

西安交通大学

硕士学位论文

基于深度学习的无监督时序数据异常检测与多变量时序数据分类的研究

学位申请人：高瀚

指导教师：鲍军鹏教授

学科名称：计算机科学与技术

2024 年 04 月

Unsupervised Time Series Data Anomaly Detection and Multivariate Time Series Data Classification Based on Deep Learning

A thesis submitted to
Xi'an Jiaotong University
in partial fulfillment of the requirements
for the degree of
Master of Engineering

By

Han Gao

Supervisor: Associate Prof. JunPeng Bao

Computer Science and technology

April 2024

硕士学位论文答辩委员会

基于深度学习的无监督时序数据异常检测与多变量时序数据分类 的研究

答辩人：高瀚

答辩委员会委员：

西安交通大学教授：XXX (注：主席)

西安交通大学教授：XXX

西安交通大学副教授：XXX

西安交通大学研究员：XXX

西安交通大学高工：XXX

答辩时间：2024 年 04 月 12 日

答辩地点：西安交通大学

摘要

时间序列数据是按时间顺序采集到的一系列观测值的集合。随着各行业智能化的普及和网络通信技术的进步,各种产品、设备和系统在不断产生大量的时序数据。这些海量时序数据的积累,催生了广泛的时序数据分析任务,如异常检测和分类等。因此如何有效学习时序数据中相关模式与特定任务之间的关系是学术界与工业界的重要研究方向。时序数据无监督异常检测模型通过挖掘大规模无标签数据的固有模式和特征,对新数据进行异常检测。现有的异常检测模型难以同时学习时序数据的长期模式和短期模式,进而难以检测出时序数据中不同尺度的异常。多变量时序数据分类模型通过学习时序数据中有助于分类的特征和规律,建立多变量时序数据到标签的映射。传统的时序数据分类模型主要用于单变量时序数据分类,难以直接应用在长序列多变量时序数据上。针对上述问题,本文开展了无监督时序数据多尺度异常检测模型和多变量时序数据分类模型的研究。具体工作如下:

(1) 针对时序数据无监督多尺度异常检测的挑战,本文提出了一种多尺度异常检测模型 MMTSAD。MMTSAD 通过降采样得到不同粒度的时序数据,将不同粒度的序列分别输入不同的自编码器。在训练阶段,自编码器通过重构训练集输入数据学习不同尺度的模式和特征,以训练集重构误差为损失函数指导自编码器参数优化,在测试阶段,自编码器通过重构测试集输入数据得到不同尺度下的重构误差,以测试集重构误差作为异常分数检测不同尺度的异常。为了综合利用多个自编码器得到的不同尺度的异常分数,本文还为 MMTSAD 模型设计了一种基于分布调整的异常分数融合算法。在实验部分,MMTSAD 在 5 个公开的实际应用数据集上进行了实验,对比了以往的 10 个异常检测模型,并在其中 4 个数据集上都取得了最优的 F1* 分数,说明了 MMTSAD 具备优秀的无监督异常检测性能。

(2) 针对长序列多变量时序数据分类的挑战,本文提出了一种基于 Patch 的多变量模型 PatchMTSC。PatchMTSC 通过对多变量数据进行 Patch 操作得到连续的子序列表示,将单变量子序列作为 Transformer 编码器的输入 token,使编码器直接学习子序列之间的依赖,同时提升模型可接受的数据长度。在多变量关系学习方面, PatchMTSC 通过傅里叶变换得到多变量时序数据的频域表示,并以一定比例采样得到稀疏频域表示,再输入自注意力机制层,以低参数量学习多变量频域之间的关系。在实验部分, PatchMTSC 在 UEA 的 20 个公开数据集上进行了实验,在其中 7 数据集上取得了相对于其他方法更好的准确率,在另外 3 个数据集上取得了次优的准确率。同时 PatchMTSC 还在 20 个数据集上的平均准确率也达到了最优。说明了 PatchMTSC 在多变量时序数据上的优秀分类性能。

关 键 词： 时间序列；异常检测；无监督学习；分类；深度学习网络

论文类型： 应用研究

ABSTRACT

Time series data refers to a collection of sequential observations gathered in chronological order. With the proliferation of automation, intelligence, and advancements in network communication technology across various industries, an abundance of time series data is continuously generated by various products, devices, and systems. In recent years, deep learning has been widely applied in the field of time series data, leading to numerous works utilizing deep learning models for unsupervised anomaly detection tasks. However, existing anomaly detection methods mainly employ fixed-length sliding windows to extract windowed data from observation data, and then individually train and detect anomalies using deep learning encoding. Yet, as the frequency of time series data collection in practical production increases, training with a single fixed-length window becomes increasingly challenging for capturing long-term patterns and anomalies of different scales within the time series. Moreover, due to the quadratic growth of time complexity and space complexity with the length 'L' of the time series, simply extending the sliding window length leads to rapid growth in model training time consumption and memory usage. Additionally, the complexity and diversity of patterns that may appear in multi-dimensional time series data require models capable of detecting anomalies at different scales. Addressing the aforementioned issues, this paper proposes a multi-scale, multi-variable time series data unsupervised anomaly detection model, MMTSAD. The model obtains time series data of different granularities through downsampling and employs AE modules at different granularities to detect anomalies by reconstructing input data. Furthermore, a distribution-adjusted anomaly score fusion calculation method is proposed to comprehensively consider the weights of anomaly scores at different scales. Multi-variable time series data classification is an important task with widespread applications in various fields such as finance, healthcare, and meteorology. However, due to the complexity of multi-variable time series data and the ever-increasing scale of time series data, traditional classification methods often struggle to effectively handle such data. Therefore, this paper proposes a model called PatchchTSC, aiming to address the challenges in multi-variable time series data classification tasks. By leveraging frequency domain information to learn relationships between multi-variable time series data with low parameter count, efficiently processing long sequence data through Patch operations, and effectively utilizing Transformer encoder tokens, the model extends the acceptable input sequence length.

KEY WORDS: Time series; anomaly detection; unsupervised learning; classification; deep learning networks

TYPE OF THESIS: Application Research

目 录

摘 要	I
ABSTRACT	III
1 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	2
1.2.1 基于深度学习的时序数据无监督异常检测方法	2
1.2.2 基于深度学习的多变量时序数据分类方法	4
1.3 论文主要研究内容	6
1.4 论文的结构安排	6
2 时间序列数据分析相关技术	8
2.1 时间序列数据概述	8
2.1.1 时间序列数据定义	8
2.1.2 多变量时间序列数据特点	9
2.2 时间序列数据异常检测方法	10
2.3 时间序列数据分类方法	11
2.4 时间序列相关模型	12
2.4.1 Transformer 模型	12
2.4.2 AutoEncoder 模型	15
2.5 本章小结	16
3 基于多尺度的无监督多维时序数据异常检测算法	17
3.1 引言	17
3.2 时间序列数据的异常检测任务	17
3.3 多尺度时序数据异常检测模型 MMTSAD	18
3.3.1 降采样模块	18
3.3.2 维度注意力模块	19
3.3.3 多尺度自编码器模块	21
3.3.4 模型损失函数	22
3.4 基于数据分布调整的多尺度异常分数融合	22
3.4.1 本文中重建误差与异常分数的定义	22
3.4.2 多尺度模块基于分布调整的异常分数融合	23
3.5 实验结果与分析	25
3.5.1 实验环境与数据集说明	25
3.5.2 异常检测结果评价指标与实验设置	26
3.5.3 与其他算法模型的对比结果	28
3.5.4 无监督学习效果实验	30

3.5.5 模型消融实验	31
3.5.6 多尺度异常检测可视化实验	33
3.5.7 不同超参数对模型性能的影响	35
3.6 本章小结	37
4 基于 Patch 的多变量时序数据分类模型	38
4.1 引言	38
4.2 多变量时序数据分类任务的定义	38
4.3 基于 Patch 的多变量时序数据分类模型 PatchMTSC	39
4.3.1 PatchMTSC 模型思路	39
4.3.2 PatchMTSC 模型结构	39
4.4 实验结果与分析	42
4.4.1 实验环境	42
4.4.2 数据集说明	42
4.4.3 异常检测结果评价指标与实验设置	45
4.4.4 与其他算法模型的对比结果	46
4.4.5 t-SNE 可视化实验	48
4.4.6 模型消融实验	50
4.4.7 不同频域采样方法对模型性能的影响	50
4.4.8 不同超参数对模型性能的影响	51
4.5 本章小结	52
5 结论与展望	53
5.1 结论	53
5.2 展望	54
致谢	55
参考文献	56
攻读学位期间取得的研究成果	60
答辩委员会会议决议	61
常规评阅人名单	62
声明	

CONTENTS

ABSTRACT (Chinese)	I
ABSTRACT (English)	III
1 Introduction	1
1.1 Research Background and Significance.....	1
1.2 Research Status	2
1.2.1 Unsupervised Anomaly Detection for Time Series Data Based on Deep Learning	2
1.2.2 Deep Learning Based Classification for Multivariate Time Series Data	4
1.3 Main Research Contents of Thesis.....	6
1.4 Thesis Organization.....	6
2 Technologies related to time series data analysis	8
2.1 Overview of Time Series Data	8
2.1.1 Time series data definition	8
2.1.2 Characteristics of multivariate time series data.....	9
2.2 Examples	10
2.3 Examples	11
2.4 Examples	12
2.4.1 Level of this document.....	12
2.4.2 Level of this document.....	15
2.5 Examples	16
3 Algorithm and Code	17
3.1 Algorithm	17
3.2 Level of this document	17
3.3 Algorithm	18
3.3.1 Level of this document.....	18
3.3.2 Level of this document.....	19
3.3.3 Level of this document.....	21
3.3.4 Level of this document.....	22
3.4 Algorithm	22
3.4.1 Level of this document.....	22
3.4.2 Level of this document.....	23
3.5 Algorithm	25
3.5.1 Level of this document.....	25
3.5.2 Level of this document.....	26
3.5.3 Level of this document.....	28
3.5.4 Level of this document.....	30

3.5.5 Level of this document.....	31
3.5.6 Level of this document.....	33
3.5.7 Level of this document.....	35
3.6 Algorithm	37
4 Patch-based classification model for multivariate time series data.....	38
4.1 introduction	38
4.2 Definition of the classification task for multivariate time series data	38
4.3 Patch-based multivariate time series data classification model PatchTSC	39
4.3.1 Ref	39
4.3.2 Ref equations	39
4.4 Ref.....	42
4.4.1 Ref	42
4.4.2 Ref	42
4.4.3 Level of this document.....	45
4.4.4 Level of this document.....	46
4.4.5 Level of this document.....	48
4.4.6 Level of this document.....	50
4.4.7 Level of this document.....	50
4.4.8 Level of this document.....	51
4.5 Algorithm	52
5 Requirements of other parts	53
5.1 Abstracts.....	53
5.2 Abstracts.....	54
Acknowledgements.....	55
References	56
Achievements	60
Decision of Defense Committee	61
General Reviewers List	62
Declarations	

1 绪论

1.1 研究背景及意义

相应的对时间序列数据的分析在工业自动化、商业、金融和医疗领域等许多领域中都具有重要的意义，其分析的主要目的是根据不同的任务要求，从历史已有观测数据中提取模式、趋势和规律，以完成包括异常检测、分类和预测在内的不同下游任务。其中对时间序列数据的异常检测有助于系统识别潜在的异常和事故，尽早进行异常排查，避免造成经济损失。

异常检测，即从数据中识别意外事件或状态的过程，经典的异常检测方法包括基于传统线性模型的方法^[1]、基于距离的方法^[2]、基于密度的方法^[3]和支持向量机^[4]等。然而随着实际系统采集到时序数据的规模越来越大，模式越来越复杂，这些经典方法逐渐暴露出诸多缺陷，比如无法处理多维数据，难以检测缺失标记的异常等。近年来，基于深度学习的方法逐渐被引入时序数据分析领域，深度学习通过精心设计的神经网络学习时序数据的复杂模式，不需要对数据潜在模式进行假设，因而具有更强的泛化能力。

在各种异常检测技术中，使用无监督学习的异常检测方法受到最近研究的重视，因为无监督学习不需要人工标注的数据就可以训练模型，并且可以更好地适用于训练集未见异常，具有更高的实际应用价值。

在实际生产中，系统绝大多数时间处于正常状态，只有极少时间会出现异常。所以相应的绝大多数时序数据都是系统正常运行时采集到的，只有少部分时序数据才会包含系统的异常状态，比如工业生产故障、金融高风险信贷行为以及云服务异常流量等，并且大多数异常类似边缘案例 (corner case)，难以复现或复现成本极高。因此大规模的经过标记的异常数据非常难以获得。而无监督方法可以使用任意状态采集到的时序数据作为训练样本，学习时序数据普遍模式和特征，就能对可能包含任何异常模式的测试时序数据进行异常检测。

综合以上现状，基于深度学习的无监督时序数据异常检测方法有非常高的研究价值。

时间序列数据分类是将时序数据分类或标记为预定义的类的任务。由于医疗保健、环境监测、制造等领域的系统和传感器产生了大量时间戳数据，由于绝大多数信号由多个传感器记录，所以序列中天生包含多个变量，因此多变量时序数据分类任务具有非常高的实用价值。在医疗领域中，时间序列分类用于患者监控、疾病辅助诊断和治疗优化。例如，可以辅助批量分类心电图 (ECG) 信号以诊断心脏状况，或辅助批量分类

脑电信号 (EEG) 以检测神经系统疾病。时间序列分类还在人体行为识别系统中起着至关重要的作用,可以用于医疗保健、体育、安全和人机交互等领域。通过分析加速度计或陀螺仪等传感器的时间序列数据,它可以对行走、跑步、坐着或站着等活动进行分类。在语音和音频处理领域中,时间序列分类可用于说话人识别、情感检测和语音识别等任务。通过分析语言时序信号,以对口语单词、语音中传达的情感进行分类,或根据语音特征识别说话人。传统的时序数据分类方法主要分为基于模式的和基于特征的。最近基于深度学习的时序数据分类模型逐渐得到广泛应用,因为相比传统方法,深度学习模型不依赖领域专家知识,泛化能力强且推理耗时少。

因此,基于深度学习的时序数据分类具有广泛的应用场景和研究意义。

1.2 国内外研究现状

1.2.1 基于深度学习的时序数据无监督异常检测方法

从异常检测的方式来看,主要有以下几种方法:分别是基于预测的方法、基于重建的方法以及基于聚类或密度的方法。其中广泛与深度学习相结合的主要是基于预测和重建的方法。基于预测的方法可以训练预测模型学习一段时间的历史数据模式,来预测下一个时间步的期望值,并通过预测值和实际采集值之间的差异来判断该时间戳是否处于异常状态。基于重建的方法可以通过模型捕获一段时间内数据的隐层表示,根据隐层表示重构原始数据,根据每个时间戳的重构数据与原始数据的差异来判断该时间戳是否处于异常状态。

从基础模型的种类看,目前运用于时序数据异常检测的深度学习模型主要包含 CNN 类、RNN 类、GNN 类、以及注意力类模型和混合模型等。

CNN 类模型通过堆叠卷积层,在每一层中学习原始数据不同层次的特征,并且通过引入池化层,使 CNN 类模型能够捕获时间序列中的复杂特征^[5-6]。但是 CNN 类模型仍无法明确学习时序数据代表的时间背景,只能在较短时间段内学习数据模式。因此,He 等人^[7]提出了一种基于时间卷积网络 (TCN) 的时序数据异常检测方法,其中 TCN 网络采用因果卷积和膨胀,使传统卷积网络适应具有时间性和大感受野的序列数据。

RNN 类模型是时间序列建模的首选之一,因为它们在处理时间步之间的依赖关系时表现出色。在时间序列中,每个数据点都与前一个时间步的数据点相关,而 RNN 类模型能够捕捉和利用这种时序信息。RNN 类模型在每个时间步都使用相同的权重参数,这种参数共享有助于捕获序列中的模式,并减少模型的参数数量,使其更容易训练。并且针对长期依赖捕捉和梯度消失的问题,又提出了改进模型,如 LSTM 和 GRU,使模型在处理长序列数据时更为有效。基于 RNN 的方法通常用于两种方式的异常检测。一种即为基于预测的方法^[8-9],另一种方法是使用 RNN 类模型构造自编码器 (AE) 或变分

自编码器 (VAE), 使用编码器重构原始数据^[10-11]。Li 等人^[12]提出了 InterFusion, 一种同时对多维时间序列 (MTS) 的维度间依赖和时间依赖进行建模的无监督方法, 其核心思想是通过具有两个随机潜变量的分层变分自编码器对 MTS 数据中的正常模式进行建模, 两个变量分别学习低维的维度间和时间嵌入。然而由于 RNN 类模型的时序依赖性使得每个时刻的计算都依赖于前一个时刻, 导致难以进行有效的并行计算。模型在训练和推理过程中, 需要将数据按照时间次序逐个输入到模型, 这样将显著增加模型训练和推理所消耗的时间, 限制了模型在长序列时序数据上的实际应用价值。

GNN 类模型主要包含图卷积网络 (GCN)^[13]和图注意力网络 (GAT)^[14]两种, 可以通过图结构捕捉多维时序数据中不同维度之间的依赖关系。GNN 类模型中图节点通过相互交换信息来迭代地更新它们的表示, 对应到多维时序数据异常检测模型中, 即是每个维度被表示为图中的单个节点。其中 GCN 类模型通常需要研究人员从多维时序数据中提取维度关系, 以形成图结构, 或者依据一定的先验知识直接构建图结构, 比如由相关领域的专家指定不同维度数据之间的关系。而 GAT 类模型可以并行的计算维度之间的所有关系, 因此可以不需要预先指定全局图结构。Deng 等人^[15]提出了一种基于注意力的图神经网络 GDN, 模型将向量嵌入以捕捉单个传感器特征作为节点, 并将传感器之间的相关性 (维度关系) 作为图中的边进行捕获, 该模型通过对相邻传感器上的注意力函数进行学习, 从而学习预测传感器的行为。它能够在没有监督的情况下, 通过子图、注意力权重以及预测数据与实际数据的比较来检测异常。Zhao 等人^[16]提出了包含两个并行的图注意力 (GAT) 层的 MTAD-GAT 模型, 用于自监督多变量时间序列异常检测, 这样模型可以提取时间序列不同维度之间的相关性 (维度依赖关系) 并学习时间步之间的关系 (时间依赖关系)。该模型定义了一个基于预测和重构的综合目标, 其中预测模型仅预测一个时间步, 而重构模型学习整个时间序列的潜在表示。Chen 等人^[17]提出了 GTA 模型, 充分利用了 Transformer 的序列建模能力, 并采用了双向图结构来学习多个物联网传感器之间的关系。提出了一种新的影响传播 (IP) 图卷积作为传感器之间依赖关系图结构的学习策略。模型将每个节点的邻域限制在一定范围内以进一步提高推理效率, 同时融合了多尺度扩张卷积和图卷积以提供分层的时间上下文编码。其中使用基于 Transformer 的架构来建模和预测序列, 并且作者还提出了一种使用多分支注意力来减少多头注意力的二次复杂度的替代方法。

随着 Transformer 类模型在 NLP、CV 等领域广泛应用, 其中包含的 Attention 机制也被引入时序数据分析领域中。相比 RNN 类模型需要对输入数据逐个时间步进行计算, Attention 类模型可以并行计算整个输入数据, 因为在计算输入数据注意力权重矩阵时, 每个时间步的权重计算是独立的。这使得在处理时序数据时, Attention 模型的训练和推理过程可以更有效地利用硬件资源, 加速模型的训练和推理过程。另外传统的 RNN 类模型在处理长期依赖关系时可能面临梯度消失或梯度爆炸的问题, 尤其是对于较长的

序列。**Attention** 机制通过对不同时间步的输入赋予不同的注意力权重,可以更灵活地捕捉长期依赖关系,而不会受到梯度消失的影响。由于 **Attention** 机制不包含显式的时间次序,因此需要人为指定序列中时间步的相对或绝对位置,即位置编码。Xu 等人^[18]提出了一种带有 **Attention** 机制的异常检测模型 **AnomalyTransformer**, 目的是通过同时对每个时间步建模先验关联和序列关联来捕捉关联差异,以使罕见的异常更容易被区分。因为异常很难与整个序列联系起来,而正态分布则更容易与相邻的时间戳联系起来,所以先验关联利用高斯核估计相邻的集中归纳偏差,而序列关联使用从原始数据学到的自注意权重。除了重建损失之外,还设计了一种 **MINIMAX** 方法,以增加关联差异的正常-异常区分程度。Tuli 等人^[19]提出一种具有自调节和对抗性训练的深层 **transformer** 异常检测模型 **TranAD**。模型使用基于注意力的序列编码器,根据数据中更广泛的时间趋势的知识快速执行推理。**TranAD** 使用基于焦点分数的自调节来实现鲁棒的多模态特征提取和对抗训练,以获得稳定性。此外,与模型无关的元学习 (**MAML**) 可以使用有限的数据训练模型。Li 等人^[20]提出了一种称为扩张卷积 **Transformer-GAN(DCT-GAN)** 的无监督方法。使用 **Transformer** 来处理时间序列数据,使用基于生成对抗网络 (**GAN**) 的模型来重建样本和检测异常,使用扩张的 **CNN** 结构来从潜空间中提取时间信息。该算法将多个不同尺度的 **Transformer** 生成模块组合在一起,在基于 **GAN** 的框架内获得粗粒度和细粒度信息,以提高其泛化能力。该模型采用基于权重的机制来整合生成模块,使其能够兼容不同类型的异常。

1.2.2 基于深度学习的多变量时序数据分类方法

时间序列数据分类方法主要可以分为距离度量方法、**shapelets** 类方法、深度学习方法。因为深度学习在大规模数据集上的应用优势和处理多变量时序数据的便捷性,基于深度学习的时序数据分类算法在近年来得到了广泛的应用。

因为时序数据分类任务不涉及预测时序数据,不需要遮盖未来信息,而只需要对时序数据进行特征提取。所以 **RNN** 和 **LSTM** 类模型的优势无法发挥,反而其在并行训练上的缺陷更加明显。所以实际运用并不多。**CNN** 类模型因为特定的优势得到了较为广泛的运用:**CNN** 模型可以通过改变卷积核的步长进行不同尺度的表示,通过权值共享机制控制模型参数量,保持特征提取的时间不变性,另外卷积运算不受序列顺序限制,方便并行计算。Karim 等人^[21]提出了 **LSTM-fcn**, 利用 **LSTM** 层和堆叠的 **CNN** 层从时间序列中生成特征。这些特征随后被传送到 **softmax** 层输出每个类的概率。Hao 等人发现了一个问题,即基于 **cnn** 的模型不能很好地捕获不同变量之间的长期依赖关系。Hao 等人^[22]为了解决这个问题,他们在基于 **cnn** 的模型上增加了两个交叉关注模块。Zhang 等人^[23]借鉴了传统方法和深度学习方法的优点。提出了一种旨在解决多元领域问题 **TapNet** 网络。**TapNet** 将 **LSTM**、**CNN** 和注意原型学习技术应用于 **MTSC**。Zhang

等人指出, 深度学习方法擅长在不需要嵌入领域知识的情况下学习低维特征, 而传统方法 (如 1NN-DTW) 在相对较小的数据集上表现良好。TapNet 结合了这些优势, 产生了一个可以分解为三个不同模块的网络架构: 随机维度排列、多元时间序列编码和注意原型学习。其中采用随机维度置换产生随机选择的 g 组维度, 目的是增加学习维度值组合如何影响类别值的可能性, 引入 LSTM 层的目的是为了学习纵向特征。Li 等人^[24]提出了 ShapeNet, 将时间序列子序列表示学习到统一空间中, 用于最终的 shapelet 选择。但并没有考虑时间序列子序列中的位置信息。Gao 等人^[25]提出了 RLPAM, 首先将 MTS 转换为单变量聚类序列, 然后使用强化学习来选择模式。Transformer 类模型因为其强大的长距离建模能力在时序数据分类领域得到了广泛应用。Rußwurm 等人^[26]研究了循环神经网络、卷积神经网络和自注意力模型光学卫星原始时序数据分类上的表现, 结果显示自注意力和循环神经网络, 在原始卫星时间序列上的表现优于卷积神经网络。Liu 等人^[27]提出了门控 Transformer 网络 GTN, 通过同时对通道相关性和时间步相关性进行建模, 实现对多变量时序数据的分类。Zerveas 等人^[27]提出了一种基于 Transformer 编码器的新框架, 用于学习下游任务, 例如分类和回归, 利用 transformer 网络学习时间序列的无监督表示, 以缓解数据稀疏性问题。Cheng 等人^[28]提出了一种多尺度的时序数据分类模型 FormerTime, 模型采用分层网络架构来执行多尺度特征映射, 并设计了一种新型的 Transformer 编码器, 该编码器包含时间约简的注意层和上下文位置编码生成策略。FormerTime 作者认为 Transformer 模型应用于多变量时序数据问题时有严重的问题, 在自然语言处理中, 由于词嵌入是基本元素, 因此只能学习固定尺度的表示。相比之下, 时间序列中单个时间步的信息密度太小, 无法反映与类标签相关的有用模式。以心电分类为例, 信息模式通常由一系列连续时间点或各种子序列表征, 而不是单个时间点数据。为了解决上述问题, 作者在每层网络中都使用 CNN 处理输入数据, 使用一维卷积将输入序列按照窗口投影, 进行特征提取, 通过卷积核的大小和步长控制每一层的投影窗口。作者阐述这样做的动机是时间序列和语言域之间的信息密度差异。与语言数据中的符号不同, 时间序列是人类生成的信号, 具有高度的语义和信息密集型, 而时间序列是自然冗余的, 例如, 缺失的点可以很容易地从其相邻点中恢复。作者任务采用时间切片操作的好处有以下两点: 首先, 序列数据的时间尺度可以灵活转换, 自然迫使网络生成分层特征映射。其次, 通过这种划分策略, 可以大大减少整个时间序列在送入编码器之前的序列长度, 节省了大量的计算消耗。Zuo 等人^[29]将传统的 shapelets 方法与深度学习进行了结合, 提出了 SVP-T 模型。作者采用滑动窗口的方式为所有样本生成大量的时间序列子序列, 其次使用聚类算法, 如 kmeans, 计算得到每个变量对应的 k 个簇心, 它们代表各个变量对应的 k 个子序列。通过将距离簇心最近的来自不同变量的子序列拼接起来得到一个多变量时序数据样本的 shapelets 表示, 将输入数据转变为一维。因为 shapelets 输入的是形状, 所以为了保持输入样本中 shapelets 的相对

位置，作者还设计了一个可变位置编码层 (vp 层) 来利用每个形状的变量和位置信息。

1.3 论文主要研究内容

本文采用深度学习方法，研究时序数据无监督异常检测任务和多变量时序数据分类任务。现有的时序数据无监督异常检测模型无法学习数据不同尺度的模式和特征，进而难以检测数据中不同尺度的异常。传统的时序数据分类模型通过学习单变量序列固定特征来使用特征分类，难以直接应用于多变量时序数据。针对以上两个任务的问题，本文主要工作如下：(1) 针对时序数据无监督多尺度异常检测的挑战，本文提出了一种多尺度异常检测模型 MMTSAD。MMTSAD 通过降采样得到不同粒度的时序数据，将不同粒度的序列分别输入不同的自编码器。在训练阶段，自编码器通过重构训练集输入数据学习不同尺度的模式和特征，以训练集重构误差为损失函数指导自编码器参数优化，在测试阶段，自编码器通过重构测试集输入数据得到不同尺度下的重构误差，以测试集重构误差作为异常分数检测不同尺度的异常。为了综合利用多个自编码器得到的不同尺度的异常分数，本文还为 MMTSAD 模型设计了一种基于分布调整的异常分数融合算法。在实验部分，MMTSAD 在 5 个公开的实际应用数据集上进行了实验，对比了以往的 10 个异常检测模型，并在其中 4 个数据集上都取得了最优的 F1* 分数，说明了 MMTSAD 具备优秀的无监督异常检测性能。

(2) 针对长序列多变量时序数据分类的挑战，本文提出了一种基于 Patch 的多变量模型 PatchMTSC。PatchMTSC 通过对多变量数据进行 Patch 操作得到连续的子序列表示，将单变量子序列作为 Transformer 编码器的输入 token，使编码器直接学习子序列之间的依赖，同时提升模型可接受的数据长度。在多变量关系学习方面，PatchMTSC 通过傅里叶变换得到多变量时序数据的频域表示，并以一定比例采样得到稀疏频域表示，再输入自注意力机制层，以低参数量学习多变量频域之间的关系。在实验部分，PatchMTSC 在 UEA 的 20 个公开数据集上进行了实验，在其中 7 数据集上取得了相对于其他方法更好的准确率，在另外 3 个数据集上取得了次优的准确率。同时 PatchMTSC 还在 20 个数据集上的平均准确率也达到了最优。说明了 PatchMTSC 在多变量时序数据上的优秀分类性能。

1.4 论文的结构安排

本文的组织结构安排如下：

第 1 章：绪论。本章首先介绍了本文工作的研究背景及实际意义。之后介绍了当前国内外基于深度学习的时序数据无监督异常检测和时序数据有监督分类的研究现状。最后说明了本文的组织结构安排。

第 2 章：时间序列数据分析相关技术。本章主要介绍了有关时间序列的相关概念和技术，首先阐述了单变量和多变量时序数据的定义和概念。其次介绍了现有的时序数据无监督异常检测方法和时序数据分类方法。最后介绍了广泛应用于时间序列异常检测和分类的深度学习模型 Transformer 和 AutoEncoder。

第 3 章：多尺度无监督异常检测模型。本章提出了一种用于解决无监督时序数据异常检测任务的模型 MMTSAD，并通过实验从多个角度说明了 MMTSAD 模型的优势。

第 4 章：多变量时序数据分类模型。本章提出了一种基于 Patch 的多变量时序数据分类模型 PatchMTSC。通过一系列的实验证实了 PatchMTSC 模型具备优秀的分类性能和泛用性。

第 5 章：总结与展望。本章重点总结了本文的工作内容，分析了在以后的科学研究中可能会出现的一些问题，同时对未来可能会进一步开展的研究趋向做出了阐述。

2 时间序列数据分析相关技术

2.1 时间序列数据概述

2.1.1 时间序列数据定义

时间序列数据是按时间顺序索引的一系列数据点。时间序列数据最常见的形式是随时间记录的观测序列。时间序列通常分为单变量(一维)和多变量(多维)。

1) 单变量时间序列数据

Univariate time series data 单变量时间序列 (UTS) 是随时间变化的单个变量的一系列数据点。单变量时间序列数据可以表示为:

$$\mathbf{X} = \{x_1, x_2, \dots, x_T\} \quad (2-1)$$

式中: T ——时间序列数据的长度; x_t ——单变量在时间戳 t 处的值。

2) 多变量时间序列数据

Multivariate time series data 多变量时间序列 (MTS) 表示依赖于同一个时间度量的多个变量, 其中每个变量都受到过去值(时间依赖)和基于相关性的其他变量(维度依赖)的影响。不同变量之间的相关性被称为维度相关性。多变量时序数据示意图如图2-1所示。

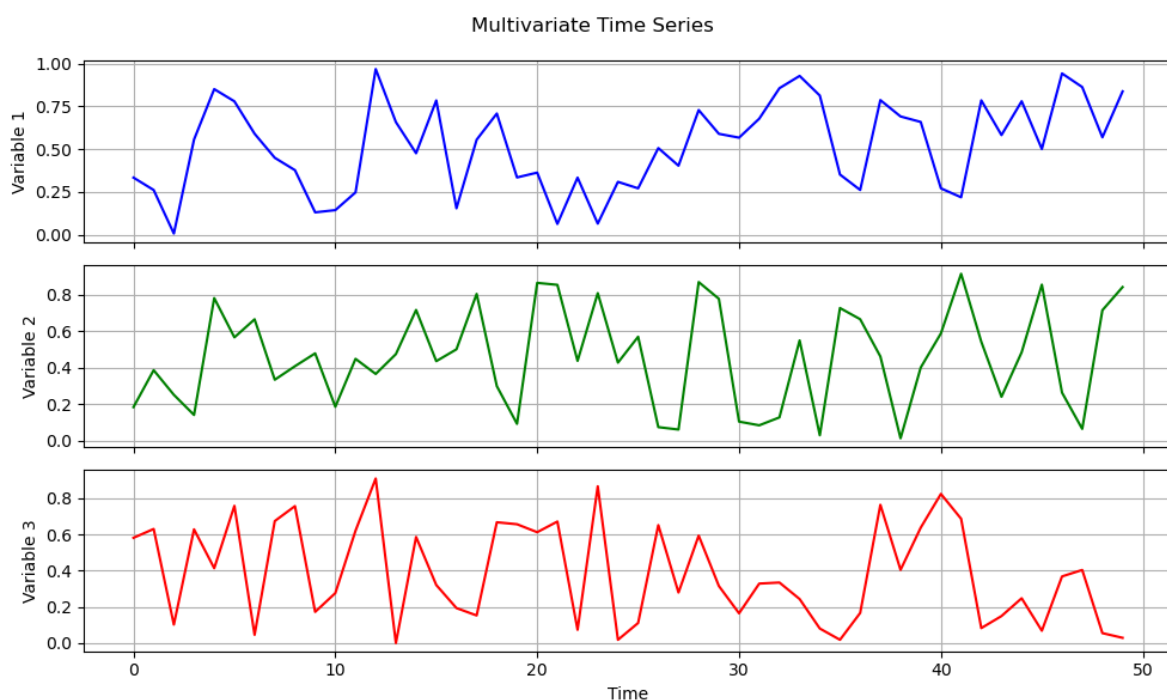


图 2-1 多变量时序数据结构示意

本文中的无监督异常检测任务和有监督分类任务的对象都是多维时序数据。

多变量时间序列数据可以表示为:

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\} \quad (2-2)$$

式中: T ——时间序列数据的长度; \mathbf{x}_t ——多变量在时间戳 t 处的值。

多变量时间序列数据在时间戳 d 处的值可以表示为:

$$\mathbf{x}_t = \{x_t^1, x_t^2, \dots, x_t^D\}, \quad t \in \{1, 2, \dots, T\} \quad (2-3)$$

式中: D ——多变量时间序列的变量数量; x_t^d ——第 d 个变量在时间戳 t 处的值。

2.1.2 多变量时间序列数据特点

多变量时序数据是一种非常复杂的数据,其主要具有以下的特点:(1)大规模:随着 IoT 设备和通信设备的快速发展以及数据存储成本的降低,时序数据的采集频率越来越高,生产实践中许多领域都会生成 TB 级别的大规模时序数据。(2)高维性:在包含 d 个传感器的系统中,同一时间点会记录 d 个传感器的值,即生成多变量时序数据,变量数为 d 的单个时间点数据通常被视为 d 维空间中的一个点。在实际应用中, d 值可能非常大,很容易达到上千个测点。在这种高维稀疏空间对时序数据进行变量关系学

习和相似度计算十分困难。(3) 趋势、周期性和随机性：时序数据通常包含分量：趋势性、季节性、周期性和一定的随机性分量。趋势性分量代表数据随时间变化的长期模式，季节性主要是数据中固定频率的季节性模式，周期性分量一般是短期有明显规律的重复性波动，当系统出现突发情况时，会导致时序数据出现随机性分量。(4) 噪声影响：时序数据通常来自真实物理系统，系统外界环境和传感器本身都可能导致数据中包含噪声，导致数据可能存在振幅和周期的飘逸等现象。这会对时序数据的特征学习带来一定干扰。

2.2 时间序列数据异常检测方法

时间序列数据的异常在概念上体现为序列中部分数据及其罕见，不符合常态，或者在逻辑上是不合理的。在数据上体现为序列中单个观测值(点)或一系列观测值(子序列)与数据的一般分布有很大的偏离。在真实世界的的数据中可能存在大量的噪声，而这些噪声可能与研究人员无关[4]。

异常检测算法首先设计模型，再使用训练集数据训练模型来拟合时间序列分布，表征数据的内在特征。无监督异常检测任务的

对于一个给定的多维时序数据 X ，异常检测的目的是为每一个时间点计算出异常分数 $\mathbf{a} = \{a_1, a_2, \dots, a_T\}$ 和相应的阈值 Θ ，

作为一个重要的现实问题，针对无监督时间序列异常检测任务已经提出了许多方法。如果根据异常判断标准进行分类，现有方法主要包括基于密度估计、基于聚类、基于重建和基于预测的方法。基于密度的方法有 mppccad^[30]和 DAGMM^[31]等，这些工作通过整合高斯混合模型来估计表征的密度，进而判断异常。在基于聚类的方法中，异常分数通常由样本到聚类簇心的距离得到。并且随着深度学习的广泛应用，一些方法将深度学习模型与聚类方法相结合进行异常检测。

Ruff 等人^[32]提出 Deep SVDD 模型将正常数据的表示收集到紧凑的聚类中。Shen 等人^[9]提出了 THOC 模型，通过分层聚类机制融合中间层的多尺度时间特征，并通过多层距离检测异常。

基于重建的模型试图通过重建模型输入数据来检测异常。最简单的重建技术是使用 UAE 对每个通道依次训练一个单独的模型^[33]。

Park 等人^[34]提出了 LSTM-VAE 模型，该模型使用 LSTM 主干进行时间建模，并使用变分自编码器(VAE)进行重建。Su 等人^[35]提出的 OmniAnomaly 进一步扩展了 LSTM-VAE 模型，采用归一化流程，利用重构概率而不是异常分数进行检测。另外还有基于提升异常分数性能的改进，USAD^[36]采用两个加权重建误差，OmniAnomaly^[35]采用“重建概率”作为替代异常评分，MTADGAT^[16]结合预测误差和重建概率，TranAD^[19]采用

综合重建误差和判别器损失作为异常评分。

基于预测的模型根据预测误差检测异常。Hundman 等人^[37]提出了一种无监督和非参数阈值方法来解释 LSTM 网络生成的预测。它建立了一个自动异常检测系统 LSTM-NDT 来监测航天器发回的遥测数据。Ding 等人^[38]提出了一种基于分层时间记忆 (Hierarchical Temporal Memory, HTM) 和贝叶斯网络 (Bayesian Network, BN) 的实时异常检测算法。Gugulothu 等人^[39]通过端到端学习框架将非时间降维技术和循环自编码器相结合, 用于时间序列建模。

2.3 时间序列数据分类方法

时间序列数据分类方法主要可以分为距离度量方法、shapelets 类方法、深度学习方法。基于距离的方法通常包含一个最临近分类器和特定的距离函数, 距离函数通过允许序列重新排列来补偿两个序列之间可能存在的偏移。一般最常用的距离函数是动态时间规整 (DTW), DTW 是一种用于比较两个时间序列之间相似性的方法。它可以解决在时间轴上长度不同、速度不同的序列之间的相似度比较问题。DTW 算法在语音识别领域和时间序列数据挖掘等领域得到了广泛应用。DTW 的基本原理是通过动态规划来计算两个序列之间的最佳匹配路径, 从而找到它们之间的最佳相似度。在这个过程中, 会考虑对齐两个序列的不同部分以最小化它们之间的总距离。具体来说, DTW 算法会计算一个距离矩阵, 该矩阵的每个元素表示将两个序列中的特定点对齐时的距离。然后, 通过在这个距离矩阵上进行动态规划来找到最佳匹配路径, 从而得到最终的相似度度量。

Shapelet^[40]是一种用于时间序列分类的特征提取方法。Shapelet 是指时间序列中具有代表性形状的子序列, 这些子序列能够很好地区分不同的类别。Shapelet 的核心思想是, 大多数子序列在不同类别之间的平均距离很小, 而只有某些特定形状的子序列在不同类别之间的平均距离较大, 因此可以作为区分不同类别的有效特征。Shapelet 类方法旨在发现这些具有代表性形状的子序列, 并将它们用作分类任务的特征。具体来说, 一个 Shapelet 可以被定义为一个长度为 l (通常比较短) 的子序列, 它在时间序列中出现的位置和形状对于区分不同类别的样本非常重要。Shapelet 类方法进行时序数据分类通常包括以下步骤:

第一步: 候选 Shapelets 生成, 通过在时间序列中使用滑动窗口, 生成一组候选 Shapelet。

第二步: Shapelet 排序, 使用某种评价准则 (如信息增益或基尼指数) 对候选 Shapelet 进行评估和排序。选择具有高评价准则值的 Shapelet 作为代表性 Shapelet。

第三步: 特征提取, 将选择的 Shapelet 用作特征, 通常使用距离度量来计算每个时

间序列与 Shapelet 之间的距离。

第四步：分类器训练，将提取的 Shapelet 特征输入到分类器中进行训练，从而实现时间序列的分类任务。

早期的 shapelets 算法枚举了所有可能的 shapelets，因此随着时间序列的规模不断扩大，模型性能显著下降。Karlsson 等人^[41]提出了一种基于形状的 MTSC 方法。该算法使用随机森林中随机选择的 shapelet 进行建模。但是使用 shapelet 进行时序数据分类也有一些缺点，比如，一般适用于单变量时序数据分类；依赖于距离度量：shapelet 的选择通常依赖于距离度量，例如欧氏距离或动态时间规整（DTW）距离。这些距离度量可能不适用于所有类型的时序数据，特别是在存在噪音或变化速度不同的情况下。

近年来，基于深度学习的模型在多变量时序数据分类任务上取得了显著的成功。首先基于距离的方法和基于 shapelet 的传统分类方法的问题是在大规模数据集上会产生昂贵的计算成本。而基于深度学习的分类模型在大规模数据上展现出明显的优势，大规模数据集有助于提升模型性能和泛化能力，同时充分利用模型的并行能力，提升训练效率。其次传统时序数据分类算法大多为单变量时序数据而设计。为了使单变量时序数据分类算法适用于多变量场景，需要对独立建立在每个维度上的模型进行集成。而基于深度学习的分类模型可以方便地同时适应单变量和多变量时序数据的分类。另外，深度学习模型可以进行端到端的学习，这显著减少了人工特征工程的工作量和对领域专家知识的需求，有助于快捷的扩展模型的应用领域。

2.4 时间序列相关模型

2.4.1 Transformer 模型

Transformer 模型是一种基于自注意力机制（Self-Attention Mechanism）的深度学习模型，最初由 Vaswani 等人^[42]于 2017 年提出，并在自然语言处理领域取得了巨大成功。它在机器翻译、语言建模、文本生成等任务上表现出色，并成为了现代深度学习模型中的一个重要组成部分。Transformer 模型的核心思想是摒弃了传统的循环神经网络（RNNs）和卷积神经网络（CNNs），而是使用了完全基于注意力机制的结构。在传统的序列模型中，信息传递是通过循环或卷积操作来实现的，这些操作存在一定的局限性，比如难以捕捉长距离依赖关系。而 Transformer 模型通过引入注意力机制，使得模型能够在不同位置之间建立直接的联系，从而更好地处理长距离依赖。这样的优势也对时间序列领域具有重大意义。Transformer 模型的核心思想是自注意力机制，它允许模型在处理序列数据时对不同位置之间的相关性进行建模，从而更好地捕捉输入序列的全局信息。自注意力机制是 Transformer 模型中的关键组成部分之一，它允许模型在处理序列数据时对不同位置之间的相关性进行建模，从而更好地捕捉输入序列的全局信息。

下面是自注意力机制的数学表达：

假设有一个输入序列 $X = \{x_1, x_2, \dots, x_n\}$ ，其中 x_i 是输入序列的第 i 个元素， n 是序列的长度。对于每个元素 x_i ，我们将其通过三个线性变换（权重矩阵）分别映射为三个向量：查询向量 q_i ，键向量 k_i ，值向量 v_i 。这些向量的计算方式如下：

$$q_i = x_i W^Q, \quad k_i = x_i W^K, \quad v_i = x_i W^V \quad (2-4)$$

其中 W^Q ， W^K ， W^V ——学习到的权重矩阵。

接下来，我们计算所有查询向量 $Q = \{q_1, q_2, \dots, q_n\}$ 和所有键向量 $K = \{k_1, k_2, \dots, k_n\}$ 的点积，并经过 Softmax 操作将点积结果转换为注意力权重。然后，将这些权重与对应的值向量 $V = \{v_1, v_2, \dots, v_n\}$ 相乘，并对结果进行加权求和，得到最终的输出表示：

$$\text{Attention}(Q, K, V) = \text{Softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (2-5)$$

其中， d_k ——查询向量和键向量的维度， QK^T ——查询与键的点积矩阵。

经过 Softmax 操作得到注意力权重，然后乘以值矩阵 V 。这样，我们就得到了自注意力机制的输出表示，它包含了输入序列中每个位置的上下文信息，能够更好地捕捉全局依赖关系。

在 Transformer 模型中，为了增强模型对不同表示空间的关注，引入了多头自注意力机制（Multi-Head Self-Attention）。多头自注意力机制允许模型同时关注输入序列的不同表示空间，通过并行计算多个自注意力机制，从而提高了模型的表示能力。多头自注意力机制模块如图2-2所示。

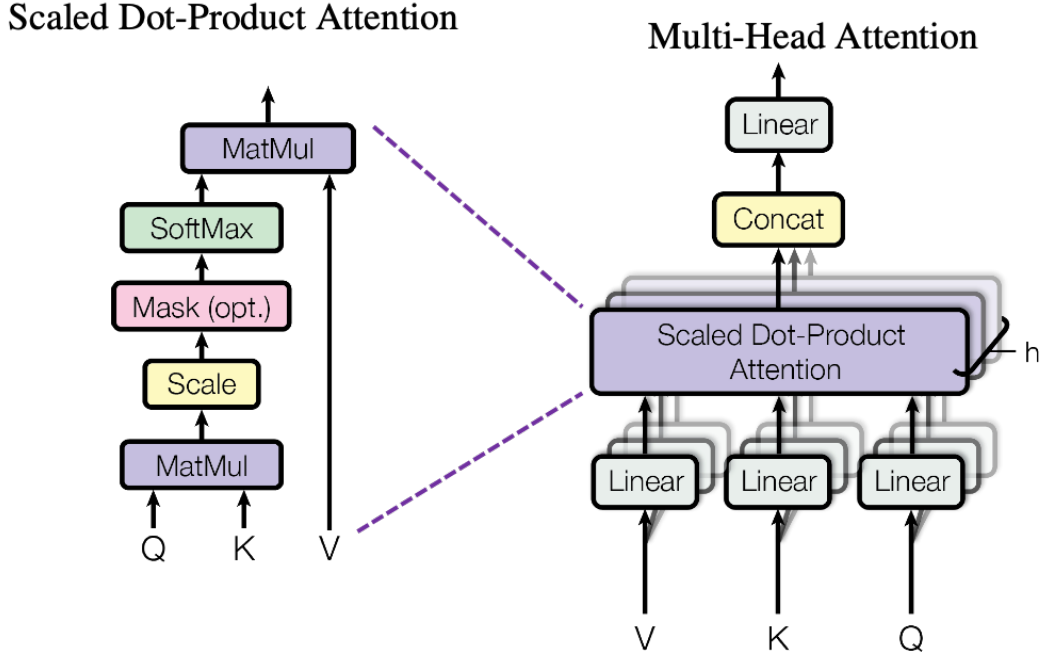


图 2-2 多头自注意力机制

具体来说，多头自注意力机制首先通过将输入序列进行线性变换，将其映射到多个不同的查询、键和值空间。然后，对每个映射后的查询、键和值执行自注意力机制。最后，将每个注意力机制的输出进行拼接，并通过另一个线性变换得到最终的多头自注意力输出。如果设定 h 个注意力头，那么查询向量公式如下：

$$Q^k = XW_Q^k, \quad K^k = XW_K^k, \quad V^k = XW_V^k \quad (2-6)$$

其中， k ——注意力头的索引； W_Q^k 、 W_K^k 、 W_V^k ——权重矩阵，用于线性变换； Q^k 、 K^k 、 V^k ——查询、键和值的映射矩阵。

$$A^k = \text{softmax} \left(\frac{Q^k (K^k)^T}{\sqrt{d_k}} \right) V^k \quad (2-7)$$

$$Z = \text{Concat}(A^1, A^2, \dots, A^h) W^O \quad (2-8)$$

其中， A^k ——第 k 个注意力头的输出； d_k ——每个注意力头的维度。Concat(\cdot) ——将多个注意力头的输出拼接在一起； W^O ——输出映射矩阵； Z ——多头注意力层的输出。

2.4.2 AutoEncoder 模型

AutoEncoder（自动编码器）是一种无监督学习的神经网络模型，其主要目的是将输入数据压缩成潜在空间的表示，并通过解码器来尝试重构原始输入数据。AutoEncoder 常用于数据压缩和降维、特征学习、去噪和生成数据。它由编码器和解码器两部分组成，AutoEncoder 结构如图2-3所示。

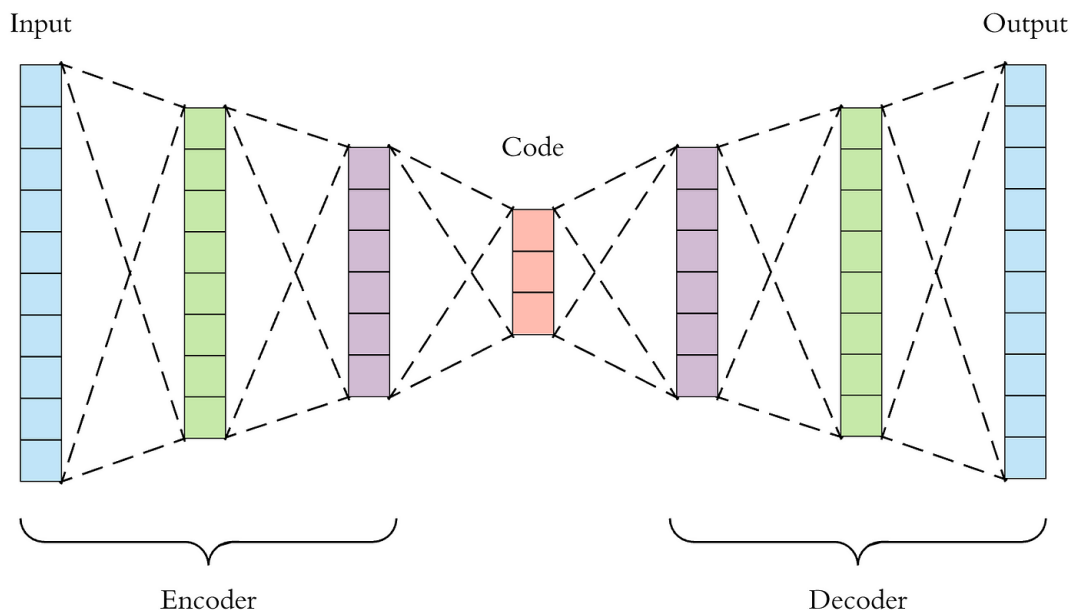


图 2-3 AutoEncoder 结构

假设我们有一个输入数据集 $X = \{x^1, x^2, \dots, x^m\}$ ，其中 x^i 是一个 n 维的输入向量。AutoEncoder 的目标是学习一个编码函数 $f_\theta: \mathbb{R}^n \rightarrow \mathbb{R}^d$ 和一个解码函数 $g_\phi: \mathbb{R}^d \rightarrow \mathbb{R}^n$ ，其中 θ 和 ϕ 分别是编码器和解码器的参数，使得重构误差最小化。

编码器将输入数据映射到潜在空间的表示。它通常由一系列的神经网络层组成，这些层逐渐减少输入数据的维度，将其转换为潜在空间中的编码。编码器的输出通常称为编码或隐藏表示。

$$z = f_\theta(x) = \sigma(Wx + b) \quad (2-9)$$

其中， x —— 输入数据； z —— 潜在空间的表示； W —— 编码器的权重矩阵； b —— 偏置向量； σ —— 激活函数。

解码器接受编码器的输出，并尝试重构原始输入数据。与编码器相反，解码器逐渐增加数据的维度，直到最终的重构与原始输入尽可能相似。解码器通常也是由一系列的神经网络层组成。

$$\hat{x} = g_\phi(z) = \sigma(W'z + b') \quad (2-10)$$

其中， \hat{x} —— 重构数据； W' —— 解码器的权重矩阵； b' —— 偏置向量； σ —— 激活函数。

AutoEncoder 的训练过程通常使用重构误差作为损失函数，该误差衡量原始输入数据与重构数据之间的差异。重构误差通常使用均方误差（Mean Squared Error, MSE）来度量：

$$L(x, \hat{x}) = ||x - \hat{x}||^2 \quad (2-11)$$

AutoEncoder 的训练目标是最小化重建误差，即最小化 $L(x, D(E(x)))$ 。

2.5 本章小结

主要介绍了时间序列数据分析的相关技术，包括时间序列数据的介绍、异常检测方法、分类方法以及相关模型。其中时间序列数据部分介绍了单变量和多变量时间序列的定义。异常检测方法部分罗列了常见的无监督异常检测方法，包括密度、聚类、重建和预测等方法及相关工作。分类方法部分罗列了距离度量、shapelet 和深度学习等分类方法及相关工作。最后介绍了用于解决时序数据问题的深度学习模型，Transformer 和 AutoEncoder 模型。

3 基于多尺度的无监督多维时序数据异常检测算法

3.1 引言

本章主要针对无监督多维时间序列数据异常检测的问题，提出了一种基于多尺度的异常检测算法 Multiscale Multivariate Time Series Anomaly Detection (MMTSAD)。以往的异常检测算法在处理长序列多维时间序列数据时，常使用固定长度的单个滑动窗口处理数据，并将得到的子序列数据作为模型的输入。

这导致现有的异常检测算法通常难以同时捕捉不同尺度的异常，尤其是序列中同时包含尺度很小的点异常和尺度较大的上下文异常的情况。另外多维数据中的不同维度之间的相关性也是以往方法难以解决的问题。

由于现有的异常检测方法在实际应用中存在以上的缺陷，针对上述问题，MMTSAD 方法以原数据作为细粒度数据并使用降采样操作处理多维时序数据获得粗粒度数据，同时使用两个滑动窗口处理不同粒度的数据得到两类子序列，作为两个异常检测模块的输入。这样方便模型在训练中同时学习不同尺度的时间序列分布，在测试中同时检测不同尺度的异常。异常检测模块采用基于 Attention 机制的自编码器对子序列数据进行编码和解码，以重建输入数据。其中用于处理粗粒度子序列的模块中引入了 GAT^[14]以学习多维数据中维度相关性。在检测阶段，本文提出了一种分布调整的多尺度异常分数融合方法，用于综合考虑模型在不同尺度模块中计算出的异常分数，得到最终的融合异常分数。

在实验部分，本章在 5 个公开的实际应用数据集上进行了实验，对比了以往的 10 个基准模型，并在其中 4 个数据集上都取得了最优的 F1* 分数，说明了异常检测模型 MMTSAD 具备优秀的异常检测性能。

3.2 时间序列数据的异常检测任务

无监督异常检测任务的训练集包含一个没有标签的多维时序数据 \mathbf{X}_{train} ，而测试集包含一个带标签 $\mathbf{y} = \{y_1, y_2, \dots, y_T\}$ 的多维时序数据 $\mathbf{X}_{test} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ ，标签与多维时序数据在每个时间戳上一一对应，表示多维时序数据在该时间戳上的是否属于异常状态。

异常检测需要在训练阶段训练模型学习数据分布和普遍特征，在测试阶段使用模型计算出测试集多维时序数据在每个时间戳上的异常分数 $\mathbf{s} = \{s_1, s_2, \dots, s_T\}$ 并给出阙值 θ 。检测标签可以通过如下公式得到：

$$\hat{y}_t = \text{Thresholding}(s_t, \theta) = \begin{cases} 1, & s_t > \theta \\ 0, & s_t \leq \theta \end{cases}, \quad t \in \{1, 2, \dots, T\} \quad (3-1)$$

异常检测算法的优化目标就是使检测标签 $\hat{\mathbf{y}} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_T\}$ 尽量与真实标签 $\mathbf{y} = \{y_1, y_2, \dots, y_T\}$ 一致。

3.3 多尺度时序数据异常检测模型 MMTSAD

多尺度时序数据异常检测模型 MMTSAD 使用了编码器-解码器的自编码器结构。具体来说，MMTSAD 的模型结构可以划分为以下四个模块：降采样模块、多尺度自编码器模块、维度注意力模块和异常分数融合模块，整体结构如图3-1所示。

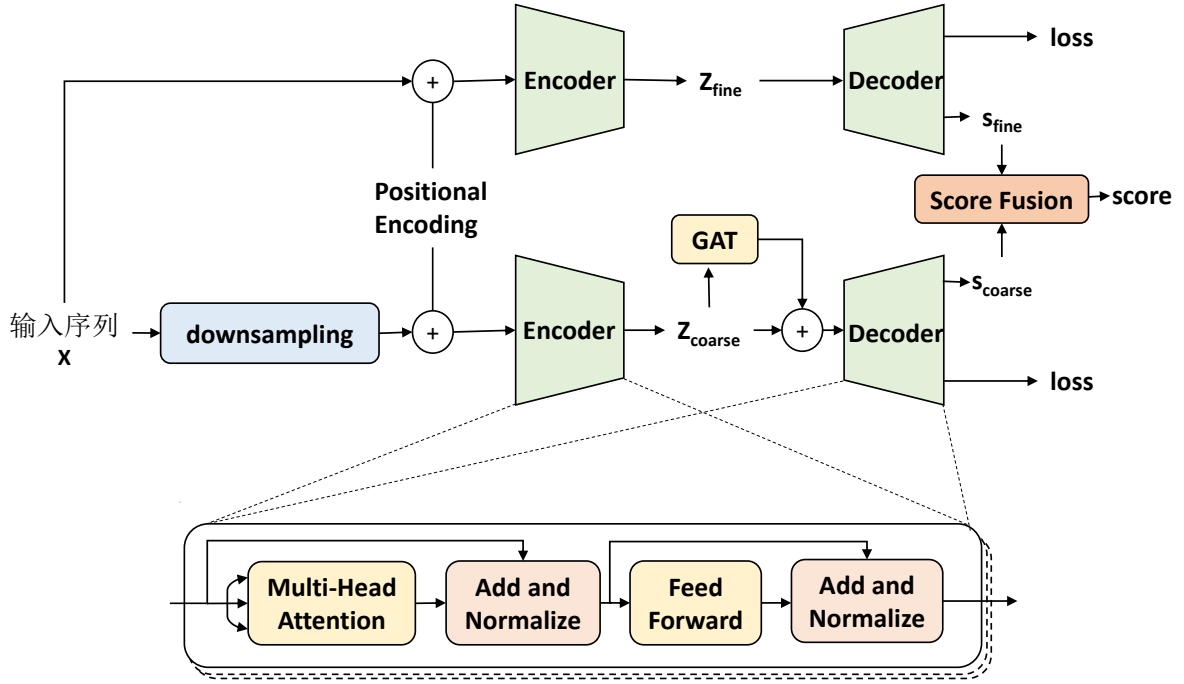


图 3-1 MMTSAD 结构图

3.3.1 降采样模块

transformer 因为其强大的长距离建模能力而在序列建模领域得到了广泛的运用，最初 transformer 首先在自然语言处理领域被提出，时序数据与自然语言数据都是典型的有序数据，自然语言中的词向量依赖于文本序列中的上下文顺序，时序数据中的数据也依赖于时间序列中的上下文顺序信息。transformer 已经通过位置编码辅助模型理解序列中不同元素的相对位置，从而更好地捕捉到序列中的顺序信息。但是时序数据的独

特之处还在于其趋势性和周期性的特点，这与自然语言的文本数据有所不同。时序数据的趋势一般包含短期趋势和长期趋势，往往长期趋势更能代表时序数据宏观上的变化方向，比如金融市场中股票价格的短期调整走势和长期走势。因此为了学习时序数据的长期趋势，检测可能出现的趋势异常，就需要扩大模型的输入序列长度。实际应用中产生的时序数据常常包含不同频率的周期，比如小时、天、周、月份就可能分别对应不同的周期。根据时域频域的不确定性原理，需要足够长的信号才能准确刻画信号的频域，即周期。所以为了模型学习到时序数据中周期信息，检测违背低频周期的异常，也需要扩大模型的输入序列长度。

然而 transformer 中计算 self-attention 矩阵的时间复杂度和空间复杂度均为 $O(N^2)$ (N 是输入模型的时间序列数据长度)，这意味着在处理长序列数据时，transformer 类模型的训练和推理会占据非常大的显存，并且消耗更长的时间。过高的训练和部署成本会显著降低模型的实用性，而且无监督异常检测任务一个常见场景就是流式数据的即时异常检测，因此对模型的推理速度要求较高。

基于以上限制，为了在一定的 transformer 输入序列长度内，扩展模型对时序数据的感受野，我们使用降采样技术处理原始粒度的数据，并且通过聚合函数和采样因子两个参数控制采样方式和采样粒度。其中聚合函数我们选择均值函数，因为均值采样可以一定程度上消除原始粒度序列中噪声的影响，使模型学习更有价值的序列模式，并且训练更容易收敛。采样因子取决于原始序列长度和采样频率，在相同输入长度的情况下，采样因子越大意味着模型对序列的感受野越大，这样有助于模型学习数据的长期趋势和低频周期，兼顾识别不同粒度的上下文异常。降采样模块的计算如图3-2所示。

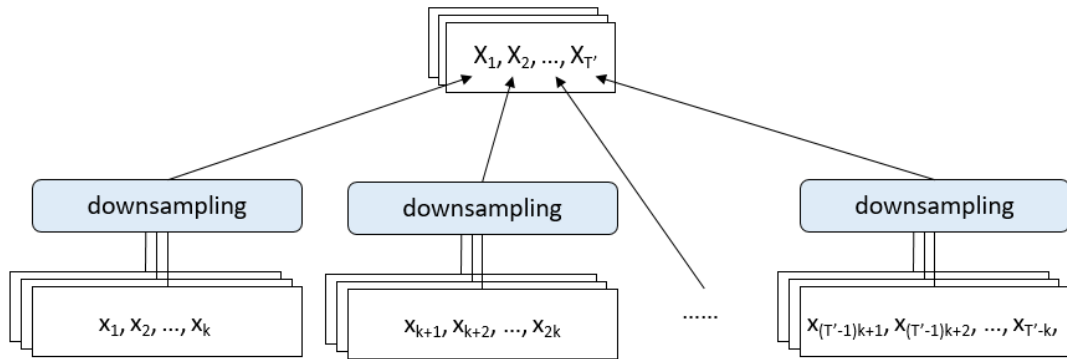


图 3-2 降采样模块计算示意图

3.3.2 维度注意力模块

本文设计了维度注意力模块用于学习变量序列之间的相互依赖关系，序列间关系的学习常使用图结构去建模，比如 GCN 和 GAT 等结构，在这类结构中，图中的每个节

点代表一个变量的时间序列。但是要使用 GCN 类图模型需要先得到图节点之间的邻接矩阵，对于有先验知识的任务，可以直接通过先验知识指定邻接矩阵。但这样处理有三个缺陷：首先是实际应用中有准确的先验知识的任务很少见，比如本章任务无监督多变量时序数据异常检测，该任务常用公开数据集都没有给出变量间的关系，其次是预先指定的邻接矩阵无法处理动态图问题，最后是难以实现不同节点之间的学习权重的分配。图注意力网络 GAT 较好的解决了上面的问题，使用 GAT 网络可以计算图中任意两个节点之间的注意力分数，并且可以根据注意力分数分配不同的权重给不同的节点，另外 GAT 构建的注意力关系可以随时间窗口的移动而变化，具有动态效果。

随着观测设备和技术的进步，实际应用中多变量时间序列数据记录的变量数量越来越多，一些大规模复杂系统产生的数据集可能在同一时间戳上记录了数百个变量的值，显然这样会使 GAT 网络的参数迅速增加，并且其中一般只有少数变量之间有一定的相关性。所以本文设计在自编码器的中间隐层表示上引入图注意力网络，相比原始数据，经过编码器压缩的中间隐变量拥有更高维的维度特征，冗余信息大幅度减少，因而图神经网络可以以极低的参数量学习隐层维度关系，帮助解码器重构原始数据。另外因为不同维度序列可能存在的高频周期时间迟滞，所以本章选择在粗粒度自编码器中引入图注意力网络，这样可以减少序列相位差异或者短期噪声趋势对模型训练的影响。维度注意力模块的计算如图3-3所示。

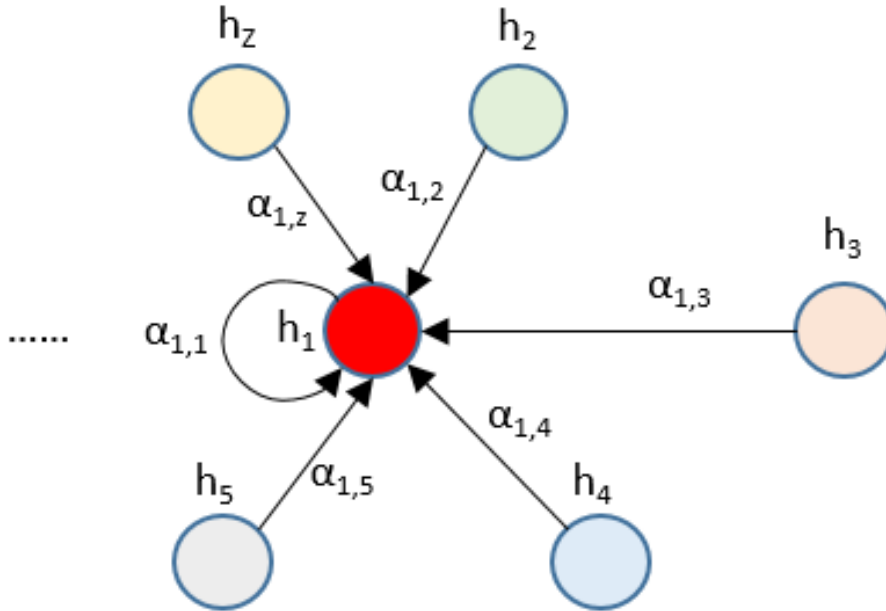


图 3-3 维度注意力模块计算示意图

粗粒度自编码器的中间隐变量 h 可以表示如下：

$$h = \{h_1, h_2, \dots, h_z, \dots, h_Z\} \quad (3-2)$$

式中： h_z ——图中的第 z 个节点，代表隐变量在第 z 个维度上的序列数据； Z ——隐变量维度。这样 h 就可以用一个图表示， h_z 代表图中的一个节点。图中节点 j 对节点 i 的重要性 e_{ij} 可以通过下面的公式计算：

$$e_{ij} = \text{LeakyReLU}(\vec{d}[Wh_i || Wh_j]) \quad (3-3)$$

式中： W ——代表图中各个节点共享的权重矩阵，用于将输入序列映射为图中节点特征； \vec{d} ——共享的单层前馈神经网络。

注意力系数 α_{ij} 表示如下：

$$\alpha_{ij} = \text{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{z=1}^Z \exp(e_{i,z})} \exp(x) \quad (3-4)$$

基于公式(3-3)和(3-4)，图中节点 i 聚合特征的输出 $h_{i,out}$ 等于图中每个节点特征由注意力系数加权后的和：

$$h_{i,out} = \sigma\left(\sum_{j=1}^Z \alpha_{ij} Wh_j\right) \quad (3-5)$$

式中： σ ——代表激活函数。

为了学习变量间关系，本章将 GAT 输出张量 h' 与隐变量 h 相加。

$$h_{out} = \{h_{1,out}, h_{2,out}, \dots, h_{z,out}, \dots, h_{Z,out}\} \quad (3-6)$$

$$h' = h + h_{out} \quad (3-7)$$

式中： h' ——结合 GAT 输出特征的隐变量。

h' 将被输入到解码器用于重构自编码器输入序列。

3.3.3 多尺度自编码器模块

如图3-1所示，我们在两个不同粒度的时序数据上使用两个不同粒度的模块（即 AutoEncoder）分别重建对应的输入序列。我们称这两个不同粒度的模块分别为细粒度模块和粗粒度模块，两个模块内部均由 AutoEncoder 构成。

AutoEncoder 中的 Encoder 用于压缩原始的时间序列数据 x ，得到输入数据 x 的潜在空间表示 z ，Decoder 接受潜在空间表示 z 作为输入，来重建原始输入数据 x 。

在细粒度自编码器中，编码器的输入序列长度和解码器的输出序列长度为 L ，意味着模型对序列的感受野同样也是 L ，而在粗粒度自编码器中，编码器的输入序列长度和解码器的输出序列长度同样为 L ，即粗粒度自编码器不会消耗比细粒度自编码器更多的

参数量、内存和计算时间，同时模型对序列的感受野可以被大幅提升到 $L * Sampling_{factor}$ 的尺度，这对于模型学习数据中的长期趋势和依赖非常重要。事实上，除了降采样外，还有一些模型也可以成倍提升序列感受野，比如 CNN 等模型中池化层等。但是因为模型最终需要重构输入数据，所以考虑到重构时模型训练优化的难度与重构精度，本文选择了降采样处理数据。

因为 Transformer 在序列建模上的优秀效果，我们使用其编码器部分作为 AutoEncoder 的 Encoder 和 Decoder 的基础结构，用于捕捉输入窗口内的时间模式。Transformer 类模型基于的自注意力机制的本身具有排列不变性，也就无法考虑到时序数据的先后次序。因此本文采用了固定位置编码来让模型感知到输入时间序列数据中每个时间步所代表的位置信息。

本章通过自编码器对时序数据的重建分数判断时间步是否处于异常状态。为了综合考虑细粒度自编码器和粗粒度自编码器的重建分数，检测不同尺度的异常，本章设计了基于分布调整的多尺度异常分数融合方法，具体方法将在 3.3.4 中介绍。

3.3.4 模型损失函数

我们的损失函数可表示为均方误差 (MSE)：

$$loss = \frac{1}{L} \sum_{i=1}^L (x^i - \hat{x}^i)^2 \quad (3-8)$$

式中： L ——输入窗口长度，即输入序列时间步长度； x^i ——输入序列第 i 个时间步上的值； \hat{x}^i ——第 i 个时间步上的重构值。

3.4 基于数据分布调整的多尺度异常分数融合

3.4.1 本文中重建误差与异常分数的定义

本文的模型基于重建误差计算测试集时间序列中每个时间点的异常分数，那么第 t 个时间点上第 i 个变量的重建误差定义为该变量的重建值与实际值的差值：

$$\Delta x_t^i = \hat{x}_t^i - x_t^i \quad (3-9)$$

式中： \hat{x}_t^i ——第 t 个时间点上第 i 个变量的重建值； x_t^i ——第 t 个时间点上第 i 个变量的原始值。

在实际应用中，通过训练学习到时间序列数据特征的模型计算出的重建值一般会以实际值为中心上下小幅度波动，在处于正常状态下的时间点上，重建值与实际值的差

异较小，而在处于异常状态下的时间点上，重建值与实际值的差异会较大。现实生产中的时序数据中正常状态占绝大多数，而异常状态较少。这些特征都与正态分布的特点非常类似。所以我们可以将重建误差 Δx_t^i 近似为服从均值为 0 的正态分布的随机变量：

$$\Delta x_t^i \sim \mathcal{N}(0, \sigma^2) \quad (3-10)$$

式中： σ —— 正态分布的标准差，代表序列重构误差的尺度参数。

那么经过尺度缩放后，即服从标准正态分布。

$$\frac{\Delta x_t^i}{\sigma} \sim \mathcal{N}(0, 1) \quad (3-11)$$

每个时间点的异常分数需要考虑该时间点上所有变量的重建误差，所以第 t 个时间点对应的异常分数定义为该时间点上所有变量的重建误差的平方和，其计算公式如下：

$$s_t = \sum_{i=1}^D (\Delta x_t^i)^2 \quad (3-12)$$

式中： D —— 多变量时间序列的变量数量。

那么经过缩放的重构误差的平方和将服从一个卡方分布：

$$\sum_{i=1}^D \left(\frac{\Delta x_t^i}{\sigma} \right)^2 \sim \chi^2(v), \quad v \leq d \quad (3-13)$$

$$s_t = \sum_{i=1}^D (\Delta x_t^i)^2 \sim \sigma^2 \chi^2(v) \quad (3-14)$$

式中： v —— 卡方分布自由度。

每个时间点的异常分数构成整个测试集时间序列的异常分数数组：

$$score = \{s_1, s_2, \dots, s_T\} \quad (3-15)$$

式中： T —— 测试集时间序列的长度。

3.4.2 多尺度模块基于分布调整的异常分数融合

本文多尺度异常检测模型包含细粒度模块和粗粒度模块，其中细粒度模块检测原始粒度的数据，粗粒度模块检测经过降采样的数据，因此两个模块计算得到的异常分数如下：

$$score_{fine} = \{s_1, s_2, \dots, s_T\} \quad (3-16)$$

$$score_{coarse} = \{s_1, s_2, \dots, s_{T'}\}, \quad T' = \lfloor \frac{T}{k} \rfloor \quad (3-17)$$

式中：\$T'\$ ——经过降采样的测试集时间序列的长度；\$k\$ ——降采样因子。

因为粗粒度模块的异常分数来自于对降采样后的序列的重构误差，所以为了将粗粒度异常分数映射到降采样前的对应时间点，本文在具体实验中将粗粒度异常分数在每个粗粒度时间点复制 \$k\$ 次，\$k\$ 为降采样因子，得到长度为 \$T_k\$ 的粗粒度异常分数数组。并截掉细粒度异常分数数组中下标大于 \$T' * k\$ 的部分，这样细粒度异常分数数组长度与粗粒度异常分数数组长度即可对齐，均为 \$T_k\$：

$$score_{coarse} = \{s_1, s_1, \dots, s_1, s_2, s_2, \dots, s_2, \dots, s_{T'}, s_{T'}, \dots, s_{T'}\} \quad (3-18)$$

$$= \{s_{1,coarse}, s_{2,coarse}, \dots, s_{t,coarse}, \dots, s_{T_k,coarse}\}, \quad T_k = T' * k \quad (3-19)$$

$$score_{fine} = \{s_{1,fine}, s_{2,fine}, \dots, s_{t,fine}, \dots, s_{T_k,fine}\} \quad (3-20)$$

式中：\$s_{t,coarse}\$ ——映射到降采样前时间尺度上的粗粒度异常分数；\$T_k\$ ——被降采样因子整除的序列长度；\$s_{t,fine}\$ ——细粒度异常分数。

因为不同尺度的模块学习相应尺度数据特征，以检测不同时间尺度上的异常，所以细粒度与粗粒度计算得到的重构误差具有不同的尺度，根据即服从不同尺度的正态分布：

$$\Delta x_{t,fine}^i \sim \mathcal{N}(0, \sigma_{fine}^2) \quad (3-21)$$

$$\Delta x_{t,coarse}^i \sim \mathcal{N}(0, \sigma_{coarse}^2) \quad (3-22)$$

式中：\$\sigma_{fine}\$ ——正态分布的标准差，代表细粒度重构误差的尺度参数；\$\sigma_{coarse}\$ ——正态分布的标准差，代表粗粒度重构误差的尺度参数。

因此根据重构误差得到的异常分数也具有不同的尺度，并且根据公式 (3-14) 可以转换为卡方分布：

$$s_{t,fine} = \sum_{i=1}^D (\Delta x_{t,fine}^i)^2 \sim \sigma_{fine}^2 \chi^2(v) \quad (3-23)$$

$$s_{t,coarse} = \sum_{i=1}^D (\Delta x_{t,coarse}^i)^2 \sim \sigma_{coarse}^2 \chi^2(v) \quad (3-24)$$

式中：\$D\$ ——多变量时间序列的变量数量。

基于以上分析，需要对异常分数的分布进行调整才能使不同粒度的异常分数具备

可比性。因为无监督异常检测需要判断每个时间步是否处于异常状态，并且细粒度模块检测未经降采样的原始粒度数据，所以本文将粗粒度异常分数的分布向细粒度异常分数的分布进行调整。最后将经过调整的粗粒度异常分数与细粒度异常分数相加得到融合异常分数，融合异常分数可以综合考虑不同尺度下的异常影响。所有时间点的融合异常分数构成最终的融合异常分数数组。

$$score_{t,fusion} = s_{t,fine} + \frac{\sigma_{fine}^2}{\sigma_{coarse}^2} * s_{t,coarse} \quad (3-25)$$

$$score_{fusion} = \{score_{1,fusion}, score_{2,fusion}, \dots, score_{T,fusion}\} \quad (3-26)$$

式中： D ——多变量时间序列的变量数量。

阈值

3.5 实验结果与分析

前面几个小节已经介绍了 MMTSAD 的基本结构。为了证明模型在实际应用中的有效性，本小节将从多个方面展开实验，主要包括：在公开数据集上与其他模型的异常检测效果进行对比的实验，证明模型无监督学习有效的实验，证明多尺度模型有效和异常分数融合方法有效的消融实验。以及不同超参数的对比实验。

3.5.1 实验环境与数据集说明

1) 实验环境

多尺度无监督多维时序数据异常检测方法 MMTSAD 的实验环境如表3-1所示。

表 3-1 实验环境表

实验环境	版本或型号
操作系统	Ubuntu 18.04.2 LTS
编程语言	Python 3.7
开发环境与组件	VSCode 1.85.1, Anaconda 4.7.10
相关程序库	Pytorch 1.12.0, numpy 1.21.5, pandas 1.3.5, matplotlib 3.5.2, scikit-learn 1.0.2
CPU	Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz
GPU	NVIDIA Tesla V100-PCIE-32GB

2) 数据集说明

本章在 5 个公开数据集上进行实验以评估模型性能：

WADI: 水分配试验台 (Water Distribution)^[43]: WADI 数据集来自一个精简的城市供水系统, 该系统有 123 个传感器和执行器, 运行了 16 天。前 14 天只有正常数据, 其余两天有 15 个异常段。

SWaT: 安全水处理试验台 (Secure Water Treatment)^[44]: SWaT 数据集是在 11 天内从一个具有 51 个传感器的按比例缩小的水处理试验台收集的。在最后 4 天内, 使用各种攻击方法注入了 41 个异常, 而在前 7 天内仅生成了正常数据。

PSM: 服务器集群指标 (Pooled Server Metrics)^[45]: PSM 数据集从 eBay 的多个应用服务器节点内部收集, 其中有 13 周的训练数据和 8 周的测试数据。

MSL 与 SMAP: 火星科学实验室 (Mars Science Laboratory rover) 和土壤水分主动被动卫星 (Soil Moisture Active Passive satellite)^[37,46]: MSL 和 SMAP 数据集是由 NASA 收集的公共数据集, 包含来自航天器监测系统意外异常 (ISA) 报告的遥测异常数据。数据集分别有 55 维和 25 维。训练集包含未标记的异常。

以上数据集的统计信息如表3-2所示。其中对于实体数量大于 1 的数据集, 训练集和测试集长度是其所有实体序列长度的总和, 异常率等于测试集中所有实体的异常时间点总数与序列长度的比率。

表 3-2 数据集原始数据统计

数据集	训练集长度	测试集长度	实体数量	数据维数	测试集异常率
WADI	1048571	172801	1	123	5.99%
SWaT	495000	449919	1	51	11.98%
PSM	132481	87841	1	25	27.76%
MSL	58317	73729	27	55	10.48%
SMAP	140825	444035	55	25	12.83%

3.5.2 异常检测结果评价指标与实验设置

1) 异常检测结果评价指标

为了对异常检测的结果进行量化评估, 本实验选择使用召回率 (Recall)、精确率 (Precision) 和 F1 分数三个指标作为衡量标准, 由于异常检测任务中测试集异常率较低, 所以三个指标中 F1 分数指标最为重要, 因为 F1 分数是精确率和召回率的调和平均值, 所以能综合考虑精确率和召回率的表现。指标计算方法为:

$$R = \frac{TP}{TP + FP}, \quad P = \frac{TP}{TP + FN}, \quad F1 = \frac{2PR}{P + R} \quad (3-27)$$

式中： R ——召回率； P ——精确率； TP ——被判断为异常且实际也为异常的时间点数量； FP ——被判断为异常但实际为正常的时间点数量； FN ——被判断为正常但实际为异常的时间点数量。

近年来，有工作提出了一种被称为点调整 (PA) 的评估技术^[47]，用于计算经过点调整的时间序列异常检测模型的最佳 F1 分数 (F1*PA)。这种评估技术认为，如果在该段内的一个点被确定为异常，则整个段是异常的。Schlegel 等人^[48]和 Sun 等人^[49]研究表明，点调整 PA 对现有方法的 F1 分数有很高的高估，难以衡量模型对异常区间的实际检测效果。所以本文采用最佳 F1 分数 (F1*)^[49]的方式计算 F1 指标，下文中统一用 F1* 代指最佳 F1* 分数。

根据公式(3-1)可以得到测试集序列中每个时间点的检测结果。

$$\hat{y}_t = \text{Thresholding}(\text{score}_{t,\text{fusion}}, \theta) = \begin{cases} 1, & \text{score}_{t,\text{fusion}} > \theta \\ 0, & \text{score}_{t,\text{fusion}} \leq \theta \end{cases}, \quad t \in \{1, 2, \dots, T'\} \quad (3-28)$$

$$\hat{y} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{T'}\} \quad (3-29)$$

式中： $\text{score}_{t,\text{fusion}}$ —— t 时间点的融合异常分数； θ ——使 F1 分数最大的阈值。

根据检测结果 \hat{y} ，可以计算出最佳 F1 分数 (F1*)。

$$F1^* = \arg \min_{\theta} F1(\hat{y}, y) \quad (3-30)$$

式中： y ——测试集标签，即序列时间点的实际状态。

2) 实验设置

表3-3显示了在实验数据集上训练 MMTSAD 所使用的超参数。

表 3-3 模型实验参数

参数\数据集	WADI	SWaT	PSM	MSL	SMAP
降采样因子	8	8	20	4	4
降采样函数	mean	mean	mean	mean	mean
细粒度滑动窗口步长	1	1	1	1	1
粗粒度滑动窗口步长	1	1	1	1	1
细粒度自编码器输入窗口长度	100	100	100	100	100
粗粒度自编码器输入窗口长度	100	100	100	100	100
多头注意力的头数量	14	9	5	11	5
训练轮数	200	200	200	200	200
优化器	Adam	Adam	Adam	Adam	Adam
细粒度模块学习率	1e-4	1e-4	1e-4	1e-4	1e-4
粗粒度模块学习率	1e-4	1e-4	1e-4	1e-4	1e-4
每个批次的样本数量	128	128	128	128	128
是否打乱训练集数据	True	True	True	True	True

3.5.3 与其他算法模型的对比结果

MMTSAD 在 5 个公开数据集上的表现如表3-4、表3-5和表3-6所示。对于多实体数据集，我们分别训练测试各个实体，最终汇总计算该数据集所有实体在测试集的平均表现。因为基于点调整的指标存在局限性^[49]，所以本章以未经点调整的指标对模型异常检测性能进行评估，具体指标分别为最优 F1 分数 (F1*)、召回率 (Recall) 和精确率 (Precision)，其中最优 F1 分数 (F1*) 指标最具参考性，因为 F1* 分数综合考虑了召回率和精确度。

表 3-4 公开数据集上的最佳 F1 分数 (F1*) 指标结果

方法\数据集	WADI	SWaT	PSM	MSL	SMAP
DeepSVDD ^[32]	0.322	0.724	0.601	0.212	0.203
LSTM-VAE ^[34]	0.227	0.776	0.455	0.215	0.235
DAGMM ^[31]	0.121	0.750	0.483	0.199	0.333
MSCRED ^[50]	0.046	0.757	0.556	0.25	0.170
OmniAnomaly ^[35]	0.223	0.782	0.452	0.207	0.227
MTAD-GAT ^[16]	0.437	0.784	0.571	0.275	0.296
GDN ^[15]	0.570	0.810	0.552	0.217	0.252
AnomalyTransformer ^[18]	0.108	0.220	0.434	0.191	0.227
TranAD ^[19]	0.415	0.669	<u>0.649</u>	0.251	0.247
NPSR ^[49]	<u>0.642</u>	<u>0.839</u>	0.648	0.551	0.505
Ours(MMTSAD)	0.653	0.855	0.652	<u>0.549</u>	0.521

表3-4展示了在 5 个公开数据集上 MMTSAD 和其他模型的 F1* 分数，可以看到，MMTSAD 在 5 个公开数据集中的 4 个取得了最高的 F1* 分数，并且在另外 1 个数据集上取得了次高的 F1* 分数。说明了 MMTSAD 模型具备优秀的异常检测性能。

表 3-5 公开数据集上的精确率 (P) 指标结果

方法\数据集	WADI	SWaT	PSM	MSL	SMAP
DeepSVDD ^[32]	0.297	0.706	0.489	0.157	0.163
LSTM-VAE ^[34]	0.196	0.792	0.387	0.154	0.153
DAGMM ^[31]	0.072	0.751	0.393	0.135	0.247
MSCRED ^[50]	0.025	0.758	0.457	0.212	0.134
OmniAnomaly ^[35]	0.186	0.812	0.377	0.144	0.177
MTAD-GAT ^[16]	0.507	0.834	0.462	0.235	0.245
GDN ^[15]	0.663	0.883	0.459	0.187	0.197
AnomalyTransformer ^[18]	0.262	0.211	0.317	0.131	0.154
TranAD ^[19]	0.343	0.654	0.512	0.221	0.186
NPSR ^[49]	<u>0.831</u>	<u>0.932</u>	0.501	<u>0.417</u>	<u>0.389</u>
Ours(MMTSAD)	0.849	0.967	<u>0.509</u>	0.420	0.394

表3-5展示了在 5 个公开数据集上 MMTSAD 和其他模型的精确率 (P) 分数，可以看到，MMTSAD 在 5 个公开数据集中的 4 个取得了最高的精确率分数。

表 3-6 公开数据集上的召回率 (R) 指标结果

方法\数据集	WADI	SWaT	PSM	MSL	SMAP
DeepSVDD ^[32]	0.351	0.742	0.779	0.326	0.269
LSTM-VAE ^[34]	0.269	0.760	0.551	0.356	0.506
DAGMM ^[31]	0.378	0.749	0.626	0.378	0.510
MSCRED ^[50]	0.287	0.756	0.709	0.304	0.232
OmniAnomaly ^[35]	0.278	0.754	0.564	0.368	0.316
MTAD-GAT ^[16]	0.383	0.739	0.747	0.331	0.373
GDN ^[15]	0.499	0.748	0.692	0.258	0.349
AnomalyTransformer ^[18]	0.068	0.229	0.687	0.352	0.431
TranAD ^[19]	<u>0.525</u>	0.684	0.886	0.290	0.367
NPSR ^[49]	0.523	<u>0.762</u>	<u>0.897</u>	0.811	<u>0.719</u>
Ours(MMTSAD)	0.530	0.766	0.905	<u>0.792</u>	0.768

表3-5展示了在 5 个公开数据集上 MMTSAD 和其他模型的召回率 (R) 分数，可以看到，MMTSAD 在 5 个公开数据集中的 4 个取得了最高的召回率分数。综合表3-5和表3-6可以看出，在 WADI 数据集和 SWaT 数据集上，MMTSAD 模型的准确率相对较

高，但召回率相对较低。在 PSM 数据集、MSL 数据集和 SMAP 数据集上，MMTSAD 模型的召回率相对较高，而精确率相对较低。这说明不同的数据集的异常分布特点可能不同，导致精确率和召回率的指标表现不够平衡。

3.5.4 无监督学习效果实验

因为本章异常检测任务是无监督学习，为了证明无监督学习在无标签的训练集上确实学习到了有助于异常检测的模型权重。本小节记录了 MMTSAD 模型在训练阶段中，每轮训练过程后的训练集 loss 值以及经过该轮训练后，模型在测试集上进行异常检测得到的 F1* 值。本小节使用 WADI 数据集和 SWaT 数据集进行实验，因为这两个数据集是单实体的。

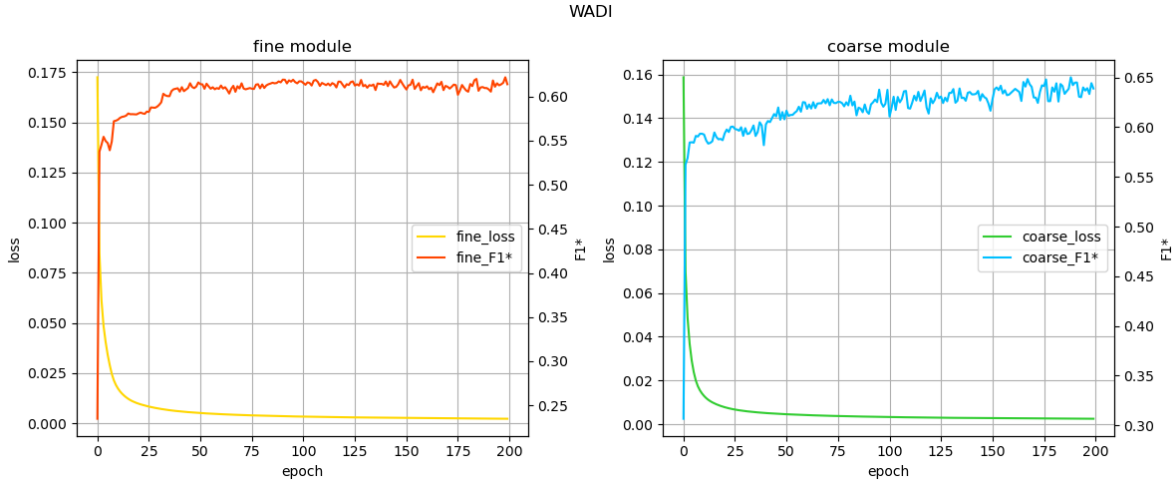


图 3-4 WADI 数据集上每轮训练后的训练集 loss 值和测试集的 F1* 值

图3-4显示了在 WADI 数据集上训练的前 200 个 epoch 过程中，细粒度模块（fine module）和粗粒度模块（coarse module）分别在每轮训练后的训练集 loss 值和测试集异常检测 F1* 值。图中，左侧子图是细粒度模块的表现，右侧是粗粒度模块的表现。可以看到随着训练轮数的增加，训练集 loss 逐渐减少，相应的模块在测试集上的 F1* 逐渐增加，说明了无监督学习过程中，在无标签的训练集上学习到的模型参数确实有助于提升模型在测试集上的异常检测效果，即无监督学习是有效的。图中可以看到粗粒度模块的训练集 loss 值相比细粒度模块的训练集 loss 值要小，这是因为经过降采样的粗粒度序列去除了很多噪声，让模型更容易重建输入序列。

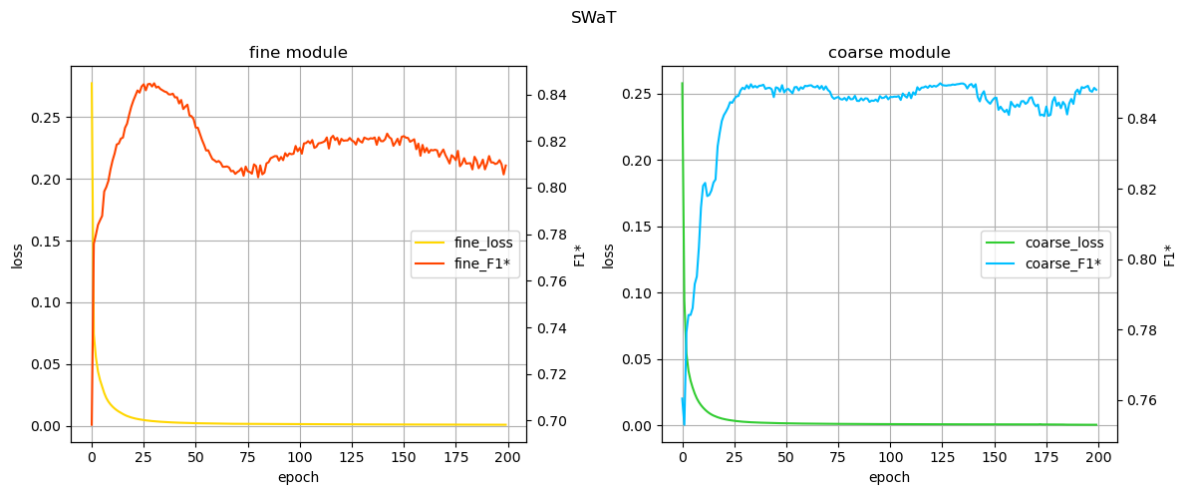


图 3-5 SWaT 数据集上每轮训练后的训练集 loss 值和测试集的 F1* 值

图3-5显示了在 SWaT 数据集上训练的前 200 个 epoch 过程中，细粒度模块（fine module）和粗粒度模块（coarse module）分别在每轮训练后的训练集 loss 值和测试集异常检测 F1* 值。随着 loss 值的降低，测试集异常检测 F1* 在上升，也证明了无监督学习的有效性。细粒度模块的 F1* 分数在大约 epoch=30 处，不再上升，转而开始下降，而粗粒度模块的 F1* 分数相对较为稳定，说明了此次训练中细粒度模块在训练集上学习时序数据普遍特征时存在过拟合现象，即训练集 loss 值在下降，测试集 F1* 值反而向下波动。粗粒度模块没有发生过拟合现象，说明降采样可以提取时序数据最通用和主要的信息用于模型训练，避免模型学习非通用的数据模式而产生训练集过拟合。

3.5.5 模型消融实验

消融实验：我们在五个公开数据集上进行了消融实验，如表 3 所示。消融实验针对 MMTSAD 中的不同尺度模块进行了研究，可以看到在所有数据集上，多尺度的 MMTSAD 都取得了比单一尺度模块更好的 F1* 分数，这说明了使用多尺度异常检测时的优秀效果。并且发现在使用单一尺度模块的情况下，细粒度模块在 WADI、MSL、SMAP 数据集上优于粗粒度模块，粗粒度模块在 SWaT、PSM 数据集上优于细粒度模块，这显示了不同数据集的异常可能主要分布在不同的尺度，因此也体现使用多尺度模块进行异常检测的必要性。

表 3-7 消融实验 F1* 指标结果

模型\数据集	WADI	SWaT	PSM	MSL	SMAP
MMTSAD	0.653	0.855	0.652	0.549	0.521
w/o coarse	0.615	0.821	0.646	<u>0.539</u>	<u>0.519</u>
w/o fine	<u>0.647</u>	<u>0.848</u>	<u>0.651</u>	0.532	0.502

为了证明本章提出的基于分布调整的多尺度异常分数融合方法的有效性,本节在 WADI 数据集和 SWaT 数据集上训练的前 200 个 epoch 过程中,记录了细粒度模块和粗粒度模块各自的 F1* 分数,以及经过多尺度融合的 F1* 分数。

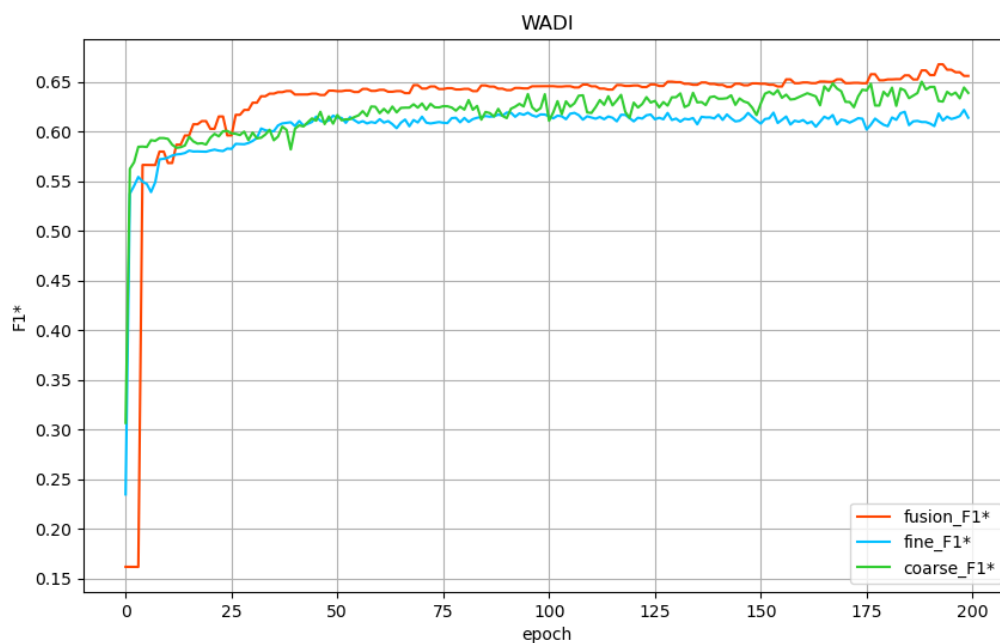


图 3-6 WADI 数据集上每轮训练后的细粒度和粗粒度模块的 F1* 值以及多尺度融合 F1* 值

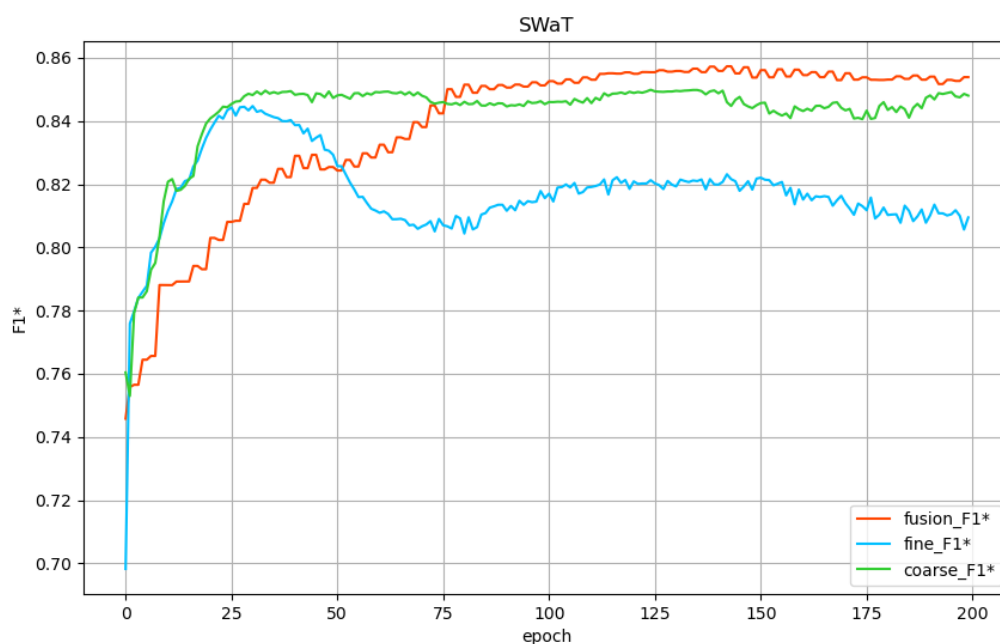


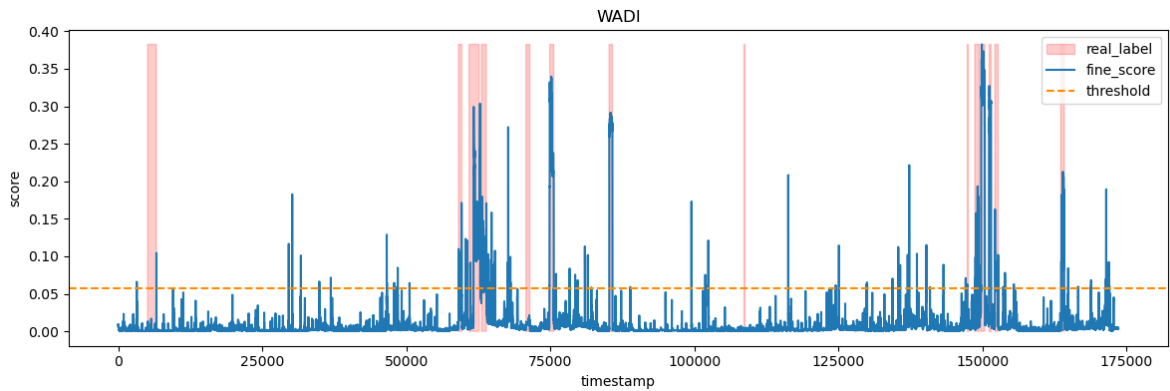
图 3-7 SWaT 数据集上每轮训练后的细粒度和粗粒度模块的 F1* 值以及多尺度融合 F1* 值

图3-6显示了在 WADI 数据集上训练的前 200 个 epoch 过程中, 在每轮训练后的细粒度模块和粗粒度模块分别进行异常检测得到的 $F1^*$ 值, 以及经过异常分数融合 (小节3.4.2所述) 后的多尺度模型得到的 $F1^*$ 值。可以看到, 在 $epoch \geq 15$ 后, 多尺度融合模型的 $F1^*$ 值一直大于两个子模块的 $F1^*$ 值, 说明了基于分布调整的多尺度异常分数融合的有效性。

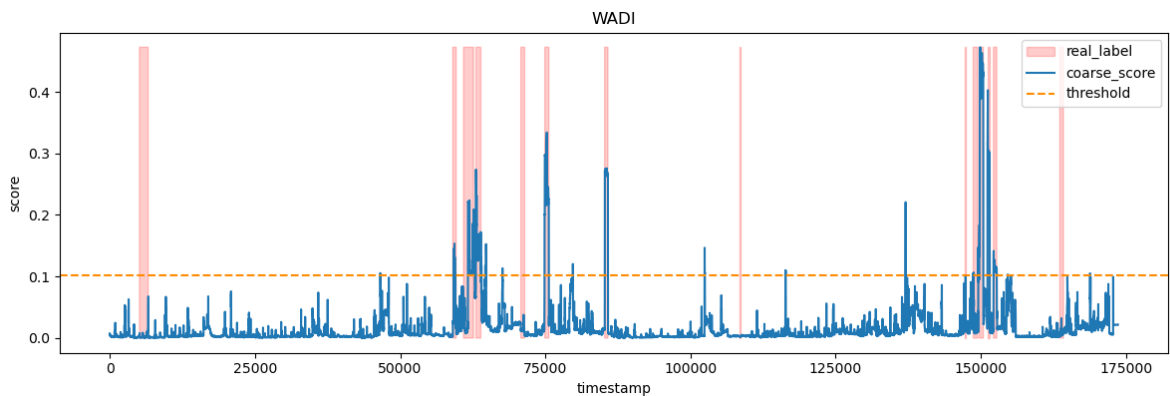
图3-7显示了在 SWaT 数据集上训练的前 200 个 epoch 过程中, 在每轮训练后的细粒度模块和粗粒度模块分别进行异常检测得到的 $F1^*$ 值, 以及经过异常分数融合 (小节3.4.2所述) 后的多尺度模型得到的 $F1^*$ 值。可以看到, 在 $epoch \geq 75$ 后, 多尺度融合模型的 $F1^*$ 值一直大于两个子模块的 $F1^*$ 值, 另外细粒度模块 $F1^*$ 分数的下跌对融合 $F1^*$ 分数并未造成明显的影响, 这是因为不同粒度模块检测出的异常有相当大的重合。以上情况说明了基于分布调整的多尺度异常分数融合的有效性和鲁棒性, 即使一个子模块发生过拟合导致异常检测性能下降, 融合异常分数仍能保证较高的异常检测性能。

3.5.6 多尺度异常检测可视化实验

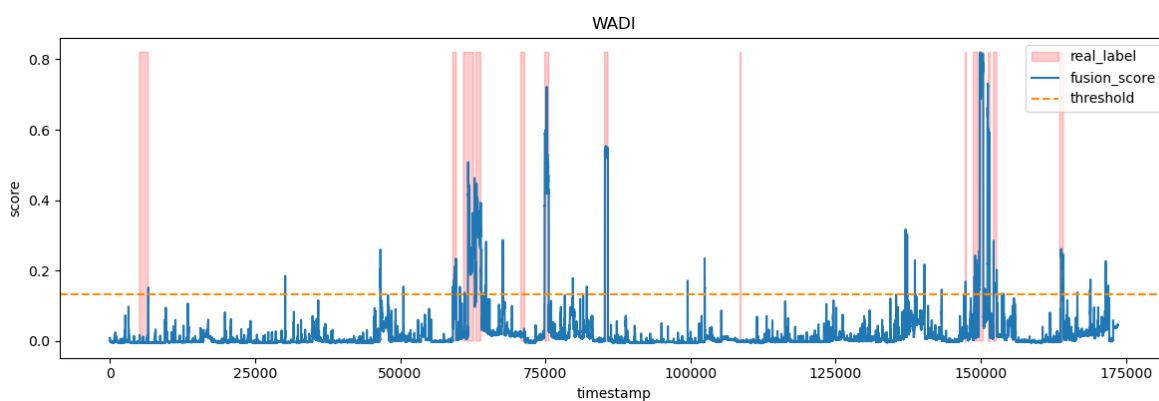
为了通过可视化的方式说明由不同粒度模块组成的多尺度异常检测模型的效果, 本节在单实体的 WADI 数据集上进行了实验。



(a) 细粒度模块



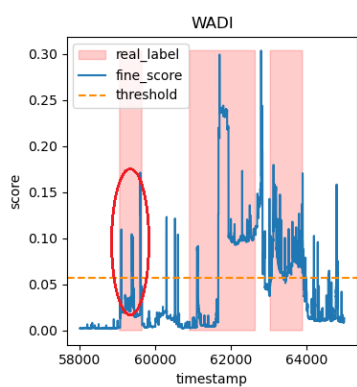
(b) 粗粒度模块



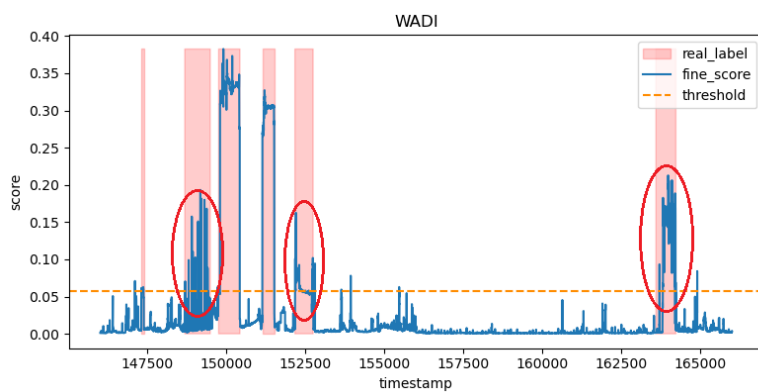
(c) 多尺度模型

图 3-8 WADI 测试集中的异常分数和真实异常标签

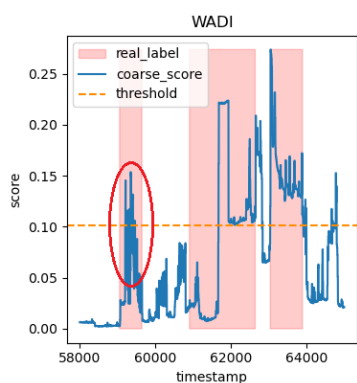
由图3-8可以看到细粒度模块计算得到的异常分数波动较大，导致了较高的误报次数，而粗粒度模块的异常分数相对波动较小，误报次数较少。经过异常分数融合后，明显减少了误报次数。



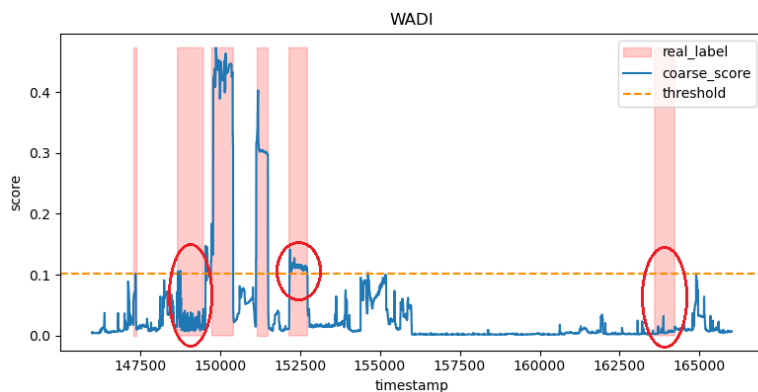
(a) 细粒度模块



(b) 细粒度模块



(c) 粗粒度模块



(d) 粗粒度模块

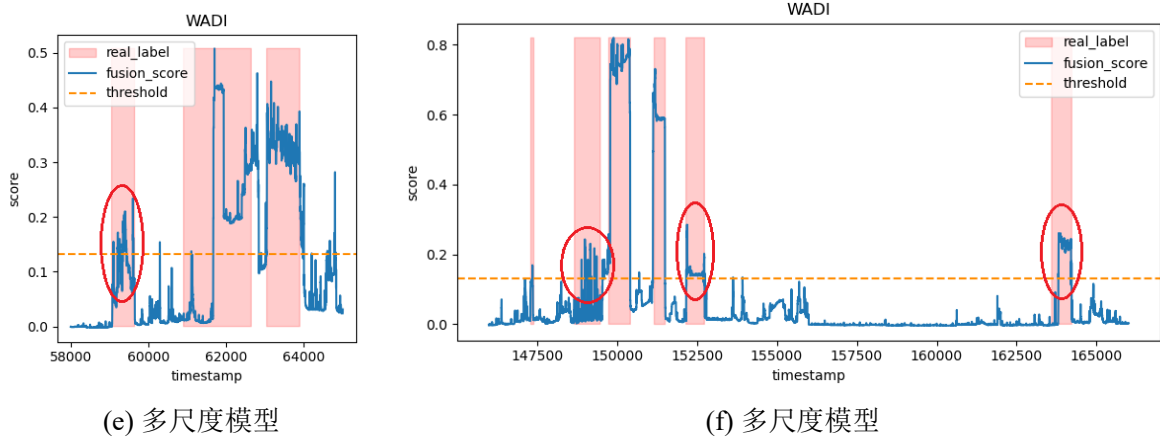


图 3-9 WADI 测试集中的异常分数和真实异常标签

为了更细致的分析，本节选取了 62000 时间步和 155000 时间步附近的异常分数进行分析。根据图3-9可以看到，58000 到 60000 时间步之间的异常，两个模块各自检测出来了该段异常的一部分，经过融合后，综合两个模块的优势部分，更完整地检测出了该段异常。147500 到 150000 之间的异常和 162500 到 165000 时间步之间的异常，粗粒度模块均未检测出或者只检测出很少一部分，而细粒度模块明显检测出了这两段异常，152500 时间步左右的异常则正好相反，细粒度模块只检测出了很少一部分，而粗粒度模块基本全部检测出来。经过融合，这 3 个异常均可以被更完整的检测出来。

3.5.7 不同超参数对模型性能的影响

为了进一步验证模型的效果，在不同超参数条件下验证模型的鲁棒性，同时对比模型在不同超参数下的表现。在本节中将设计不同的超参数，并通过实验来研究这些超参数对 MMTSAD 算法异常检测性能的影响，采用对照试验的方式来比较这些超参数。具体来说，将通过比较不同降采样因子、不同降采样函数来探究它们对模型性能的影响，本节选取上述 5 个数据集中的 WADI、SWaT 和 PSM 进行实验，因为这三个数据集均是单实体数据集，便于展示。

1) 降采样因子对比

降采样因子是决定模型尺度的关键超参数。降采样因子越大，意味着模型会重构更粗粒度的数据，更偏向于检测上下文异常和粗粒度异常。为了研究不同降采样因子对模型性能的影响，本小节将采用不同的降采样因子来训练模型，并比较它们的性能表现，实验中将降采样因子分别设置为 2、4、8、12、16，然后计算对应的 F1* 分数，代表异常检测效果。最终实验结果如图3-10(b)所示。

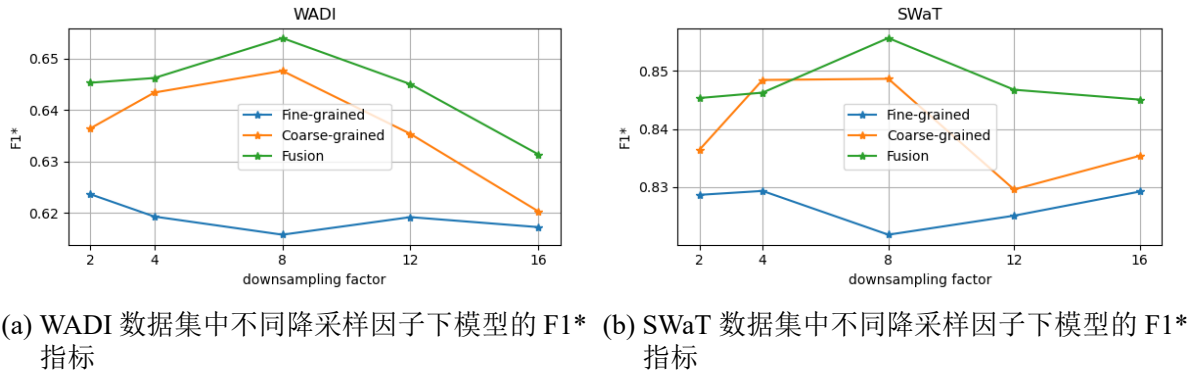


图 3-10 不同降采样因子下模型的 F1* 指标

通过对图3-10分析可知，在 SWaT 数据集上，多尺度异常检测模型在降采样因子等于 8 时得到了最优的 F1* 分数。另外可以看到，粗粒度模块的 F1* 分数（图中橙色曲线）与细粒度模块的 F1* 分数（图中蓝色曲线）总体上较为接近，但经过异常分数融合的多尺度模型的 F1* 分数（图中绿色曲线）在几乎所有降采样因子上都优于以上单个模块。因为数据中包含了不同尺度的异常，不同粒度的模块有利于检测出不同尺度的异常，也有利于对一个异常区间检测更加完整。验证了本章算法使用多个尺度模块组成多尺度异常检测模型的有效性和鲁棒性。

2) 降采样函数对比

为了研究不同降采样函数对模型性能的影响，本小节将采用不同的降采样函数来训练模型，并比较它们的性能表现，实验中将降采样函数分别设置为 mean 求平均值函数、min 求最小值函数、max 求最大值函数、median 求中位数函数，然后计算对应的 F1* 分数，代表异常检测效果。最终实验结果如图3-11所示。

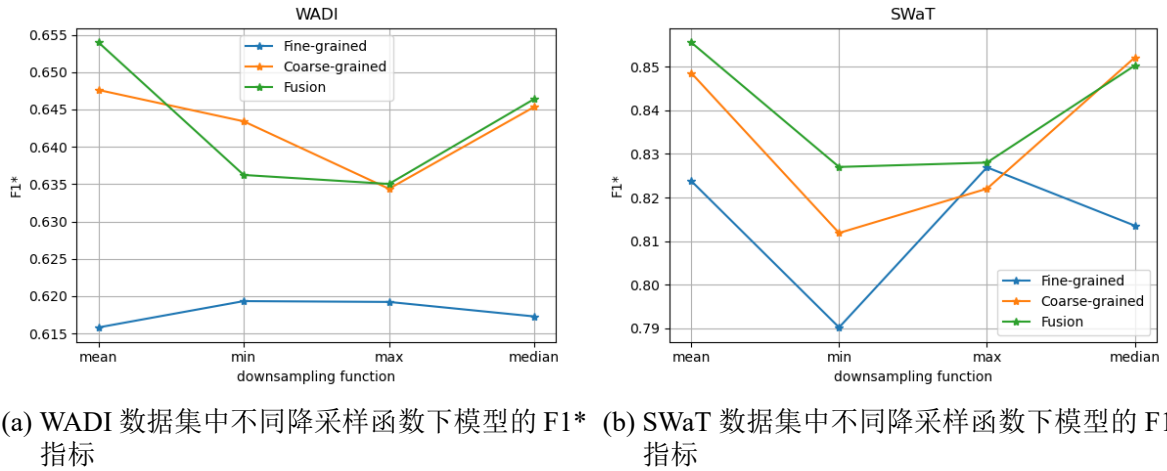


图 3-11 不同降采样函数下模型的 F1* 指标

通过对图3-11分析可知，在两个数据集上，都是 `mean` 函数作为降采样函数取得了最优的 $F1^*$ 值。

3.6 本章小结

本章介绍了一种基于多尺度的无监督多维时序数据异常检测算法——Multiscale Multivariate Time Series Anomaly Detection (MMTSAD)。该算法旨在解决传统异常检测方法在处理长序列多维时间序列数据时的局限性，特别是在同时捕捉不同尺度的异常和处理多维数据中维度相关性方面的不足。

MMTSAD 算法的核心思想是利用原数据作为细粒度数据，并通过降采样操作获得粗粒度数据，同时使用两个滑动窗口处理不同粒度的数据，得到两类子序列作为两个异常检测模块的输入。这样，模型能够在训练中同时学习不同尺度的时间序列分布，并在测试中同时检测不同尺度的异常。

异常检测模块采用基于 Attention 机制的自编码器对子序列数据进行编码和解码，以重建输入数据。其中，粗粒度子序列处理模块中引入了图注意力网络（GAT）以学习多维数据中维度相关性。在检测阶段，提出了一种分布调整的多尺度异常分数融合方法，综合考虑模型在不同尺度模块中计算出的异常分数，得到最终的融合异常分数。

在实验部分，本章在 5 个公开的实际应用数据集上进行了实验，对比了以往的 10 个基准模型，并在其中 4 个数据集上都取得了最优的 $F1^*$ 分数，说明了异常检测模型 MMTSAD 具备优秀的异常检测性能。

总结来说，MMTSAD 算法通过多尺度学习和异常分数融合的方法，有效地提高了无监督多维时序数据异常检测的准确性和鲁棒性，尤其在处理长序列和多维相关性数据方面表现出色。这些特点使得 MMTSAD 算法在实际应用中具有广泛的应用前景。

4 基于 Patch 的多变量时序数据分类模型

4.1 引言

多变量时序数据分类是一项重要的任务，它在许多领域中都具有广泛的应用，如金融、医疗、气象等。然而，由于多变量时序数据的复杂性以及时序数据规模的日益增加，传统的分类方法往往难以有效地处理这些数据。因此，本文提出了一种名为 PatchTSC 的模型，旨在解决多变量时序数据分类任务中的挑战，通过频域信息以低参数量学习多变量时序数据之间的关系，通过 Patch 操作高效处理长序列数据，有效利用 Transformer 编码器的 token，扩大模型可接受的输入序列长度。

模型主要包括三个模块：Patch 模块、变量独立的编码器模块和频域 Transformer 模块。通过以上使用以上模块，可以提升模型对序列信息的学习，减少模型参数规模，同时兼顾对变量间关系模式的学习。在本章中将详细介绍 Patch 模块、变量独立编码器模块和频域 Transformer 模块的设计和实现。

实验部分中，本章的方法在 20 个公开数据集中的 7 个取得了相对于其他方法更好的准确率，在 3 个数据集上取得了次优的准确率。同时本章方法在 20 个数据集上的平均准确率也达到了最优。说明了本章方法在多变量时序数据分类任务上有优秀的性能和泛化性。

4.2 多变量时序数据分类任务的定义

在多变量时间序列分类中，每个样本可以表示为 $\{\mathbf{X}, y\}$ ，其中 \mathbf{X} 代表样本的多变量时序数据， y 代表这个单变量时序数据对应的类别标签，是离散类变量。

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\} \quad \mathbf{x}_t = \{x_t^1, x_t^2, \dots, x_t^D\} \quad (4-1)$$

$$y \in \{1, 2, \dots, C\} \quad (4-2)$$

式中： T ——样本时序数据长度； \mathbf{x}_t ——单个时间点对应的多变量记录； D ——多变量时间序列的变量数量； C ——样本类别数量。

时序数据的分类器的作用是将输入的样本时序数据映射到类别变量值 t 的概率分布。

4.3 基于 Patch 的多变量时序数据分类模型 PatchMTSC

4.3.1 PatchMTSC 模型思路

4.3.2 PatchMTSC 模型结构

针对多变量时序数据分类任务，本章提出了 PatchTSC 模型。模型主要包含 Patch 模块、变量独立编码器模块、频域 Transformer 模块和 softmax 模块四个部分。模型总体架构如图4-1所示。

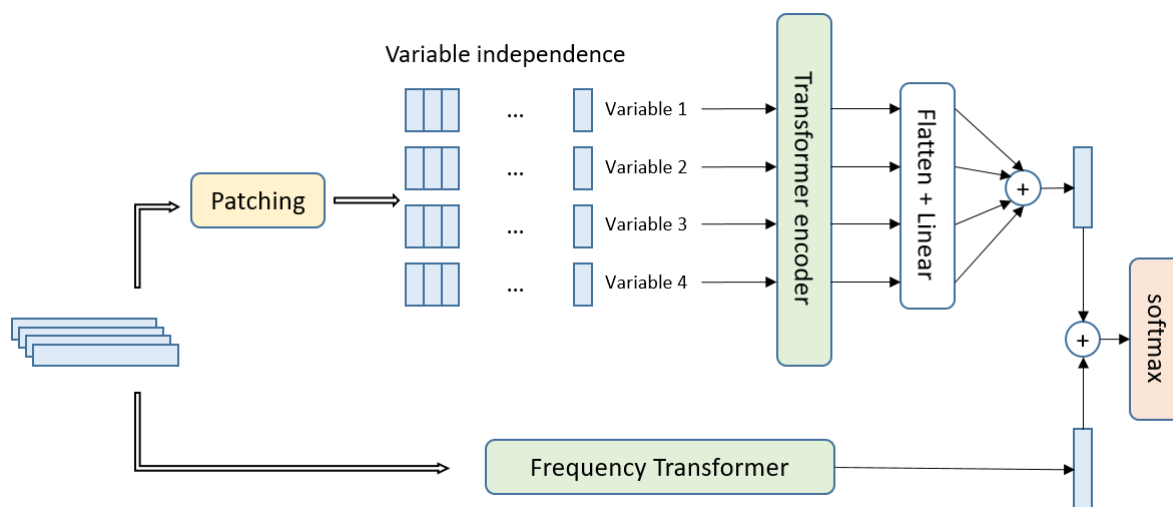


图 4-1 模型总体架构

其中，Patch 模块、变量独立编码器模块和频域 Transformer 模块会在接下来的三个小节中详细介绍。

1) 多变量时序数据 Patching 模块

Ref equations Patch 模块主要解决编码器单个 token 对应的序列信息少的问题，同时显著减少了编码器的输入长度，降低了模型的时间和空间复杂度。以下具体说明 Patch 模块的作用和实现细节。在以往使用 Transformer 编解码器的时序分析模型中，Transformer 模型的一个 token 常常对应一个时间点上的数据，而在自然语言处理领域中，每个词向量对应 Transformer 模型的一个 token。显然时间序列中单个时间点数据不像自然语言处理中的词向量那样具有丰富的信息。为了使 Transformer 中每个 Token 包含更丰富的序列信息，本章将一个 patch 数据作为 Transformer 的输入 token。一个 patch 即代表序列中的一个长为 `patch_len` 的子序列，子序列相比单个点数据包含了更加丰富的序列信息和趋势信息。另一方面，将原始序列划分为 patch 序列，大幅减少了序列长度，降低了后续模块的计算复杂度。Patch 处理流程如图4-2所示。

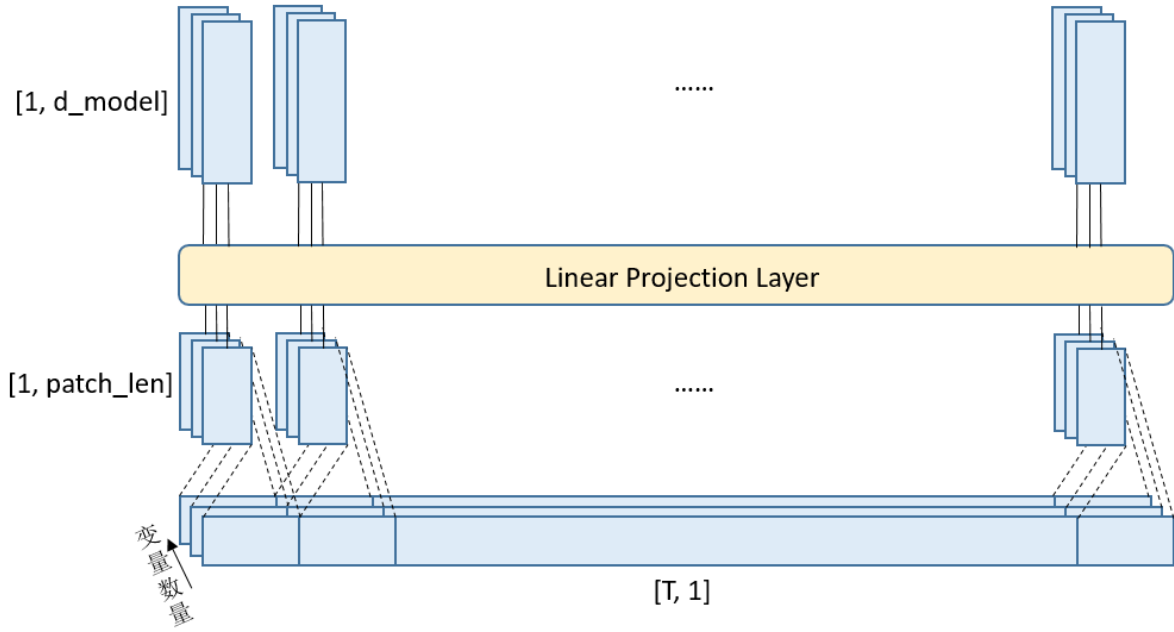


图 4-2 Patch 处理流程

Patch 操作针对每个变量序列独立进行，数据输入维度为 $[batch_size, num_vars, T]$ ，其中 $batch_size$ 是每批次的样本数量，在多变量时序数据分类任务中，每个样本是一个完整的多变量时间序列。 num_vars 代表变量的数量， T 代表完整时间序列的长度，即记录中时间戳的数量。经过 Patching 操作后，输出序列维度为 $[batch_size, num_vars, patch_num, patch_len]$ ，其中 $patch_len$ 是模型超参数， $patch$ 数量可以由公式 $patch_num = \lfloor \frac{T}{patch_len} \rfloor$ 计算得到。此时一个 $patch$ 即是一个长为 $patch_len$ 的张量，代表单变量的一个子序列。接着经过一个共享的线性层将 $patch$ 的映射到新的维度 d_model ，即 Transformer 层的输入维度。最终每个变量序列都会被转换成一个 $patch$ 序列，每个 $patch$ 的维度为 $[1, d_model]$ ， $patch$ 序列长度为 $patch_num$ ，是原始序列长度的 $\frac{1}{patch_len}$ 。因为 Transformer 在时间和空间上的复杂度为 $O(N^2)$ ，其中 N 是输入令 $token$ 的数量。这在实际中成为计算时间和内存的主要瓶颈。所以使用 $patch$ 能显著降低模型参数和空间时间复杂度，提升了模型处理长序列数据的能力。

2) 多变量通道独立编码器模块

Ref equations 本小节采用 Transformer 的 $encoder$ 作为编码器，多变量时间序列是一个多通道信号，Transformer 每个输入 $token$ 可以由来自单个通道或多个通道的数据表示。

通道独立的编码器，计算过程中只处理单个通道的信息，也就是说每个输入 $token$ 只包含来自单个变量的时间序列。并且所有变量序列共享相同的嵌入和 Transformer $encoder$ 权重。

因此 Transformer encoder 可以设置较少的输入维度，减少模型的参数数量。因为通道独立的 transformer encoder 无法捕捉变量间的关系模式。所以本章使用频域 Transformer 模块学习变量之间的关系，频域 Transformer 模块将在下一小节说明。

3) 频域 Transformer 模块

Ref equations 为了捕捉多变量时序数据中变量之间的关系，本章提出了频域 Transformer 模块，用于在频域中学习变量间的模式。在多变量时序数据分类任务中，需要一次处理整个序列的数据，所以从时域角度处理变量间的关系会产生巨大的计算量。大多数时间序列数据在频域表示中信号数量会大幅减少，即具有稀疏的表示，所以本章选择在频域中学习变量间的关系。在频域中可以捕获时间序列中的重要模式，通过捕获的重要模式学习变量间关系和变量频域自身模式。频域 Transformer 模块总体架构如图4-3所示。

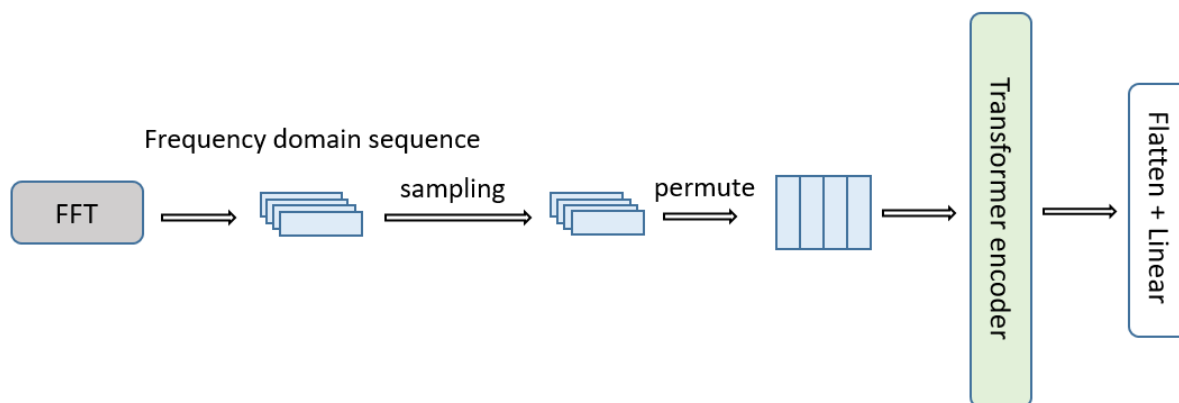


图 4-3 频域 Transformer 模块

在具体实现上，本章通过快速傅里叶变换（FFT）得到原始数据的频域信号，快速傅里叶变换时间复杂度为 $O(L \cdot \log L)$ ， L 为序列长度，相比离散傅里叶变换（DFT） $O(L^2)$ 的时间复杂度，更适合处理长序列数据。

经过傅里叶变换得到的频域数据中会包含不同周期的信号，如果简单地保留所有频率分量可能会导致较差的模型学习效果，因为时间序列数据中的许多高频的信号都是由无规则的噪声引起的。所以本章通过对再通过采样保留其中部分信号，具体是随机选择固定数量的频域信号分量，其中包括部分低频率分量和部分高频率分量，可以更好地表示时间序列。因为只简单保留低频分量或高频分量都可能造成信息遗漏，丢失有助于分类的频域信息。所以选取固定数量的高低频信号有利于在较少的信号数量下保持数据的重要特征，减少频域 Transformer 的参数数量。经过采样的频域信号分量交换维度后输入 Transformer encoder，用于对变量间频域关系和变量内频域关系的学习。

4.4 实验结果与分析

4.4.1 实验环境

多尺度无监督多维时序数据异常检测方法 MMTSAD 的实验环境如表4-1所示。

表 4-1 实验环境表

实验环境	版本或型号
操作系统	Ubuntu 18.04.2 LTS
编程语言	Python 3.7
开发环境与组件	VSCode 1.85.1, Anaconda 4.7.10
相关程序库	Pytorch 1.12.0, numpy 1.21.5, pandas 1.3.5, matplotlib 3.5.2, scikit-learn 1.0.2
CPU	Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz
GPU	NVIDIA Tesla V100-PCIE-32GB

4.4.2 数据集说明

加州大学河滨分校维护的 UCR 数据集^[51]是时间序列数据分析领域最著名的公开数据集。但是 UCR 数据集中所有的数据都是单变量的，并且实际应用中产生的多变量时间序列数据越来越多。基于以上原因，东英吉利大学和加州大学河滨分校在 2018 年发布了多变量的 UEA 公开数据集^[52]。UEA 数据集由 30 个数据集组成，这些数据集来自广泛的领域，为了方便模型训练和测试以及模型之间的对比，维护人员将每个数据集的数据格式化为相同的长度，剔除了包括缺失数据的序列，并且已经预先将数据划分出训练集和测试集。UEA 数据集中主要包括以下领域的数据集：

1) 人类活动识别

人类活动识别 (HAR) 是基于加速度计/陀螺仪数据预测活动 (类值) 的问题。数据是三维或六维坐标。属于该类的数据集如下：**BasicMotions (BM)** 数据集：四名学生戴着智能手表进行了四项活动，分别是站立、步行、跑步和打羽毛球。手表可以收集 3D 加速度计和 3D 陀螺仪的数据。

Cricket 数据集 (CK)：通过在板球裁判手腕上放置加速度计来收集裁判在做出不同动作时的信号，需要依据信号数据区分各种动作。

ERing 数据集 (ER)：根据手指传感器记录的信号区分手部和手指手势。

Handwriting 数据集：根据智能手表记录的加速度值区分手在写 26 个英文字母中的哪一个字母。

Libras 数据集 (LB)：根据视频中手部动作映射出的时序数据区分 15 种手部动作。

NATOPS 数据集 (NT): 根据手、肘部、手腕和拇指上的传感器产生的信号区分海军航空兵指挥信号。

RacketSports 数据集 (RS): 根据智能手表记录的信号区分羽毛球或壁球玩家所做的动作。

UWaveGestureLibrary 数据集 (UGL): 根据加速度计数据区分简单手部手势。

2) 运动的分类

区别于人类活动识别类数据集由加速度计和陀螺仪收集数据, 运动分类数据集由传感器记录空间位移得到。

ArticulatoryWordRecognition (AWR) 数据集: 电磁发音仪记录了多个英语母语人士说 25 个单词时嘴唇的运动数据, 根据该数据判断发音者说了哪个字母。

CharacterTrajectories 数据集 (CT): 根据三维笔尖速度轨迹数据区分 20 个不同的字符。

PenDigits 数据集 (PD): 根据笔尖在数字屏幕上的 x 和 y 坐标数据进行手写阿拉伯数字分类。

3) 心电图 (ECG) 分类

心电图分类是典型的多变量时序数据任务, 但这个问题目前仍较难解决。

AtrialFibrillation 数据集 (AF): 该数据集样本是房颤的 5 秒片段, 包含两个 ECG 信号, 每个信号以每秒 128 个样本的速度采样。分类标签为:n、s 和 t。n 类描述为非终止性心房颤动 (即在原始记录数据后至少 1 小时未终止)。s 级被描述为在记录过程后至少一小步后自行终止的心房颤动。第 t 类被描述为立即终止, 即在记录结束后一秒内终止。

4) 脑电图 (EEG) 脑磁图 (MEG) 分类

EEG 和 MEG 分类在医学, 心理学和人机交互方面有着广泛的应用, 该类型数据集主要来自脑机接口竞赛。

FingerMovements 数据集 (FM): 根据食指和小拇指打字时产生的 EEG 数据区分打字的手指。

MotorImagery 数据集 (MI): 根据大脑右侧记录的 EEG 数据区分受试者在想左手小指的动作还是舌头的动作。

SelfRegulationSCP1 数据集 (SCP1): 根据健康受试者在电脑屏幕上上下移动光标时产生的皮质电位数据, 区分皮质阳性或阴性。

SelfRegulationSCP2 数据集 (SCP2): 与 SelfRegulationSCP1 数据集类似, 但是受试者是 ALS 患者。

FaceDetection 数据集 (FD): 根据受试者的 MEG 数据区分受试者的面部与很模糊的面部。

HandMovementDirection 数据集 (HMD): 两名受试者在听到提示音后, 只用手和手腕在他们选择的四个方向 (右、上、下、左) 中的一个方向移动操纵杆。任务是从活动期间记录的脑磁图 (MEG) 数据中对运动方向进行分类。

5) 其他领域时序数据分类

UEA 数据集中的其他数据集如下:

PEMS-SF 数据集 (PM): 根据高速公路上不同车道的占用率区分当天是一周中的哪一天。

LSST 数据集 (LSST): 根据测量的六个不同的天文滤光器中的光子通量数据区分天体。

UEA 数据集具体信息如表4-2所示。

表 4-2 UEA 数据集统计信息

数据集	训练集样本 数量	测试集样本 数量	变量数量	序列长度	类别数量
ArticularyWordRecognition	275	300	9	144	25
AtrialFibrillation	15	15	2	640	3
BasicMotions	40	40	6	100	4
CharacterTrajectories	1422	1436	3	182	20
Cricket	108	72	6	1197	12
ERing	30	270	4	65	6
FaceDetection	5890	3524	144	62	2
FingerMovements	316	100	28	50	2
HandMovementDirection	160	74	10	400	4
Handwriting	150	850	3	152	26
Libras	180	180	2	45	15
LSST	2459	2466	6	36	14
MotorImagery	278	100	64	3000	2
NATOPS	180	180	24	51	6
PenDigits	7494	3498	2	8	10
PEMS-SF	267	173	963	144	7
RacketSports	151	152	6	30	4
SelfRegulationSCP1	268	293	6	896	2
SelfRegulationSCP2	200	180	7	1152	2
UWaveGestureLibrary	120	320	3	315	8

4.4.3 异常检测结果评价指标与实验设置

1) 异常检测结果评价指标

采用分类精度作为评价指标。分类精度 = 正确分类的样本数/测试集样本总数

2) 实验设置

表4-3显示了在实验数据集上训练 PatchMTSC 所使用的超参数。

表 4-3 模型实验参数

数据集	patch 长度	patch 数量	变量独立编码器输入 维度	变量独立编码器 头数
ArticularyWordRecognition	2	72	4	1
AtrialFibrillation	4	160	8	2
BasicMotions	2	50	4	1
CharacterTrajectories	2	91	4	1
Cricket	8	150	16	4
ERing	1	65	4	1
FaceDetection	1	62	4	1
FingerMovements	1	50	4	1
HandMovementDirection	4	100	8	2
Handwriting	1	152	4	1
Libras	1	45	4	1
LSST	1	36	4	1
MotorImagery	10	300	16	4
NATOPS	1	51	4	1
PenDigits	1	8	4	1
PEMS-SF	2	72	4	1
RacketSports	1	30	4	1
SelfRegulationSCP1	4	224	8	2
SelfRegulationSCP2	8	144	16	4
UWaveGestureLibrary	2	158	4	1

除了以上超参数之外，其他的超参数所有数据集均保持一致，包括优化器为 Adam，学习率 $5e-4$ ，每个批次样本数量为 32。

4.4.4 与其他算法模型的对比结果

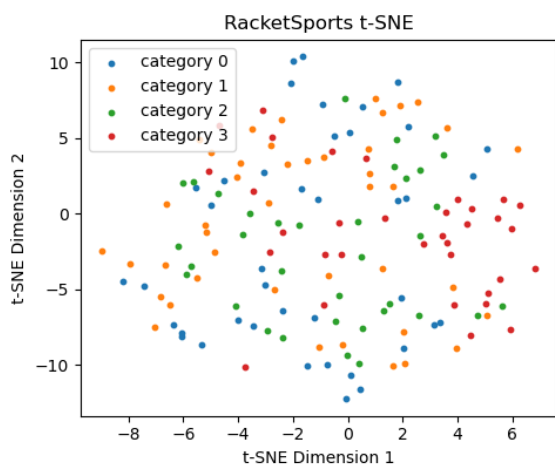
PatchMTSC 在上述公开数据集上的表现如表4-5所示，表中 N/A 代表该方法无法对应数据集进行计算，也不会参与平均准确率的计算。

表 4-4 模型分类准确率

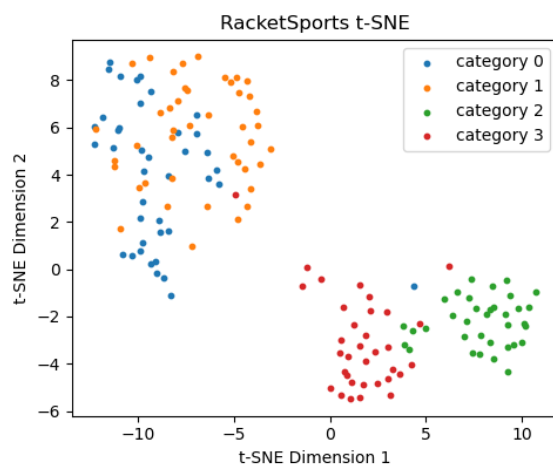
数据集\方法	DTWI ^[53]	DTWD ^[53]	WM ^[54]	ShapeNet ^[55]	ROCKET ^[56]	TST ^[57]	SVP-T ^[29]	Ours
AWR	0.980	0.987	0.990	0.987	0.996	0.983	<u>0.993</u>	0.983
AF	0.267	0.220	0.333	0.400	0.249	0.200	0.400	0.400
BM	1.000	0.975	1.000	1.000	0.990	0.975	1.000	1.000
CT	0.969	0.989	0.990	0.980	N/A	N/A	0.990	0.986
CK	0.986	1.000	1.000	0.986	1.000	0.958	1.000	0.972
ER	0.914	0.929	0.133	0.133	0.981	0.933	0.937	<u>0.944</u>
FD	0.000	0.529	0.545	0.602	<u>0.694</u>	0.681	0.512	0.709
FM	0.520	0.530	0.490	0.580	0.553	0.776	0.600	<u>0.730</u>
HMD	0.306	0.231	0.365	0.338	0.446	0.608	0.392	<u>0.581</u>
HW	0.316	0.286	0.605	0.452	<u>0.567</u>	0.305	0.433	0.400
LB	<u>0.894</u>	0.870	0.878	0.856	0.906	0.844	0.883	0.877
LSST	0.575	0.551	0.590	0.590	<u>0.632</u>	0.381	0.666	0.626
MI	N/A	0.500	0.500	0.610	0.531	N/A	<u>0.650</u>	0.660
NT	0.850	0.883	0.870	0.883	0.885	0.900	<u>0.906</u>	0.911
PD	0.939	0.977	0.948	0.977	0.996	0.974	<u>0.983</u>	0.945
PM	0.734	0.711	0.000	0.751	0.856	0.919	<u>0.867</u>	0.855
RS	0.842	0.803	0.934	0.882	<u>0.928</u>	0.829	0.842	0.802
SCP1	0.765	0.775	0.710	0.782	0.866	<u>0.925</u>	0.884	0.931
SCP2	0.533	0.539	0.460	0.578	0.514	0.589	<u>0.600</u>	0.616
UGL	0.868	0.903	0.916	0.906	0.944	0.903	<u>0.941</u>	0.856
平均准确率	0.697	0.709	0.662	0.713	0.764	0.760	<u>0.773</u>	0.789
最优数量	1	1	5	2	<u>6</u>	3	5	7

由表4-5可以看到，本章的方法在 20 个公开数据集中的 7 个取得了相对于其他方法更好的准确率，在 3 个数据集上取得了次优的准确率。同时本章方法在 20 个数据集上的平均准确率也达到了最优。说明了本章方法在多变量时序数据分类任务上有优秀的性能和泛化性。

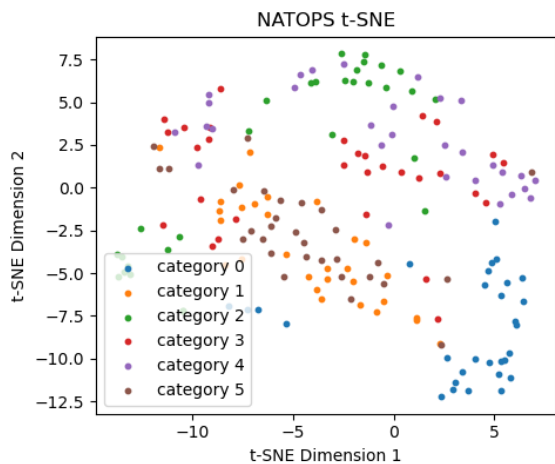
4.4.5 t-SNE 可视化实验



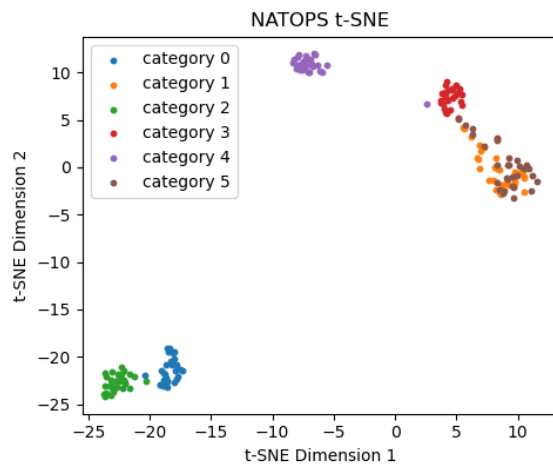
(a) RS 数据集训练前 t-SNE 可视化



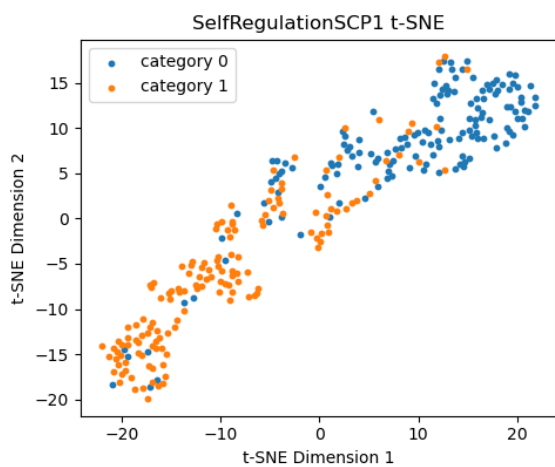
(b) RS 数据集训练后 t-SNE 可视化



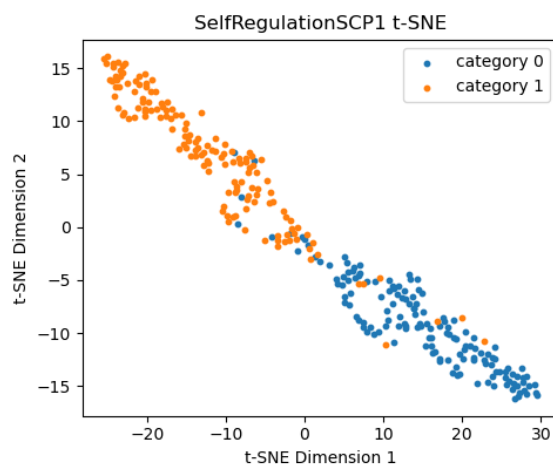
(c) NT 数据集训练前 t-SNE 可视化



(d) NT 数据集训练后 t-SNE 可视化

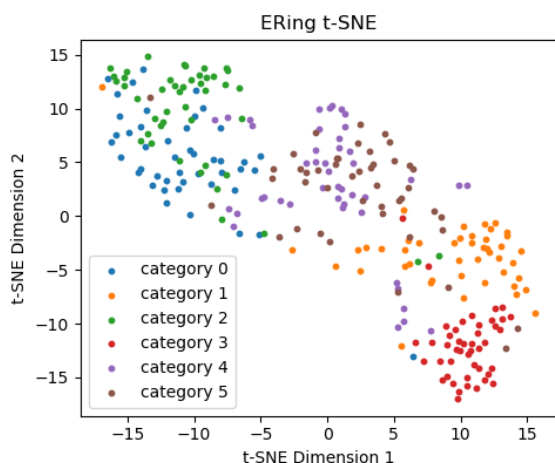


(e) SCP1 数据集训练前 t-SNE 可视化

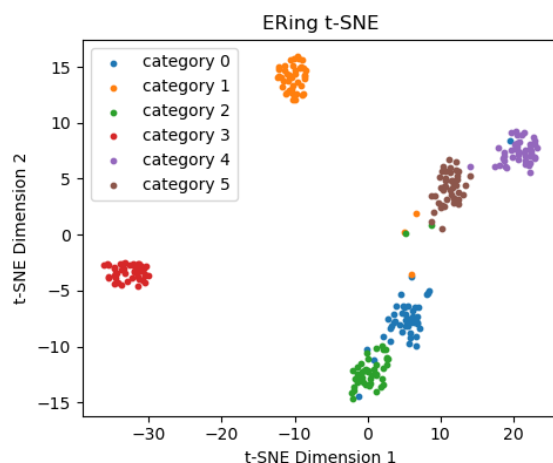


(f) SCP1 数据集训练后 t-SNE 可视化

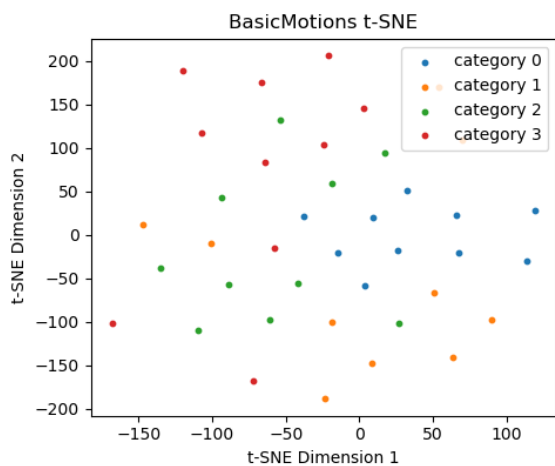
图 4-4 t-SNE 可视化



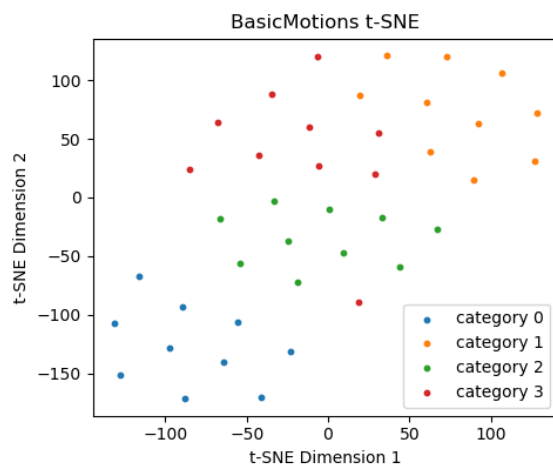
(a) ER 数据集训练前 t-SNE 可视化



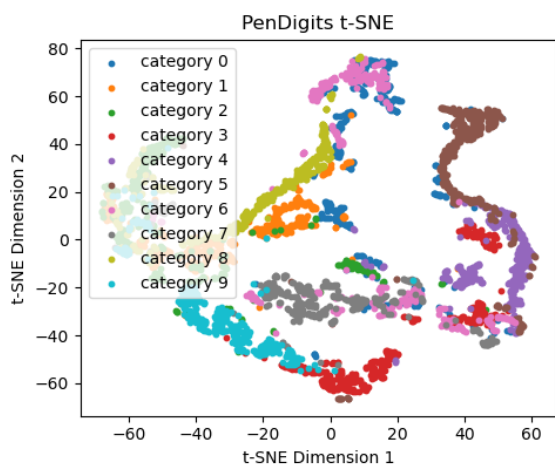
(b) ER 数据集训练后 t-SNE 可视化



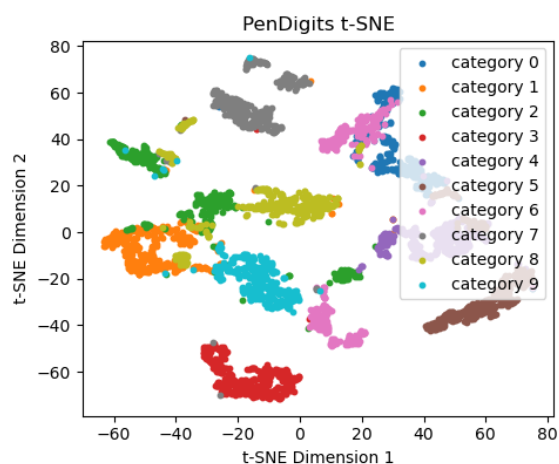
(c) BM 数据集训练前 t-SNE 可视化



(d) BM 数据集训练后 t-SNE 可视化



(e) PD 数据集训练前 t-SNE 可视化



(f) PD 数据集训练后 t-SNE 可视化

图 4-5 t-SNE 可视化

4.4.6 模型消融实验

PatchMTSC 在上述公开数据集上的表现如表4-5所示。

表 4-5 模型分类准确率

数据集\方法	w/o 频域 Transformer	Ours(PatchMTSC)
ArticularyWordRecognition	0.980	0.983
AtrialFibrillation	0.400	0.400
BasicMotions	1.000	1.000
CharacterTrajectories	0.980	0.986
Cricket	0.958	0.972
ERing	0.929	0.944
FaceDetection	0.681	0.709
FingerMovements	0.600	0.730
HandMovementDirection	0.576	0.581
Handwriting	0.400	0.400
Libras	0.844	0.877
LSST	0.590	0.626
MotorImagery	0.660	0.660
NATOPS	0.883	0.911
PenDigits	0.939	0.945
PEMS-SF	0.855	0.855
RacketSports	0.803	0.802
SelfRegulationSCP1	0.925	0.931
SelfRegulationSCP2	0.600	0.616
UWaveGestureLibrary	0.856	0.856

4.4.7 不同频域采样方法对模型性能的影响

因此本节以实验方式研究了随机采样、高频分量中随机采样、低频分量中随机采样和不采用这四种方式对模型分类性能的影响。实验选择 UEA 数据集中的 FaceDetection、MotorImagery 和 NATOPS 数据集进行实验，其结果如表4-6所示，实验结果对应的柱状图如图4-6所示。

表 4-6 采样方式对比实验

采样方法\数据集	FD	MI	NT	SCP1
随机采样	0.709	0.660	<u>0.911</u>	<u>0.931</u>
高频分量随机采样	0.695	0.640	0.888	0.915
低频分量随机采样	0.703	<u>0.650</u>	0.894	0.919
不采样	<u>0.706</u>	<u>0.650</u>	0.916	0.933

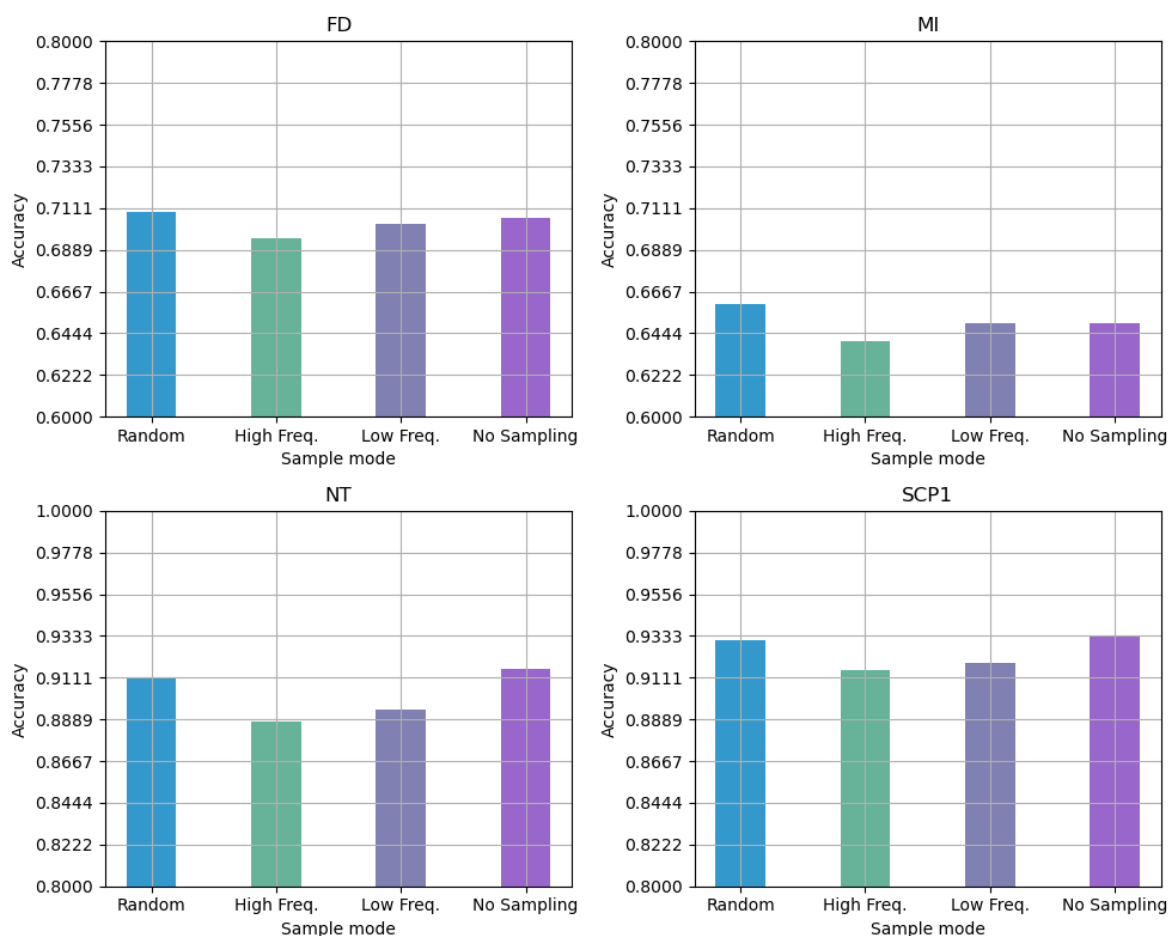


图 4-6 采样方式对比实验

4.4.8 不同超参数对模型性能的影响

为了进一步验证模型的效果，在不同超参数条件下验证模型的鲁棒性，同时对模型在不同超参数下的表现。在本节中将设计不同的超参数，并通过实验来研究这些超参数对 MMTSAD 算法异常检测性能的影响，采用对照试验的方式来比较这些超参数。具体来说，将通过比较不同降采样因子、不同降采样函数来探究它们对模型性能的影响，本节选取上述 5 个数据集中的 WADI、SWaT 和 PSM 进行实验，因为这三个数

数据集均是单实体数据集，便于展示。

1) 不同 patch 长度的对比

降采样因子是决定模型尺度的关键超参数。降采样因子越大，意味着模型会重构更粗粒度的数据，更偏向于检测上下文异常和粗粒度异常。为了研究不同降采样因子对模型性能的影响，本小节将采用不同的降采样因子来训练模型，并比较它们的性能表现，实验中将降采样因子分别设置为 2、4、8、12、16，然后计算对应的 F1* 分数，代表异常检测效果。最终实验结果如图所示。patch_len

2) 不同编码器输入维度的对比

4.5 本章小结

本章节详细介绍了一种针对多变量时序数据分类任务的算法模型 PatchTSC。该模型针对多变量时序数据的复杂性和规模增长带来的挑战，提出了一种新颖的解决方案。PatchTSC 模型主要由三个模块组成：Patch 模块、变量独立编码器模块和频域 Transformer 模块。其中，Patch 模块通过将长序列划分为包含更多信息的子序列，提高了序列处理的效率，并减少了模型的参数量。变量独立编码器模块利用 Transformer 的 encoder 结构，独立处理每个变量的序列信息。频域 Transformer 模块则在频域中捕捉变量间的关系模式，有效利用了时间序列数据在频域中的稀疏特性。实验结果表明，PatchTSC 模型在 20 个公开数据集中的 7 个上取得了最佳准确率，在 3 个数据集上取得了次优的准确率，且在所有数据集上的平均准确率达到最优。这些结果证明了 PatchTSC 模型在处理多变量时序数据分类任务时的高性能和良好的泛化能力。

5 结论与展望

5.1 结论

现有的异常检测方法主要是采用固定长度的滑动窗口从观测数据中截取窗口数据，并逐个对窗口数据进行深度学习的编码训练和异常检测。然而随着实际生产中时序数据的采集频率越来越高，使用单一固定长度的窗口数据进行训练将更加难以学习到时间序列的长期模式和不同尺度的异常。同时因为自注意力机制的时间复杂度和空间复杂度都随时间序列的长度 L 呈二次增长，所以单纯加长滑动窗口长度会使模型训练的时间消耗和内存占用迅速增长。另外因为多维时序数据可能出现的模式的复杂性和多样性，要求模型可以检测出不同尺度的异常。由于现有的异常检测方法在实际应用中存在以上的缺陷，针对上述问题，MMTSAD 方法以原数据作为细粒度数据并使用降采样操作处理多维时序数据获得粗粒度数据，同时使用两个滑动窗口处理不同粒度的数据得到两类子序列，作为两个异常检测模块的输入。这样方便模型在训练中同时学习不同尺度的时间序列分布，在测试中同时检测不同尺度的异常。异常检测模块采用基于 Attention 机制的自编码器对子序列数据进行编码和解码，以重建输入数据。其中用于处理粗粒度子序列的模块中引入了 GAT[14] 以学习多维数据中维度相关性。在检测阶段，本文提出了一种分布调整的多尺度异常分数融合方法，用于综合考虑模型在不同尺度模块中计算出的异常分数，得到最终的融合异常分数。MMTSAD 模型在 5 个公开的实际应用数据集上进行了实验，对比了以往的 10 个基准模型，并在其中 4 个数据集上都取得了最优的 F1* 分数，说明了异常检测模型 MMTSAD 具备优秀的异常检测性能。总结来说，MMTSAD 算法通过多尺度学习和异常分数融合的方法，有效地提高了无监督多维时序数据异常检测的准确性和鲁棒性

多变量时序数据分类是一项重要的任务，它在许多领域中都具有广泛的应用，如金融、医疗、气象等。然而，由于多变量时序数据的复杂性以及时序数据规模的日益增加，传统的分类方法往往难以有效地处理这些数据。因此，本文提出了一种名为 PatchTSC 的模型，旨在解决多变量时序数据分类任务中的挑战，通过频域信息以低参数量学习多变量时序数据之间的关系，通过 Patch 操作高效处理长序列数据，有效利用 Transformer 编码器的 token，扩大模型可接受的输入序列长度。模型主要包括三个模块：Patch 模块、变量独立的编码器模块和频域 Transformer 模块。通过以上使用以上模块，可以提升模型对序列信息的学习，减少模型参数规模，同时兼顾对变量间关系模式的学习。在本章中将详细介绍 Patch 模块、变量独立编码器模块和频域 Transformer 模块的设计和实现。PatchMTSC 模型在 20 个公开数据集中的 7 个取得了相对于其他方法更好的准确率，在 3 个数据集上取得了次优的准确率。同时本章方法在 20 个数据集上的平均准确

率也达到了最优。说明了模型在多变量时序数据分类任务上有优秀的性能和泛化性。

5.2 展望

本文针对无监督时序数据异常检测任务提出的 MMTSAD 模型和对多变量时序数据异常检测任务提出的 PatchMTSC 模型已经取得了初步成果，但针对这两种算法仍然存在一些值得改进和探索的地方：

（1）当前 MMTSAD 包含细粒度和粗粒度两个模块，如果数据集规模较大，异常情况更加复杂，可能需要更多的模块才能有效进行异常检测。因此在实际应用中可以考虑设计动态添加更多尺度模块来满足不同的异常检测需求。

（2）当前少部分数据集本身是由时序数据的频谱生成而来，这样的数据集就不再适用于 PatchMTSC，因为序列的语义已经发生了变化。因此可以研究对于时序数据生成的频谱序列如何进行分类，比如使用 Transformer 编码器时不在添加位置编码让模型不再学习序列先后信息等。

致 谢

致谢中主要感谢导师和对论文工作有直接贡献和帮助的人士和单位。一般致谢的内容有：

- (一) 对指导或协助指导完成论文的导师；
- (二) 对国家自然科学基金、资助研究工作的奖学金基金、合同单位、资助或支持的企业、组织或个人；
- (三) 对协助完成研究工作和提供便利条件的组织或个人；
- (四) 对在研究工作中提出建议和提供帮助的人；
- (五) 对给予转载和引用权的资料、图片、文献、研究思想和设想的所有者；
- (六) 对其他应感谢的组织和个人。

致谢言语应谦虚诚恳，实事求是。字数不超过 1000 汉字

用于双盲评审的论文，此页内容全部隐去。

参考文献

- [1] Shyu M L, Chen S C, Sarinnapakorn K, et al. A novel anomaly detection scheme based on principal component classifier[J]. Proceedings of the IEEE foundations and new directions of data mining workshop, 2003: 172-179.
- [2] Angiulli F, Pizzuti C. Fast outlier detection in high dimensional spaces[C]//European conference on principles of data mining and knowledge discovery. 2002: 15-27.
- [3] Breunig M M, Kriegel H P, Ng R T, et al. LOF: identifying density-based local outliers[C]//Proceedings of the 2000 ACM SIGMOD international conference on Management of data. 2000: 93-104.
- [4] Schölkopf B, Platt J C, Shawe-Taylor J, et al. Estimating the support of a high-dimensional distribution[J]. Neural computation, 2001, 13(7): 1443-1471.
- [5] Munir M, Siddiqui S A, Dengel A, et al. DeepAnT: A deep learning approach for unsupervised anomaly detection in time series[J]. Ieee Access, 2018, 7: 1991-2005.
- [6] Ren HS, Xu BX, Wang YJ, et al. Time-series anomaly detection service at microsoft[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 3009-3017.
- [7] He YD, Zhao JB. Temporal convolutional networks for anomaly detection in time series[C]//Journal of Physics: Conference Series: vol. 1213: 4. 2019: 042050.
- [8] Wu WT, He LG, Lin WW, et al. Developing an unsupervised real-time anomaly detection scheme for time series with multi-seasonality[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(9): 4147-4160.
- [9] Shen LF, Li ZC, Kwok J. Timeseries anomaly detection using temporal hierarchical one-class network[C]//Advances in Neural Information Processing Systems(NIPS): vol. 33. 2020: 13016-13026.
- [10] Abdulaal A, Liu, ZH, Lancewicki T. Practical approach to asynchronous multivariate time series anomaly detection and localization[C]//Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021: 2485-2494.
- [11] Zhang YX, Wang JD, Chen YQ, et al. Adaptive memory networks with self-supervised learning for unsupervised anomaly detection[J]. IEEE Transactions on Knowledge and Data Engineering, 2022.
- [12] Li ZH, Zhao YJ, Han JQ, et al. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding[C]//Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021: 3220-3230.
- [13] KIPF T N, WELING M. Semi-Supervised Classification with Graph Convolutional Networks[C]//International Conference on Learning Representations. 2016.
- [14] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.
- [15] Deng AL, Hooi B. Graph neural network-based anomaly detection in multivariate time series[C]//Proceedings of the AAAI conference on artificial intelligence: vol. 35: 5. 2021: 4027-4035.

-
- [16] Zhao H, Wang YJ, Duan JY, et al. Multivariate time-series anomaly detection via graph attention network[C]//2020 IEEE International Conference on Data Mining (ICDM). 2020: 841-850.
- [17] Chen ZK, Chen DS, Zhang X, et al. Learning graph structures with transformer for multivariate time-series anomaly detection in IoT[J]. IEEE Internet of Things Journal, 2021, 9(12): 9179-9189.
- [18] Xu JH, Wu HX, Wang JM, et al. Anomaly transformer: Time series anomaly detection with association discrepancy[C]//International Conference on Learning Representations. 2021.
- [19] Tuli S, Casale G, Jennings N R. Tranad: Deep transformer networks for anomaly detection in multivariate time series data[J]. Proceedings of the VLDB Endowment, 2022, 15(6): 1201-1214.
- [20] Li YF, Peng XY, Zhang J, et al. Dct-gan: Dilated convolutional transformer-based gan for time series anomaly detection[J]. IEEE Transactions on Knowledge and Data Engineering, 2021.
- [21] KARIM F, MAJUMDAR S, DARABI H, et al. Multivariate LSTM-FCNs for time series classification[J]. Neural networks, 2019, 116: 237-245.
- [22] HAO Y, CAO H. A new attention mechanism to classify multivariate time series[C]//Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. 2020.
- [23] ZHANG X, GAO Y, LIN J, et al. Tapnet: Multivariate time series classification with attentional prototypical network[C]//Proceedings of the AAAI conference on artificial intelligence: vol. 34: 04. 2020: 6845-6852.
- [24] LI G, CHOI B, XU J, et al. Shapenet: A shapelet-neural network approach for multivariate time series classification[C]//Proceedings of the AAAI conference on artificial intelligence: vol. 35: 9. 2021: 8375-8383.
- [25] GAO G, GAO Q, YANG X, et al. A reinforcement learning-informed pattern mining framework for multivariate time series classification[C]//31st International Joint Conference on Artificial Intelligence (IJCAI). 2022.
- [26] RUSSWURM M, KÖRNER M. Self-attention for raw optical satellite time series classification[J]. ISPRS journal of photogrammetry and remote sensing, 2020, 169: 421-435.
- [27] LIU M, REN S, MA S, et al. Gated transformer networks for multivariate time series classification [J]. arXiv preprint arXiv:2103.14438, 2021.
- [28] CHENG M, LIU Q, LIU Z, et al. Formertime: Hierarchical multi-scale representations for multivariate time series classification[C]//Proceedings of the ACM Web Conference 2023. 2023: 1437-1445.
- [29] ZUO R, LI G, CHOI B, et al. SVP-T: a shape-level variable-position transformer for multivariate time series classification[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 37: 9. 2023: 11497-11505.
- [30] Yairi T, Takeishi N, Oda T, et al. A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction[J]. IEEE Transactions on Aerospace and Electronic Systems, 2017, 53(3): 1384-1401.
- [31] ZONG B, SONG Q, MIN M R, et al. Deep autoencoding gaussian mixture model for unsupervised anomaly detection[C]//International conference on learning representations. 2018.
- [32] Ruff L, Vandermeulen R, Goernitz N, et al. Deep one-class classification[C]//International conference on machine learning. 2018: 4393-4402.

- [33] Garg A, Zhang WY, Samaran J, et al. An evaluation of anomaly detection and diagnosis in multivariate time series[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(6): 2508-2517.
- [34] Park D, Hoshi Y, Kemp C C. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder[J]. IEEE Robotics and Automation Letters, 2018, 3(3): 1544-1551.
- [35] SU Y, ZHAO Y, NIU C, et al. Robust anomaly detection for multivariate time series through stochastic recurrent neural network[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 2828-2837.
- [36] Audibert J, Michiardi P, Guyard F, et al. Usad: Unsupervised anomaly detection on multivariate time series[C]//Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 2020: 3395-3404.
- [37] Hundman K, Constantinou V, Laporte C, et al. Detecting spacecraft anomalies using lstms and non-parametric dynamic thresholding[C]//Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018: 387-395.
- [38] Ding N, Gao HB, Bu HY, et al. Multivariate-time-series-driven real-time anomaly detection based on bayesian network[J]. Sensors, 2018, 18(10): 3367.
- [39] Gugulothu N, Malhotra P, Vig L, et al. Sparse neural networks for anomaly detection in high-dimensional time series[C]//AI4IOT workshop in conjunction with ICML, IJCAI and ECAI. 2018: 1551-3203.
- [40] YE L, KEOGH E. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification[J]. Data mining and knowledge discovery, 2011, 22: 149-182.
- [41] KARLSSON I, PAPAPETROU P, BOSTRÖM H. Generalized random shapelet forests[J]. Data mining and knowledge discovery, 2016, 30: 1053-1085.
- [42] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
- [43] Ahmed C M, Palleti V R, Mathur A P. WADI: a water distribution testbed for research in the design of secure cyber physical systems[C]//Proceedings of the 3rd international workshop on cyber-physical systems for smart water networks. 2017: 25-28.
- [44] Goh J, Adepu S, Junejo K N, et al. A dataset to support research in the design of secure water treatment systems[C]//Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, October 10–12, 2016, Revised Selected Papers 11. 2017: 88-99.
- [45] Abdulaal A, Liu ZH, Lancewicki T. Practical approach to asynchronous multivariate time series anomaly detection and localization[C]//Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021: 2485-2494.
- [46] Entekhabi D, Njoku E G, O’neill P E, et al. The soil moisture active passive (SMAP) mission[J]. Proceedings of the IEEE, 2010, 98(5): 704-716.
- [47] Xu HW, Chen WX, Zhao NW, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications[C]//Proceedings of the 2018 world wide web conference. 2018: 187-196.

-
- [48] Schlegel U, Arnout H, El-Assady M, et al. Towards a rigorous evaluation of XAI methods on time series[C]//2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). 2019: 4197-4201.
- [49] Lai C Y, Sun F K, Gao ZQ, et al. Nominality Score Conditioned Time Series Anomaly Detection by Point/Sequential Reconstruction[C]//Thirty-seventh Conference on Neural Information Processing Systems. 2023.
- [50] ZHANG C, SONG D, CHEN Y, et al. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data[C]//Proceedings of the AAAI conference on artificial intelligence: vol. 33: 01. 2019: 1409-1416.
- [51] DAU H A, BAGNALL A, KAMGAR K, et al. The UCR time series archive[J]. IEEE/CAA Journal of Automatica Sinica, 2019, 6(6): 1293-1305.
- [52] BAGNALL A, DAU H A, LINES J, et al. The UEA multivariate time series classification archive, 2018[J]. arXiv preprint arXiv:1811.00075, 2018.
- [53] BAGNALL A, LINES J, BOSTROM A, et al. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances[J]. Data mining and knowledge discovery, 2017, 31: 606-660.
- [54] SCHÄFER P, LESER U. Multivariate time series classification with WEASEL+ MUSE[J]. arXiv preprint arXiv:1711.11343, 2017.
- [55] CHANG A X, FUNKHOUSER T, GUIBAS L, et al. Shapenet: An information-rich 3d model repository[J]. arXiv preprint arXiv:1512.03012, 2015.
- [56] DEMPSTER A, PETITJEAN F, WEBB G I. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels[J]. Data Mining and Knowledge Discovery, 2020, 34(5): 1454-1495.
- [57] ZERVEAS G, JAYARAMAN S, PATEL D, et al. A transformer-based framework for multivariate time series representation learning[C]//Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021: 2114-2124.
- [58] WANG C, ZHUANG Z, QI Q, et al. Drift doesn't Matter: Dynamic Decomposition with Diffusion Reconstruction for Unstable Multivariate Time Series Anomaly Detection[C]//Thirty-seventh Conference on Neural Information Processing Systems. 2023.
- [59] Jhin S Y, Lee J H, Park N. Precursor-of-Anomaly Detection for Irregular Time Series[C]//Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023: 917-929.
- [60] Song J H, Kim K W, Oh J, et al. MEMTO: Memory-guided Transformer for Multivariate Time Series Anomaly Detection[C]//Thirty-seventh Conference on Neural Information Processing Systems. 2023.

攻读学位期间取得的研究成果

[1]

用于双盲评审的论文，只列出已发表的学术论文的篇名、发表刊物名称，必须隐去各类论文检索号、期号、卷号、页码；专利号；日期等。

答辩委员会会议决议

论文取得的主要创新性成果包括：

论文工作表明作者在 ××××× 具有 ××××× 知识,具有 ××××× 能力,论文 ××××××××××, 答辩 ××××××××××××××××。

答辩委员会表决, (× 票/一致) 同意通过论文答辩, 并建议授予 ××× (姓名) ××× (门类) 学博士/硕士学位。

常规评阅人名单

本学位论文共接受 7 位专家评阅，其中常规评阅人 5 名，名单如下：

XXX	教授	西安交通大学
XXX	教授	西安交通大学
XXX	副教授	西安交通大学
XXX	研究员	西安交通大学
XXX	高工	西安交通大学