \mu_{\omega_\theta}(s) \sum_{a \in \bar{a}} (Q_\omega(s, a) - \mathcal{V}_{\omega_\theta}(s)) \nabla_\theta \omega_\theta(a|s, \theta) = \\ &E_{\omega_\theta}[\nabla_\theta \ln \omega_\theta(a|s) (Q_\omega(s, a) - \mathcal{V}_{\omega_\theta}(s))] \end{aligned} \quad (23)$$

式中, $E_{\omega_\theta}[\cdot]$ 为 $\ln \omega_\theta(a|s)$ 对 θ 的导数与 $Q_\omega(s, a)$ 、 $\mathcal{V}_{\omega_\theta}(s)$ 之差的乘积关于 ω_θ 的期望。为了简化式 (23), 将 $Q_\omega(s, a) - \mathcal{V}_{\omega_\theta}(s)$ 定义为优势函数 $A_{\omega_\theta}(s, a)$, 则式 (23) 可以改写成如下形式:

$$\nabla_\theta J(\theta) = E_{\omega_\theta}[\nabla_\theta \ln \omega_\theta(a|s) A_{\omega_\theta}(s, a)] \quad (24)$$

一般用时序差分 (TD) 误差 δ_{ω_θ} 代替优势函数 $A_{\omega_\theta}(s, a)$ 来计算策略梯度, 因为 TD 误差 δ_{ω_θ} 是优势函数 $A_{\omega_\theta}(s, a)$ 的无偏估计。TD 误差 δ_{ω_θ} 定义如下:

$$\delta_{\omega_\theta} = r + \gamma \mathcal{V}_{\omega_\theta}(s_-) - \mathcal{V}_{\omega_\theta}(s) \quad (25)$$

式中, r 为环境反馈的回报, γ 为折扣系数, s_- 为 s 的后继状态。于是, 策略梯度 $\nabla_\theta J(\theta)$ 计算公式变为:

$$\nabla_\theta J(\theta) = E_{\omega_\theta}[\nabla_\theta \ln \omega_\theta(a|s) \delta_{\omega_\theta}] \quad (26)$$

因此, 参数集 θ 更新过程可表示为:

$$\theta_{t+1} = \theta_t + \beta E_{\omega_\theta}[\nabla_\theta \ln \omega_\theta(a|s) \delta_{\omega_\theta}] \quad (27)$$

也可以将该策略梯度 $\nabla_\theta J(\theta)$ 实例化为对均方损失 (MSE) 函数相反数的梯度, 于是参数集 θ 更新过程可表示为:

$$\theta_{t+1} = \theta_t - \beta \nabla_\theta J(\theta_t) = \theta_t + \beta (y(\theta_t) - \hat{y}) \nabla_{\theta, y}(\theta_t) \quad (28)$$

式中, $\nabla_{\theta, y}(\theta_t)$ 为 $y(\theta_t)$ 对 θ_t 的偏微分向量。

比较式 (27) 和 (28), 为使该两式一致, 令 $r = 0$, $\mathcal{V}_{\omega_\theta}(s_-) = y_-(\theta_t) |\phi_t|$, $\mathcal{V}_{\omega_\theta}(s) = y(\theta_t) |\phi_t|$, $\ln \omega_\theta(a|s) = y(\theta_t)$, ϕ_t 为状态价值函数 $\mathcal{V}_{\omega_\theta}(s)$ 在 t 时刻的权重, $y_-(\theta_t)$ 为智能体与环境互动后预测值 $y(\theta_t)$ 的后继更新。于是式 (27) 可进一步推导为:

$$\theta_{t+1} = \theta_t + \beta E_{\omega_\theta}[(\gamma y_-(\theta_t) - y(\theta_t)) |\phi_t| \nabla_{\theta, y}(\theta_t)] \quad (29)$$

式 (29) 完成了一次 PG-Informer 对参数集 θ 的更新过程, 重复该更新过程直至 MSE 小于训练阈值或训练步数达到最大训练步数, 此时参数集 θ 不再更新, 即策略梯度学习算法被视为训练至收敛。由式 (30) 可知, 策略梯度学习算法能够追踪参数集 θ 的历史更新信息并快速找到 θ 更新的最优策略 ω_θ^* , 从而加快了 PG-Informer 参数的收敛, 即提高了对 PG-Informer 参数集 θ 的训练速度。

2 基于 PG-Informer 模型的滚动轴承剩余寿命预测方法

2.1 方法描述

所提出的基于 PG-Informer 的滚动轴承 RUL 预测方法具体实现过程如下:

1) 首先对滚动轴承的原始振动加速度数据进行小波阈值降噪, 然后从降噪数据中提取最大值、标准差、方根幅值、均方根值、峰峰值、偏斜度、峭度、峰值指标、裕度指标、波形指标、脉冲指标、频域幅值平均值、频域幅值方差、频域幅值偏斜度指标、频域幅值峭度指标、重心频率、标准偏差频率、均方根频率、均方频率、峰值频率、平方根比率、频域频率偏斜度和频域频率峭度共计 23 个时、频域特征^[18], 并将该 23 个时、频域特征构造 (1 维特征向量) 模糊熵^[19] 作为滚动轴承性能退化特征。在正常状态下, 滚动轴承的振动时域信号近似为高斯分布, 信号频率成分比较简单, 此时信号的模糊熵很小; 随着滚动轴承最细微的性能退化的出现, 信号中出现了微弱的故障频率, 此时信号能量分布、频率成分开始分散, 模糊熵开始随之增加; 当明显的性能退化出现时, 噪声信号频率成分所占比例急剧增加, 能量分散加剧, 从而导致模糊熵显著增大。因此, 模糊熵能定量描述滚动轴承振动时、频域特征随时间的变化, 能较好反映滚动轴承的性

能退化特征的变化。

2)将得到的模糊熵序列输入到PG-Informer中用于训练PG-Informer的网络参数。

3)利用训练好的PG-Informer生成一步解码预测滚动轴承以后的性能退化特征趋势。

4)结合训练用的性能退化特征序列和预测得到的性能退化特征趋势值,建立3参数威布尔分布可靠度模型,从而预测出滚动轴承的失效阈值时间点和RUL。

2.2 PG-Informer的训练过程

PG-Informer的训练过程如下:假设输入的滚动轴承性能退化特征(即模糊熵)序列为 $\mathbf{x} = [x_1, x_2, \dots, x_{\tilde{n}}]$ (\tilde{n} 为用于训练的模糊熵序列的长度),将该时间序列重构为训练输入集 \mathbf{X} 和训练目标集 $\hat{\mathbf{y}}$,表示如下:

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & \cdots & x_{L_x} \\ x_2 & x_3 & \cdots & x_{L_x+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\tilde{n}-L_x} & x_{\tilde{n}-L_x+1} & \cdots & x_{\tilde{n}-1} \end{bmatrix}, \hat{\mathbf{y}} = \begin{bmatrix} x_{L_x+1} \\ x_{L_x+2} \\ \vdots \\ x_{\tilde{n}} \end{bmatrix}.$$

将训练输入集 \mathbf{X} 的每一行作为一个 L_x 维训练样本输入,该训练样本输入的训练目标为训练目标集 $\hat{\mathbf{y}}$ 中对应行上的元素,故共有 $(\tilde{n} - L_x)$ 个训练样本。

因此,PG-Informer的训练过程可归纳如下:1)初始化参数集 θ ,即随机初始化PG-Informer的编码器、解码器中的权值和全连接层的权重,并将全连接层的偏置初始化为0。2)将训练样本集 \mathbf{X} 输入PG-Informer得到性能退化特征趋势预测值 \mathbf{y} 。3)使用第1.2节的策略梯度学习算法对PG-Informer的参数集 θ 进行更新,将更新后的参数集 θ 作为下一次训练的初始值。4)重复步骤2)和3)的更新过程,直至关于训练目标集 $\hat{\mathbf{y}}$ 和性能退化特征趋势预测值 \mathbf{y} 的均方误差损失小于设定的训练阈值或训练步数达到最大训练步数,就完成了对PG-Informer的训练。

2.3 PG-Informer的预测过程

将样本 $\tilde{\mathbf{x}} = [x_{\tilde{n}-L_x+1}, x_{\tilde{n}-L_x+2}, \dots, x_{\tilde{n}}]$ 输入训练好的PG-Informer,由PG-Informer的生成式解码器的一步解码方式就可一次性输出 L_y 个滚动轴承性能退化特征趋势预测值,即 $\mathbf{y} = [y_{\tilde{n}+1}, y_{\tilde{n}+2}, \dots, y_{\tilde{n}+L_y}]$ 。

2.4 剩余寿命预测

结合训练用的性能退化特征(模糊熵)序列 \mathbf{x} 和预测得到的性能退化特征趋势值 $\mathbf{y} = [y_{\tilde{n}+1}, y_{\tilde{n}+2}, \dots, y_{\tilde{n}+L_y}]$,建立用于滚动轴承RUL预测的可靠度模型。

首先,滚动轴承的基本故障率函数 $\lambda(t, s)$ 可视为服从3参数威布尔分布^[20],故该故障率函数可以计算如下:

$$\lambda(t, s) = \left(\frac{\eta}{\alpha}\right) \left(\frac{t}{\alpha}\right)^{\eta-1} \exp(\tau s) \quad (30)$$

式中, s 为性能退化特征集合, $s = [x_1, x_2, \dots, x_{\tilde{n}}, y_{\tilde{n}+1}, y_{\tilde{n}+2}, \dots, y_{\tilde{n}+L_y}]$, α 为尺度参数, η 为形状参数, τ 表示协变量回归参数。

然后,由于3参数威布尔分布可靠度函数 $R(t, s)$ 与故障率函数 $\lambda(t, s)$ 满足 $\lambda(t, s) = -\frac{d}{dt} \ln R(t, s)$,于是结合式(30), t 时刻的可靠度可表示为:

$$R(t, s) = \exp\left(-\int \lambda(t, s) dt\right) = \exp\left(-\left(\frac{t}{\alpha}\right)^{\eta} \exp(\tau s)\right) \quad (31)$$

至此,完成3参数威布尔分布可靠度模型的建立。假设起始预测时刻为 t_s ,预测得到的滚动轴承失效阈值对应的时刻(预测的可靠度 R_{fp} 为10%所对应的时刻^[21])为 t_{fp} ,则所预测的滚动轴承RUL(记为 T_p)为:

$$T_p = t_{fp} - t_s + 1 \times \Delta T = R^{-1}(R_{fp}) - N_s \times \Delta T + 1 \times \Delta T = (N_{fp} - N_s + 1) \times \Delta T \quad (32)$$

式中, ΔT 为性能退化特征曲线中相邻两点之间的时间间隔, R_{fp} 为所预测的失效阈值点对应的可靠度值, N_s 为起始预测时间点, N_{fp} 为所预测的失效阈值时间点。

3 实例分析

3.1 实验平台

使用美国辛辛那提大学智能维护系统中心的实验平台得到的滚动轴承全寿命周期数据^[22]来验证基于PG-Informer的滚动轴承RUL预测方法的有效性。该轴承实验台和传感器布置如图4所示。实验过程如下:交流电机以2 000 r/min的恒定转速旋转,并通过摩擦带连接到轴上,转轴上安装了美国威斯康星州Rexnord公司生产的4个ZA-2115双列滚子轴承,并通过弹簧机构向这4个轴承施加26 690 N的径向载荷。4个轴承都由润滑油循环系统强制润滑,安装在润滑油反馈管中的磁性塞收集润滑油中的碎屑,作为轴承退化的证据。当附着在磁性塞上的累积碎屑超过一定水平并导致电气开关闭合时,就认为轴承已失效,测试将停止。将高灵敏度石英压电加速度计PCB 353B33安装在4个轴承座上用于测量这4个轴承的振动加速度数据;使用美国国家仪器公司的DAQ 6062E

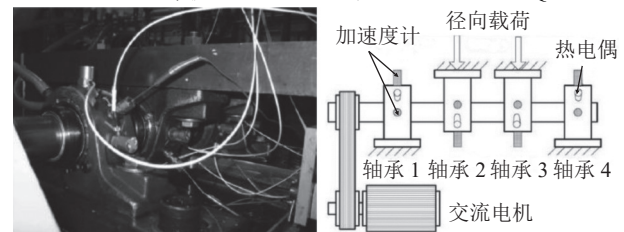


图4 轴承试验台及传感器放置实物图和示意图

Fig. 4 Physical picture and schematic diagram of bearing test bench and sensor placement

数据采集卡进行振动加速度数据采集,采样频率为 20 kHz, 每间隔 10 min 采集 1 次数据, 1 次采集的采样长度为 20 480 个点。

3.2 滚动轴承剩余寿命预测

使用滚动轴承全寿命周期数据集的子数据集^[23]中的 1 号滚动轴承运行数据作为实验数据。1 号滚动轴承连续运行 9 840 min (约 7 d) 后发生外圈失效, 利用采集到的该全寿命期间振动加速度数据提取性能退化特征 (模糊熵), 图 5 为 1 号滚动轴承全寿命周期内的模糊熵曲线。由图 5 可知: 从起始点 1 到 702 点处, 模糊熵保持稳定, 表明该滚动轴承处于正常运行阶段; 从第 703 点到第 926 点, 模糊熵逐渐增大, 表明该轴承处于外圈初始退化阶段, 即早期故障阶段; 从 927 点 (利用第 2.4 节中的方法, 按可靠度 $R_{fp} = 10\%$ 求出的失效阈值点 $N_f = 927$ ^[21]) 到最后一个点 (第 984 点), 模糊熵曲线快速上升, 表明该轴承的外圈缺陷迅速扩展, 最终失效。

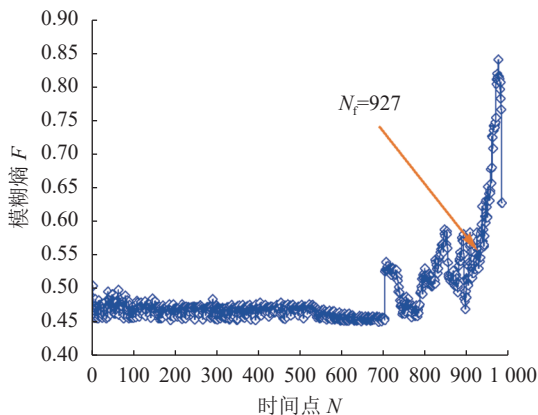


图 5 1 号滚动轴承全寿命周期内的模糊熵曲线

Fig. 5 Fuzzy entropy curve of No. 1 rolling bearing in its life cycle

对于滚动轴承而言, 初始退化阶段及其随后的恶性退化阶段是性能退化趋势观测和 RUL 预测中最关键的时间区间, 因此, 将第 725 到第 824 点的 (100 个) 模糊熵作为训练样本, 将第 N_s ($N_s = 825$) 点及其后续数据 (即其余 L_y ($L_y = 160$) 个模糊熵) 作为测试样本。对 PG-Informer 中的参数设置如下: 训练样本的维度 $L_x = 50$; 在输入表示中, 1 维卷积滤波器的卷积核宽度为 3, 步幅为 1, 输入表示的特征维数 $d_{\text{model}} = 16$, 权重大小平衡因素 $a = 1$; 在 PG-Informer 的网络结构中, MHPSSA 机制和 MHA 机制的头数 h 均为 8, 1 维卷积滤波器的卷积核宽度为 3, 1 维最大池化层的步幅为 2, 填充像素为 2; 在策略梯度学习算法中, 折扣系数 $\gamma = 0.9$, 学习率 $\alpha' = 0.001$, 最大训练步数 $n'_{\text{max}} = 700$, MSE 训练阈值 $S_{\text{MSE}_{\text{min}}} = 5 \times 10^{-4}$ 。完成 PG-Informer 的参数设置后, 就可依据第 2.2 节和第 2.3 节的 PG-

Informer 训练和预测过程得到性能退化特征趋势预测结果, 如图 6 所示。显然, 在被预测的时间点区间 [825, 984] 内, PG-Informer 预测得到的性能退化特征趋势值与实际值非常接近, 两者变化趋势较为一致。

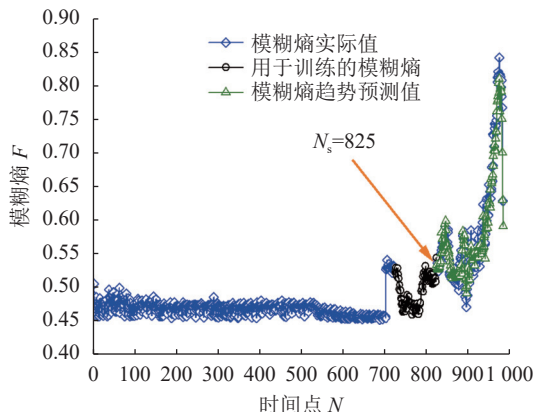


图 6 用于训练的模糊熵和 PG-Informer 预测得到的模糊熵
Fig. 6 Fuzzy entropies for training PG-Informer and those predicted by PG-Informer

然后, 将作为训练样本的模糊熵序列和 PG-Informer 预测得到的模糊熵趋势序列输入第 2.4 节所示的 3 参数威布尔分布可靠度模型, 可得到 1 号滚动轴承的可靠度预测曲线, 如图 7 所示, 其中, 可靠度为 10% 所对应的的时间点, 即失效阈值时间点预测值为 $N_{fp} = 920.3$ 。

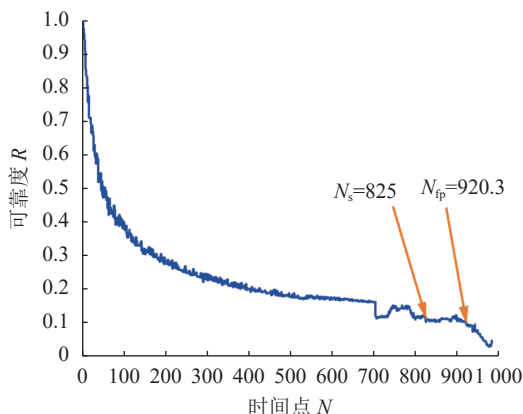


图 7 由 3 参数威布尔分布可靠度模型得到的可靠度
Fig. 7 Reliabilities obtained from three-parameter exponential Weibull distribution reliability model

于是, 所要预测的 1 号滚动轴承的 RUL 就是预测的失效阈值时间点 $N_{fp} = 920.3$ 和起始预测时间点 $N_s = 825$ 之间的时间差值 T_P ($T_P = (N_{fp} - N_s + 1) \times \Delta T = 963 \text{ min}$); 同时, 实际的 RUL 值为 T_{Pa} ($T_{Pa} = (N_f - N_s + 1) \times \Delta T = 1 030 \text{ min}$), 其中, ΔT ($\Delta T = 10 \text{ min}$) 为图 8 模糊熵曲线中相邻两点的的时间间隔。因此, PG-Informer 模型的预测误差 δ_P ($\delta_P = \frac{T_{Pa} - T_P}{T_{Pa}} \times 100\%$) 为 6.50%。

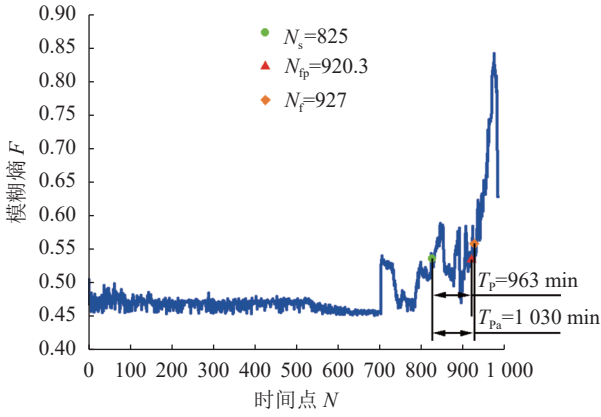


图8 基于PG-Informer的方法预测的RUL和实际的RUL
Fig. 8 RUL predicted by PG-Informer-based method and actual RUL

3.3 对比分析

3.3.1 训练速度

为了在相同条件下比较PG-Informer与LSTM、GRU、ConvLSTM、Transformer这5种机器学习方法的训练速度,后4种方法的性能退化特征也用模糊熵表示,同时,将后4种方法的最大训练步数和MSE训练阈值分别设置为与PG-Informer相同的值,将LSTM、GRU、ConvLSTM和Transformer的学习率 η' ^[24]都设为0.001。图9为5种机器学习方法的学习曲线。由图9可以看出,PG-Informer的参数在训练步数达到第394步时训练至收敛,而LSTM、GRU、ConvLSTM和Transformer分别在第700、650、236和600步训练至收敛,故PG-Informer相较LSTM、GRU、Transformer这3种方法具有更快的训练速度,仅比ConvLSTM的训练速度略慢。

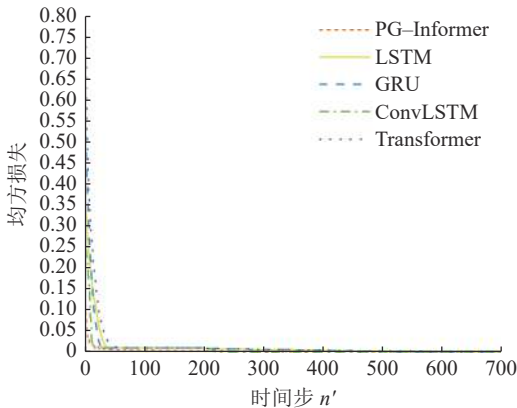


图9 5种机器学习方法的学习曲线

Fig. 9 Learning curves of five machine learning methods

3.3.2 非线性逼近能力

为定量评价PG-Informer、LSTM、GRU、ConvLSTM和Transformer这5种机器学习方法的非线性逼近能力,以均方误差MSE(记为 S_{MSE})、平均绝对误差MAE(记为 S_{MAE})、平均绝对百分误差MAPE(记为

S_{MAPE})和拟合度 R^2 作为非线性逼近能力评价指标,即:

$$S_{MSE} = \frac{1}{L_y} \sum_{i=1}^{L_y} (\hat{y}_i - y_i)^2 \quad (33)$$

$$S_{MAE} = \frac{1}{L_y} \sum_{i=1}^{L_y} |\hat{y}_i - y_i| \quad (34)$$

$$S_{MAPE} = \frac{1}{L_y} \sum_{i=1}^{L_y} \left| \frac{\hat{y}_i - y_i}{\hat{y}_i} \right| \quad (35)$$

$$R^2 = 1 - \frac{S_{MSE}}{S_{VAR}} = 1 - \frac{\left(\sum_{i=1}^{L_y} (\hat{y}_i - y_i)^2 \right) / L_y}{\left(\sum_{i=1}^{L_y} (\hat{y}_i - \hat{y}_{AVE})^2 \right) / L_y} \quad (36)$$

式(33)~(36)中, y_i 和 \hat{y}_i 分别为被预测的模糊熵趋势值及其对应的实际值, \hat{y}_{AVE} 和 S_{VAR} 分别是 L_y 个实际模糊熵的均值与方差。为减少随机偶然性误差,对以上5种机器学习方法的非线性逼近能力评价指标分别进行了统计分析,得到在50次重复RUL预测中的非线性逼近能力评价指标平均值,即 \bar{S}_{MSE} 、 \bar{S}_{MAE} 、 \bar{S}_{MAPE} 和 \bar{R}^2 ,对比结果见表1。

表1 5种机器学习方法的非线性逼近能力对比

Tab. 1 Comparison of nonlinear approximation abilities of five machine learning methods

方法	\bar{S}_{MSE}	\bar{S}_{MAE}	$\bar{S}_{MAPE}/\%$	\bar{R}^2
PG-Informer	0.000 73	0.018 9	3.18	0.906 8
LSTM	0.001 79	0.030 4	4.66	0.771 7
GRU	0.001 51	0.027 6	4.22	0.807 4
ConvLSTM	0.002 46	0.036 7	5.63	0.803 1
Transformer	0.001 91	0.041 7	7.11	0.756 9

由表1可知,相比于其他4种方法,PG-Informer的 \bar{S}_{MSE} 、 \bar{S}_{MAE} 和 \bar{S}_{MAPE} 值最小,PG-Informer的 \bar{R}^2 值最接近1,这表明PG-Informer在多次重复实验中非线性逼近能力更好。

3.3.3 时间空间复杂度

表2为PG-Informer与LSTM、GRU、ConvLSTM和Transformer用于1号滚动轴承RUL预测的时间和空间复杂度的比较结果。由表2可知,PG-Informer的时间和空间复杂度虽然略大于LSTM、GRU和ConvLSTM,但比Transformer的明显要小。

3.3.4 参数维度和参数量

表3为PG-Informer与LSTM、GRU、ConvLSTM和Transformer的参数维度和参数量。由表3可知,PG-Informer的参数维度和参数量虽然多于LSTM、GRU和ConvLSTM,但比Transformer明显少得多。

表 2 5种机器学习方法的时间和空间复杂度对比

Tab. 2 Comparison of time and space complexities of five machine learning methods

方法	时间复杂度	空间复杂度
PG-Informer	$O(L_x \lg L_x) = O(84.95)$	$O(L_x \lg L_x) = O(84.95)$
LSTM	$O(L_x) = O(50)$	$O(L_x) = O(50)$
GRU	$O(L_x) = O(50)$	$O(L_x) = O(50)$
ConvLSTM	$O(L_x) = O(50)$	$O(L_x) = O(50)$
Transformer	$O(L_x^2) = O(2\ 500)$	$O(L_x^2) = O(2\ 500)$

表 3 5种机器学习方法的参数维度和参数量对比

Tab. 3 Comparison of parameter dimension and parameter number of five machine learning methods

方法	参数维度	参数量
PG-Informer	W_1^q 的维度为[16, 16]; W_1^k 的维度为[16, 16]; W_1^v 的维度为[16, 16]; W_2^q 的维度为[16, 16]; W_2^k 的维度为[16, 16]; W_2^v 的维度为[16, 16]; A 的维度为[1, 16]; b 的维度为[1]	12 561
LSTM	LSTM层的输出矩阵维度为 $[N', 4]$ (N' 为输入数据长度); 压缩(Dense)层的输出矩阵维度 $[N', 1]$	101
GRU	GRU层的输出矩阵维度为 $[N', 4]$; Dense层的输出矩阵维度 $[N', 1]$	89
ConvLSTM	1维卷积层的输出矩阵维度为[1, 1, 1, 64]; 1维最大池化层的输出矩阵维度为[1, 1, 1, 64]; 展平层的输出矩阵维度为[1, 1, 64]; LSTM层的输出矩阵维度为[1, 4]; Dense层的输出矩阵维度[1, 1]	1 237
Transformer	编码器多头注意力(MHA)层输出矩阵维度为[96, 32]; 编码器归一化层输出矩阵维度为[32]; 编码器前馈层输出矩阵维度为[32, 128]; 解码器遮盖多头注意力(MMHA)层输出矩阵维度为[96, 32]; 解码器归一化层输出矩阵维度为[32]; 解码器MHA层输出矩阵维度为[32, 32]; 解码器前馈层输出矩阵维度为[32, 128]	107 680

3.3.5 预测精度

为保证条件相同, LSTM、GRU、ConvLSTM和Transformer的RUL预测方法采用与PG-Informer相同的可靠度模型, 即第2.4节的3参数威布尔分布可靠度模型。图10~13分别为LSTM、GRU、ConvLSTM和Transformer 4种方法分别预测得到的1号滚动轴承的RUL。

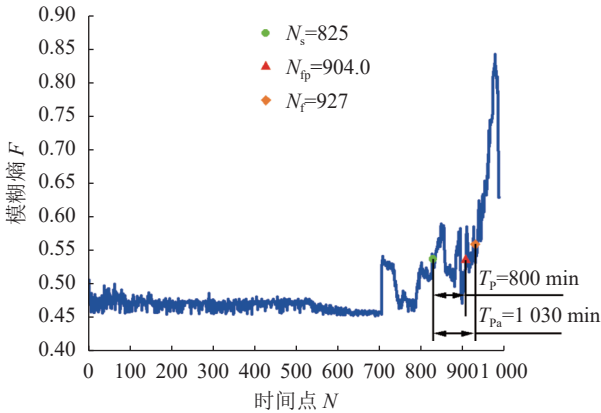


图 10 基于LSTM的方法预测的RUL和实际的RUL

Fig. 10 RUL predicted by LSTM-based method and actual RUL

由图8、10、11、12和13可知, 基于PG-Informer的RUL预测方法和其他4种方法预测的RUL预测值 T_p 分别为963 min、800 min、910 min、930 min和950 min, 预测误差 δ_p 分别为6.50%、22.33%、11.65%、9.71%和7.77%。显然, 基于PG-Informer的方法的RUL预测误差最小, 这表明基于PG-Informer的方法具有比其他4种方法更高的RUL预测精度。

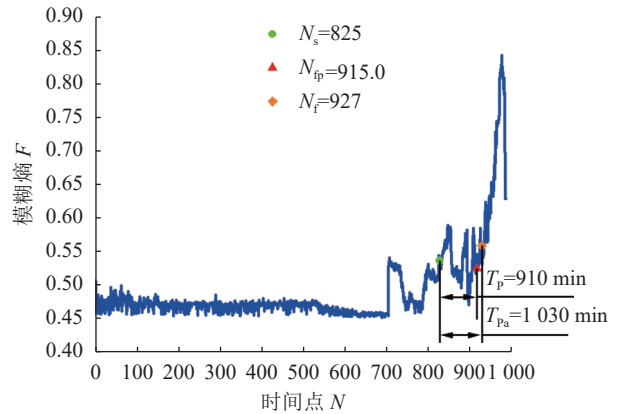


图 11 基于GRU的方法预测的RUL和实际的RUL

Fig. 11 RUL predicted by GRU-based method and actual RUL

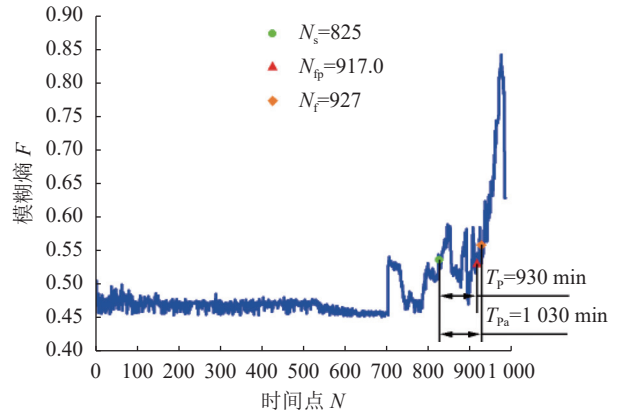


图 12 基于ConvLSTM的方法预测的RUL和实际的RUL

Fig. 12 RUL predicted by ConvLSTM-based method and actual RUL

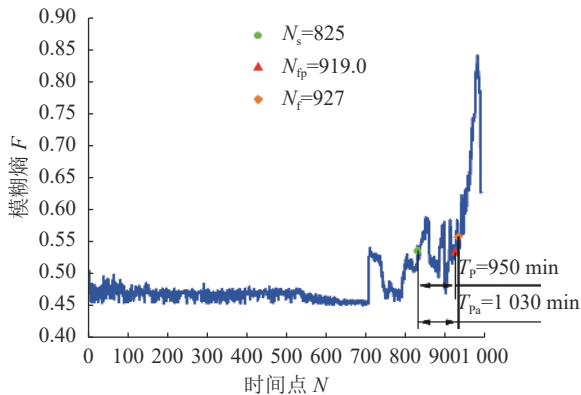


图 13 基于Transformer的方法预测的RUL和实际的RUL
Fig. 13 RUL predicted by Transformer-based method and actual RUL

3.3.6 鲁棒性

图 14~17 为 PG-Informer、LSTM、GRU、ConvLSTM 和 Transformer 预测方法执行 50 次的 RUL 预测值, 模糊熵趋势预测值和模糊熵真实值的均方误差 MSE、平均绝对误差 MAE 及平均绝对百分误差 MAPE。

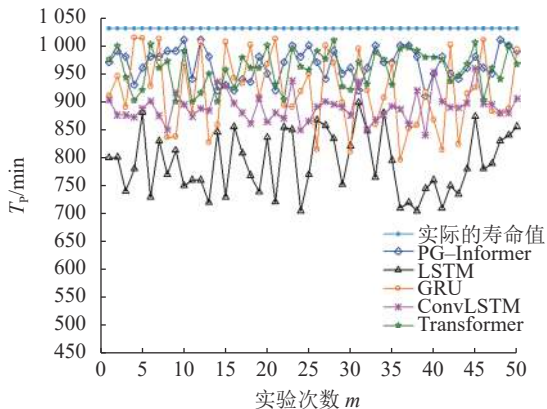


图 14 5种RUL预测方法执行50次预测的RUL预测值 T_p
Fig. 14 Predicted RUL values T_p obtained from 50 RUL predictions performed by five RUL prediction methods

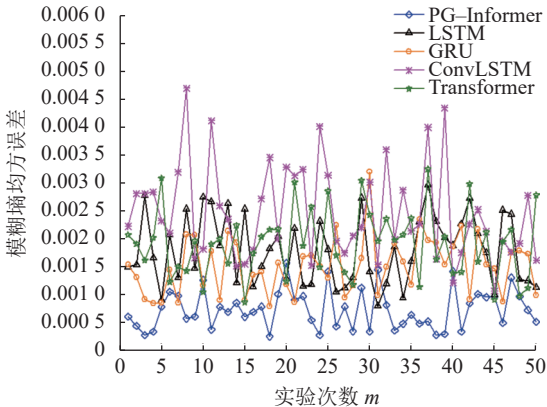


图 15 5种RUL预测方法执行50次寿命预测得到的模糊熵均方误差
Fig. 15 Mean square errors of fuzzy entropy obtained from 50 RUL predictions performed by five RUL prediction methods

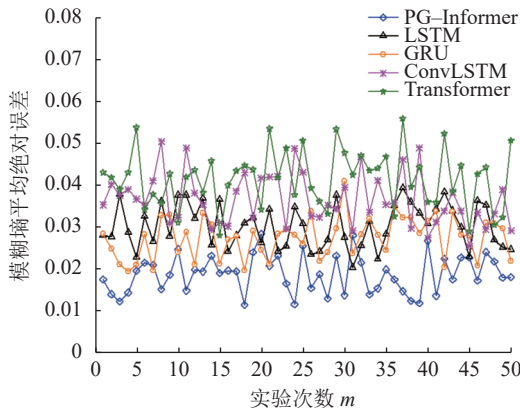


图 16 5种RUL预测方法执行50次寿命预测得到的模糊熵平均绝对误差

Fig. 16 Mean absolute errors of fuzzy entropy obtained from 50 RUL predictions performed by five RUL prediction methods

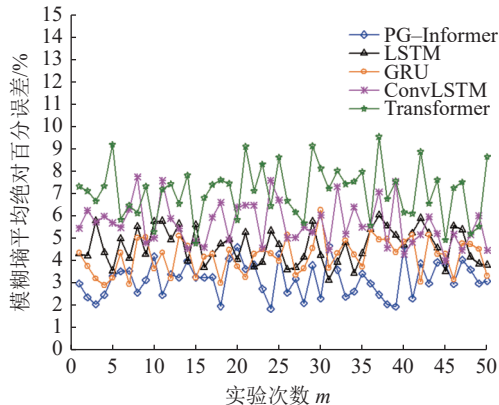


图 17 5种RUL预测方法执行50次寿命预测得到的模糊熵平均绝对百分误差

Fig. 17 Mean absolute percentage errors of fuzzy entropy obtained from 50 RUL predictions performed by five RUL prediction methods

由图 14~17 可知, 基于 PG-Informer 的预测方法相较于其他 4 种方法, 其 RUL 预测值 T_p 最接近 RUL 真实值, 且其模糊熵趋势预测结果的 MSE、MAE 和 MAPE 都最小。换言之, 在 50 次重复 RUL 预测实验中, 基于 PG-Informer 的预测方法表现出比其他 4 种方法更好的鲁棒性。

3.3.7 预测时间

采用编程软件为 Pycharm 2 022.1, RAM 内存为 8G, CPU 采用 3.6GHz Intel, 记录每种方法执行 50 次 RUL 预测的平均预测时间 (即提取模糊熵时间、PG-Informer 训练时间、模糊熵趋势预测时间和 RUL 计算时间之和)。图 18 为 5 种方法平均预测时间的比较。图 18 中, 基于 PG-Informer 的方法的预测时间为 132.37 s, 而基于 LSTM、GRU、ConvLSTM 和 Transformer 的方法的预测时间分别为 34.19、28.87、22.94 和 208.91 s。显然, 基于 PG-Informer 的方法的预测时间

虽然长于基于LSTM、GRU和ConvLSTM的方法的预测时间,但比基于Transformer的方法的预测时间明显要短,因而其具有较高的计算效率。

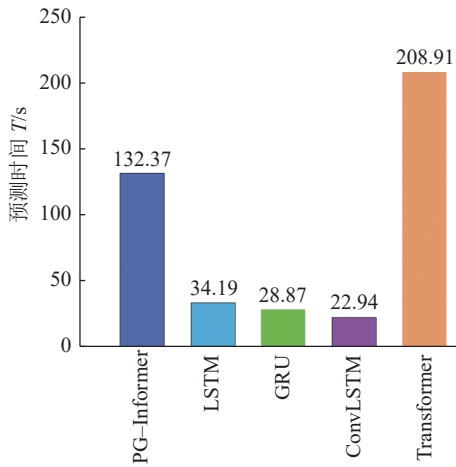


图 18 5种RUL预测方法的RUL预测时间对比

Fig. 18 Comparison of RUL prediction time of five RUL prediction methods

4 结 论

本文提出一种新型编解码器——策略梯度Informer(PG-Informer)模型,并将其应用于滚动轴承RUL预测。主要结论如下:

1)在PG-Informer理论模型中,所设计的PSSA机制能够提高PG-Informer的非线性逼近能力,减少其时间和空间复杂度。

2)在PG-Informer理论模型中采用SAD操作,减少了PG-Informer的参数维度和参数量,提高了PG-Informer对时间序列的预测鲁棒性。

3)PG-Informer中的生成式解码器只需对解码输入进行一步解码即可输出预测结果,无需动态多步解码,提升了PG-Informer对时间序列的预测速度。

4)引入策略梯度学习算法训练PG-Informer的参数,提高了对PG-Informer参数的训练速度。

5)ZA-2115双列滚子轴承RUL预测实例表明:所提出的方法预测得到的RUL值相较于基于LSTM、GRU、ConvLSTM和Transformer的RUL预测方法精度最高、预测误差最小、鲁棒性最好;所提出的方法的时间和空间复杂度、参数维度和参数量相较于基于Transformer的预测方法都更小;所提出方法的RUL预测时间比基于Transformer的预测方法更短。

然而,与LSTM、GRU和ConvLSTM等方法相比,所提出的PG-Informer模型的时间和空间复杂度、参数维度和参数量仍然偏大,因而造成基于PG-Informer模型的滚动轴承RUL预测方法的RUL预测时间相较于基于LSTM、GRU和ConvLSTM的RUL预测方法

仍然偏长。为了弥补这一不足,今后的研究重点将放在进一步降低PG-Informer的时间和空间复杂度以及减少PG-Informer的参数维度和参数量这两方面,进而缩短基于PG-Informer模型的RUL预测方法的预测时间。

参考文献:

- [1] Jiang Guoqian,Zhou Wenda,Chen Qi,et al.Dual residual attention network for remaining useful life prediction of bearings[J].*Measurement*,2022,199:111424.
- [2] Wei Yupeng,Wu Dazhong,Terpenny J.Bearing remaining useful life prediction using self-adaptive graph convolutional networks with self-attention mechanism[J].*Mechanical Systems and Signal Processing*,2023,188:110010.
- [3] Wang Han,Wang Dongdong,Liu Haoxiang,et al.A predictive sliding local outlier correction method with adaptive state change rate determining for bearing remaining useful life estimation[J].*Reliability Engineering & System Safety*, 2022,225:108601.
- [4] Xu Guanji,Hou Dongming,Qi Hongyuan,et al.High-speed train wheel set bearing fault diagnosis and prognostics:A new prognostic model based on extendable useful life[J]. *Mechanical Systems and Signal Processing*,2021,146: 107050.
- [5] Zhu Jun,Chen Nan,Peng Weiwen.Estimation of bearing remaining useful life based on multiscale convolutional neural network[J].*IEEE Transactions on Industrial Electronics*, 2019,66(4):3208–3216.
- [6] Ding Yifei,Jia Minping,Zhuang Jichao,et al.Deep imbalanced regression using cost-sensitive learning and deep feature transfer for bearing remaining useful life estimation[J]. *Applied Soft Computing*,2022,127:109271.
- [7] Wang Jiujian,Yang Shaopu,Liu Yongqiang,et al.A method of bearing remaining useful life estimation based on convolutional long short-term memory neural network[J].*Journal of Mechanical Engineering*,2021,57(21):88–95.[王久健,杨绍普,刘永强等.一种基于空间卷积长短时记忆神经网络的轴承剩余寿命预测方法[J].*机械工程学报*,2021,57(21): 88–95.]
- [8] Yao Dechen,Li Boyang,Liu Hengchang,et al.Residual life prediction of rolling bearing based on attention GRU algorithm[J]. *Vibration and Impact*,2021,40(17):116–123.[姚德臣,李博阳,刘恒畅等.基于注意力GRU算法的滚动轴承

- 剩余寿命预测[J].*振动与冲击*,2021,40(17):116–123.]
- [9] Qin Yi, Chen Dingliang, Xiang Sheng, et al. Gated dual attention unit neural networks for remaining useful life prediction of rolling bearings[J]. *IEEE Transactions on Industrial Informatics*, 2020, 17(9): 6438–6447.
- [10] Zhao Sen, Zhang Yong, Wang Shang, et al. A recurrent neural network approach for remaining useful life prediction utilizing a novel trend features construction method[J]. *Measurement*, 2019, 146: 279–288.
- [11] Mo Yu, Wu Qianhui, Li Xiu, et al. Remaining useful life estimation via transformer encoder enhanced by a gated convolutional unit[J]. *Journal of Intelligent Manufacturing*, 2021, 32(7): 1997–2006.
- [12] Shen Li, Wang Yangzhu. TCCT: Tightly-coupled convolutional transformer on time series forecasting[J]. *Neurocomputing*, 2022, 480: 131–145.
- [13] Li Yasong, Zhou Zheng, Sun Chuang, et al. Variational attention-based interpretable transformer network for rotary machine fault diagnosis[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2024, 35(5): 6180–6193.
- [14] Wu Ting, Peng Junjie, Zhang Wenqiang, et al. Video sentiment analysis with bimodal information-augmented multi-head attention[J]. *Knowledge-Based Systems*, 2022, 235: 107676.
- [15] Chen Renxiang, Zhou Jun, Hu Xiaolin, et al. Fault diagnosis method of rotating machinery based on deep Q-learning and continuous wavelet transform[J]. *Journal of Vibration Engineering*, 2021, 34(5): 1092–1100. [陈仁祥, 周君, 胡小林等. 基于深度Q学习和连续小波变换的旋转机械故障诊断方法[J]. *振动工程学报*, 2021, 34(5): 1092–1100.]
- [16] Uchibe E, Doya K. Forward and inverse reinforcement learning sharing network weights and hyperparameters[J]. *Neural Networks*, 2021, 144: 138–153.
- [17] Qian Yiming, Xiong Fangzhou, Liu Zhiyong. Zero-shot policy generation in lifelong reinforcement learning[J]. *Neurocomputing*, 2021, 446: 65–73.
- [18] Wang Xu, Wang Tianyang, Ming Anbo, et al. Semi-supervised hierarchical attribute representation learning via multi-layer matrix factorization for machinery fault diagnosis[J]. *Mechanism and Machine Theory*, 2022, 167: 104445.
- [19] Zheng Jinde, Chen Minjun, Cheng Junsheng, et al. Multiscale fuzzy entropy and its application in rolling bearing fault diagnosis[J]. *Journal of Vibration Engineering*, 2014, 27(1): 145–151. [郑近德, 陈敏均, 程军圣, 等. 多尺度模糊熵及其在滚动轴承故障诊断中的应用[J]. *振动工程学报*, 2014, 27(1): 145–151.]
- [20] Zhang Yong, Xin Yuqi, Liu Zhiwei, et al. Health status assessment and remaining useful life prediction of aero-engine based on BiGRU and MMoE[J]. *Reliability Engineering & System Safety*, 2022, 220: 108263.
- [21] Wang Biao, Lei Yaguo, Li Naipeng, et al. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings[J]. *IEEE Transactions on Reliability*, 2018, 69(1): 401–412.
- [22] Qiu Hai, Lee J, Lin Jing, et al. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics[J]. *Journal of Sound and Vibration*, 2006, 289(4/5): 1066–1090.
- [23] Tobon-Mejia Diego Alejandro, Medjaher Kamal, Zerhouni Nouredine, et al. A data-driven failure prognostics method based on mixture of Gaussians hidden Markov models[J]. *IEEE Transactions on reliability*, 2012, 61(2): 491–503.
- [24] Adewoyin Rilwan A, Dueben Peter, Watson Peter, et al. TRU-NET: A deep learning approach to high resolution prediction of rainfall[J]. *Machine Learning*, 2021, 110(8): 2035–2062.

Remaining Useful Life Prediction of Rolling Bearings Based on Policy Gradient Informer Model

XIONG Jiahao¹, LI Feng^{1*}, TANG Baoping², WANG Yongchao¹, LUO Ling³

(1. School of Mechanical Eng., Sichuan Univ., Chengdu 610065, China;

2. State Key Lab. of Mechanical Transmissions, Chongqing Univ., Chongqing 400044, China;

3. National Inst. of Measurement and Testing Technol., Chengdu 610021, China)

Abstract: As a typical encoder–decoder, the transformer architecture has inherent limitations such as secondary time complexity, high memory usage, and a complex model structure; these issues can lead to lower prediction accuracy and decreased computational efficiency when applied to the prediction of the remaining useful life (RUL) of rolling bearings. For this reason, herein, a novel encoder–decoder, the Policy Gradient Informer (PG-Informer) model, is proposed and applied to the prediction of the RUL of rolling bearings for the first time. First, in the new encoder–de-

coder architecture of PG-Informer, a probabilistic sparse self-attention mechanism is used to replace the original self-attention mechanism of the transformer architecture to improve its nonlinear approximation ability and reduce its time and space complexity. Then, the self-attention distillation operation is used to reduce its number of parameters and their dimensions and enhance the prediction robustness of time series. Moreover, the generative decoder of PG-Informer only needs to decode the decoding input in one step to output the prediction results without dynamic multistep decoding, which improves the prediction speed of time series. Finally, a policy-gradient learning algorithm is constructed to improve the training speed of the PG-Informer parameters. These advantages enable the proposed rolling-bearing RUL-prediction method to obtain higher prediction accuracy, better robustness, and higher computational efficiency. Results for the No. 1 rolling bearing at the Center for Intelligent Maintenance Systems of the University of Cincinnati show that the proposed method was able to predict an RUL value of 963 min, representing a prediction error of only 6.50% when compared to the experimental result; compared to the transformer-based RUL prediction method, this represents a higher prediction accuracy, a smaller prediction error, and greater robustness. The proposed method consumed only 132.37 s for RUL prediction, shorter than the time taken by the transformer-based RUL-prediction method. These results verify the effectiveness and advantages of the proposed method.

Key words: Informer model; probabilistic sparse self-attention mechanism; policy gradient; rolling bearing; remaining useful life prediction

(编辑 吴芝明)

引用格式: Xiong Jiahao, Li Feng, Tang Baoping, et al. Remaining useful life prediction of rolling bearings based on Policy Gradient Informer model[J]. *Advanced Engineering Sciences*, 2024, 56(4): 273–286. [熊佳豪, 李锋, 汤宝平, 等. 基于策略梯度 Informer 模型的滚动轴承剩余寿命预测[J]. *工程科学与技术*, 2024, 56(4): 273–286.]