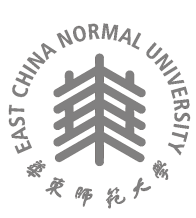
2022 届研究生硕士学位论文

分类号：_____ 学校代码：10269

密　级：_____ 学　　号：51194506010



華東師範大學

East China Normal University

硕士学位论文

MASTER'S DISSERTATION

论文题目：多模态时间序列异常检测研究

院　　系:	**计算机科学与技术学院**
专　　业:	**计算机科学与技术**
研究方向:	**模式识别与机器学习**
指导教师:	**孙仕亮 教授**
学位申请人:	**丁超越**

2022 年 5 月

Thesis of Master's Degree in 2022 School Code: 10269

Student Number: 51194506010

East China Normal University

Title: Research on Multimodal Time Series Anomaly Detection

Department:	School of Computer Science and Technology
Major:	Computer Science and Technology
Research Area:	Pattern Recognition and Machine Learning
Supervisor:	Prof. Shiliang Sun
Candidate:	Chaoyue Ding

May, 2022

丁超越 硕士学位论文答辩委员会成员名单

姓 名	职 称	单 位	备 注
岳晓冬	副教授	上海大学	主席
续晋华	副教授	华东师范大学计算机科学与技术学院	
王峰	副研究员	华东师范大学计算机科学与技术学院	

摘　要

时间序列异常检测旨在从时间序列数据中识别异常模式。长期以来，时间序列异常检测一直是一个重要的研究领域。随着时序中模态数量的增长，时序的复杂程度以及异常检测的难度都会逐步增大。本文针对三种不同复杂程度的数据类型，即单个模态、两个模态、多个模态（大于等于三个模态），依次提出了三种不同的时序异常检测框架，用于有效利用不同类型数据中的信息。针对单个模态数据集上时序的概念漂移问题，本文提出基于概念漂移检测的在线 Transformer 模型。在两个模态的数据集上，本文探索在基于语音和文本的多模态对抗攻击异常样本上的异常检测方法。在多个模态的数据集上，本文设计了一个多模态空间-时间图注意网络，它采用一个多模态图注意网络和一个时间卷积网络来捕获多模态时间序列中的空间-时间相关性。

首先，为了解决单个模态时序数据中的概念漂移问题，我们提出结合了概念漂移检测模块（CDAM）和在线学习的 Transformer 异常检测模型。CDAM 模块负责动态地调整模型的学习率。CDAM 和在线学习共同促进了在线稀疏 Transformer 将知识从旧概念数据的模型转移到新概念数据的模型中。此外，由于 Transformer 中自注意力的时间复杂度较高，我们设计了根方稀疏自注意力来代替原来的自注意力，大大降低了其计算复杂度。

其次，为了更好地挖掘两个模态数据中的信息，我们提出了一个多模态深度融合 Transformer，称为 MDFT。具体来说，音频和文本特征分别由音频和文本编码器提取。我们设计多模态注意机制，以捕获音频和语言特征之间的互补信息，并获得联合的多模态表示。该表示接着被传播到密集层以产生检测结果。

最后，为了明确地捕获多个模态时序数据中的单变量时序之间的空间-时间关系。我们提出了一个多模态的空间-时间图注意网络（MST-GAT）。MST-GAT 首先采用了一个多模态图注意网络（M-GAT）和一个时间卷积网络来捕获多模态时间序列中的空间-时间相关性。具体来说，M-GAT 使用一个多头注意模块和两个关系注意模块（即模态内和模态间注意）来明确地建模模态相关性。此外，MST-GAT 使用重构和预测模块来联合优化模型参数。

关键词：　多模态学习，异常检测，时间序列，图注意力网络，Transformer

ABSTRACT

Time series anomaly detection aims to identify anomalous patterns from time series data. Time series anomaly detection has been an important research area for a long time. As the number of modalities in a time series grows, the complexity of the time series and the difficulty of anomaly detection gradually increase. In this paper, three time series anomaly detection frameworks are proposed for three data types with different complexity levels, i.e., single modality, two modalities, and multiple modalities (greater than or equal to three modalities), to effectively leverage the information in different types of data. For the concept drift problem of time series on a single modal dataset, this paper proposes an online Transformer model based on concept drift detection. On two modal datasets, this paper explores anomaly detection methods for anomalous samples based on multimodal adversarial attacks on speech and text. On a multimodal dataset, this paper designs a multimodal spatial-temporal graph attention network, which employs a multimodal graph attention network and a temporal convolutional network to capture the spatial-temporal correlation in multimodal time series.

First, to solve the concept drift problem in single modal time series, we propose a Transformer anomaly detection model that combines a concept drift detection module (CDAM) and online learning. The CDAM module is responsible for dynamically adjusting the learning rate of the model. The CDAM and online learning together facilitate the transfer of knowledge from the model of old concept data to the model of new concept data by an online sparse Transformer. In addition, owing to the high time complexity of self-attention in the Transformer, we design root-squared sparse self-attention to replace the standard self-attention, which greatly reduces its computational complexity.

Second, to better mine the information in the two modal data, we propose a multimodal deep fusion Transformer, termed MDFT. Specifically, audio and text features are extracted by audio and text encoders, respectively. We design the multimodal attention mechanism to capture the complementary information between audio and speech features and obtain a joint multimodal representation. This representation is then propagated to the dense layer to produce detection results.

Finally, to explicitly capture the spatial-temporal relationships between univariate time series of multiple modalities. We propose a multimodal spatial-temporal graph attention network (MST-GAT). MST-GAT first employs a multimodal graph attention network (M-GAT) and a temporal convolutional network to capture the spatial-temporal correlations in multimodal time series. Specifically, M-GAT uses a multi-head attention module and two relational attention modules (i.e., intra-modal attention and inter-modal attention) to explicitly model modal correlations. Furthermore, MST-GAT uses reconstruction and prediction modules to jointly optimize the model parameters.

Keywords: multimodal learning, anomaly detection, time series, graph attention networks, Transformer

目 录

插 图 目 录

表 格 目 录

第一章　绪论

1.1　研究背景与意义

异常检测是数据挖掘领域的一个重要研究课题，它从给定的数据集中检测异常的数据。准确地检测异常很重要，因为它们表明重大但罕见的事件的出现，并能为技术专家排除问题提供关键信息。例如，自动驾驶汽车系统中的异常数据可能表明汽车出现故障或者当前驾驶环境存在危险；交通监控中的异常可能意味着行人出现了违反交通规则的行为，而医疗影像中的异常可能表明检查者可能存在健康问题。异常检测已被广泛应用于许多应用领域，如医疗和公共卫生、欺诈检测、入侵检测、图像处理、网络流量检测和机器人行为监控等[1-3]。

随着物联网和大数据技术的快速发展，各种设备上的传感器随着时间的推移产生了大量的时间序列数据，挖掘这些数据中有用的信息已成为研究人员和从业者的一项重要工作，其中包括检测可能代表设备故障或受到攻击的异常值或异常事件。高效且稳健的时间序列异常检测有助于监控系统行为，从而避免潜在风险和经济损失。然而，从时间序列数据中检测异常值具有挑战性。现实世界时间序列的一个基本特征是异常行为（即概念）的定义通常随时间而变化。这种从旧概念转变为新概念的现象称为概念漂移[4]。概念漂移给离线异常检测方法带来了挑战。它们无法适应数据分布和异常定义的变化。最近的研究表明，使用在线学习的模型可以用来适应时间序列异常检测中的概念漂移[5]。例如，Online RNN-AD[6] 采用自回归方法与 GRU[7] 进行时间序列预测，并通过预测误差计算异常分数以获得检测结果。众所周知，现实世界中的时间序列通常具有长期和短期的重复模式，但是使用 RNN 的方法（例如 LSTM[8] 和 GRU）仍然难以捕获时间序列中的长期依赖关系[7]。此外，使用基于预测误差的动态学习率的在线学习很容易使模型过拟合异常，从而将异常视为正常点。最近，Transformer[9] 被提出作为一种新的结构来替代 RNN 在许多任务中，如自然语言处理（NLP）[10] 和时间序列预测[11]。Transformer 利用自注意力机制并行处理数据[9]。与基于 RNN 的模型不同，Transformer 可以捕获时间序列中的长期依赖关系。并且 Transformer 与在线学习相结合，可以解决过度拟合异常问题。这些优点使 Transformer 成为

时间序列异常检测的良好候选者。然而标准 Transformer 无法解决概念漂移问题，并且它的的时间复杂度为 $O(L^2)$，不利于线上场景的部署。因为在实际部署中，要求模型具有较低的时间复杂度。在本文中，我们提出在线稀疏 Transformer，它将在线学习与 Transformer 相结合，用于时间序列异常检测。我们设计了概念漂移适应方法（CDAM），并用它来动态调整 Transformer 的学习率，使在线稀疏 Transformer 能够快速适应新的概念数据，避免过拟合异常。此外，我们设计了根平方稀疏 Transformer，其时间复杂度从 $O(L^2)$ 降低到 $O(L\sqrt{L})$，并且性能仍然保持不变与标准 Transformer 相比。实验结果显示，在线稀疏 Transformer 在大多数情况优于其他时间序列异常检测方法。

我们周围的世界涉及多种模态，包括视觉、声音、质地、气味和味道等。随着信息技术和物联网技术的发展，指数级增加的传感器产生出大量的多模态数据。多模态机器学习旨在让机器像人类一样从多个方面去感受和理解我们周围的世界，具体来说就是构建一个可以处理和融合多种模态数据的模型。多模态模型相较于单模态模型通常具有更好的性能，但它们能否检测时间序列中对抗攻击产生的异常仍然是一个有待研究的问题。如今已经有许多工作证明了单模态深度学习模型能够有效地检测对抗攻击产生的异常，但是多模态模型对于检测时间序列中对抗攻击异常的有效性还尚未进行具体深入的研究。深度神经网络促进了许多应用，例如自动语音识别[12-15] 和自然语言处理[16-18]。然而，之前的工作表明神经网络容易受到对抗攻击的影响[19-21]，导致研究人员对相关的安全问题非常关注。一个例子是针对语音时序信号的对抗攻击。攻击者向输入音频添加了一个非常小的优化扰动，这是人类无法检测到的，这可以欺骗受害者模型产生不正确输出。因此，检测对抗攻击异常是非常重要的，它可以及时发现对抗攻击的出现或防止受害者模型产生错误的输出。已经有不少研究者提出了针对对抗攻击异常的异常检测算法。与图像域相比，只有少量工作探索了时间序列上的对抗攻击异常检测。例如，Samizade 等人[22] 使用提出的卷积神经网络（CNN）来检测音频时间序列上的黑盒和白盒攻击异常。此外，与使用单模态数据的方法相比，研究人员已经证明了多模态方法的优越性。最近，研究已将多模态数据引入用于检测对抗攻击异常。例如，MATCH[23] 利用医学数据中文本模态和数值模态

之间的一致性来防御单一模态上的对抗攻击。MATCH 没有考虑多种模态之间信息的完全融合，它只考虑不同模态之间的一致性，而忽略了它们之间的互补性。受计算机视觉中的视觉 Transformer（ViT）[24] 模型的启发，本文提出了一种新的多模态时间序列异常检测模型，称为多模态深度融合 Transformer（MDFT），它有效地结合了从语音时间序列和对应的文本模态中的信息。在 MDFT 中，音频和文本输入通过设计的 Transformer 编码器单独编码成特征表示。然后，将从每个编码器获得的表示传播到融合模块，融合模块通过跨模态注意力机制融合不同模态的表示，得到融合后的多模态表示。最后，将多模态表示传播到分类器，得到最终结果。多个实验上的结果证明了 MDFT 模型适用于检测语音时序上的对抗攻击异常。

传统上，经验丰富的工程师为每个监控的时间序列手动创建静态阈值以执行异常检测。然而，随着数据量的飞速增长，这种手动设置阈值的方式将是劳动密集型的[25]。此外，确定每个传感器的最佳阈值具有挑战性，尤其是在实体中存在多模态传感时。已经提出了很多异常检测方法来缓解上述不足，这些方法通过整合一个实体中所有单变量时间序列的检测结果来检测异常[26-27]。然而，多模态时间序列的实体通常涉及大量互连的单变量时间序列，这些单变量时间序列不断生成多模态时间序列数据，而这些传感器数据通常以复杂的非线性方式相互关联。因此，单个单变量时间序列无法响应实体的整体状态，简单地结合多个单变量时间序列的检测结果的方法往往表现不佳。由于多模态时间序列的复杂空间依赖性（例如拓扑结构和模态相关性）和时间依赖性（例如周期和趋势），多模态时间序列异常检测一直具有挑战性。此外，多模态时间序列不仅包含来自同一模态的时间序列之间的相关性（称为模态内相关性），还包含来自不同模态的时间序列之间的相关性（称为模态间相关性）。以前的多模态时间序列异常检测方法考虑了时间依赖性，包括支持向量回归[28]、自回归综合移动平均（ARIMA）[29] 和基于循环神经网络（RNNs）的模型[30]。这些方法可以捕获时间维度的动态变化，但忽略了不同时间序列之间的空间依赖性。为了缓解上述问题，已经有工作通过使用卷积神经网络（CNNs）来更好地建模空间关系[26]。然而，CNN 通常应用于图像视频和语音数据等常规数据，由于多模态时间序列之间的复杂拓扑结构，这

将导致图形数据的性能较差。图神经网络（GNNs）[31] 是在图数据中构建复杂拓扑关系的更有效范式，它也被开发用于异常检测并取得了可喜的成果。具体来说，Hang 等人[32] 采用图神经网络和门控循环单元（GRU）[33] 来研究时间序列的时空关系。尽管以前的方法取得了丰硕的进展，但它们未能明确地捕获到多模态时间序列之间的多模态相关性。为此，本文提出了多模态时空图注意网络，称为 MST-GAT，它采用流行的图注意网络（GATs）[34] 来显式捕获多模态时间序列之间的模态依赖关系。更具体地说，我们设计了多模态图注意网络（M-GAT），它包括一个多头注意力模块和两个关系注意力模块，即模态内和模态间注意，以捕获多模态时间序列之间的空间依赖关系。明确构建多模态时序数据中的依赖关系，有利于获得更好的输入数据特征表示。然后，我们引入了一个时间卷积网络，通过时间片上的标准卷积操作来捕获每个时间序列中的时间依赖性。此外，我们联合优化了重建和预测模块以整合它们的优势。重建模块负责重建输入数据，而预测模块旨在预测下一个时间戳的特征。重构概率和预测误差进一步用于解释检测到的异常。

综上所述，本文首先针对时间序列上的概念漂移问题提出了基于概念漂移检测的在线 Transformer 模型。之前的工作已经证明了多模态学习有助于得到更好的表示。鉴于此，本文接着提出了一个用于语音和文本的多模态数据上的异常检测方法。此外，为了更好地建模多模态时间序列中的空间和时间依赖关系，本文最后设计了一个多模态的空间-时间图注意网络，它采用了一个多模态图注意网络和一个时间卷积网络来建模多模态时间序列中的空间和时间相关性。

1.2 本文主要贡献

本文的主要贡献如下：

第一，我们提出在线稀疏 Transformer，它将在线学习和 Transformer 结合起来用于检测时序数据上的异常。具体来说，我们提出了概念漂移适应方法（CDAM），并用它来动态调整 Transformer 的学习率，这使得在线稀疏 Transformer 能够快速适应新的数据分布，并且避免过度拟合异常值。此外，我们设计了根平方稀疏 Transformer，其时间复杂度从 $O(L^2)$ 降低到 $O(L\sqrt{L})$，与标准 Transformer 相比，

其仍然保持着具有竞争力的性能。实验结果表明，在大多数情况下，所提出的方法在两个基准数据集上优于其他对比模型。

第二，我们提出了一个新的时间序列异常检测模型，称为多模态深度融合 Transformer（MDFT），有效地结合了从语音和文本模态中获得的信息。受计算机视觉中的视觉 Transformer（ViT）模型的启发，MDFT 被设计为一个基于 ViT 架构的多模态异常检测模型。在 MDFT 中，音频和文本输入通过设计的 Transformer 编码器被单独编码成特征表示。然后，从每个编码器得到的表示被传播到融合模块，该模块通过跨模态注意机制融合不同模态的表示，从而得到融合了多种模态信息的特征表示。最后，多模态表示被传播给分类器，以获得检测结果。在人工生成的多模态数据集上，实验结果证明了 MDFT 模型的有效性。

第三，我们提出了 MST-GAT，一种基于图注意网络的新型多模态时间序列异常检测方法。MST-GAT 尝试在多模态时间序列数据中明确建立空间-时间依赖关系以进行异常检测。MST-GAT 通过联合优化基于变分自动编码器的重建模块和基于多层感知机（MLP）的预测模块，以整合它们的优势。MST-GAT 取得了最高的 F1 分数（0.60 以上），最好的 AUC（0.92 以上），在基准数据集上的表现超过了强大的基线。消融研究进一步证明了 MST-GAT 中不同模块的有效性。此外本文在重建和预测结果的基础上为 MST-GAT 设计了一种有效的异常解释方法，实验证明 MST-GAT 具有良好的解释能力，能够获得与人类直觉一致的结果。

1.3 本文的组织结构

本文共分为六章，详细安排如下：

第一章梳理了时间序列异常检测模型及多模态学习的研究现状，并说明了本文的研究内容、研究意义以及主要的贡献。

第二章介绍了文中涉及到的相关背景知识。这一章主要包括：时间序列异常检测、多模态机器学习和图神经网络。

第三章针对时间序列异常检测中常见的概念漂移的问题，提出基于 Transformer 的在线异常检测模型。针对带有概念漂移的时间序列数据，该模型结合概念漂移检测模块和在线学习来增强模型对概念漂移的适应能力。接着，本章介绍

了所用的数据集以及实验设置。通过在两个数据集上的对比实验来验证提出模型的效果。

第四章设计了一个基于 Transformer 的多模态时间序列异常检测模型 MDFT, 并详细介绍了构成 MDFT 的基本组件和它们各自发挥的作用，包括文本编码器、音频编码器、多模态融合模块和异常检测器。本章在两个人工生成的多模态数据集上测试了 MDFT 的性能。

第五章提出多模态的空间-时间图注意网络（MST-GAT），以更好地捕获多模态时间序列数据中的时空依赖性。首先描述了 MST-GAT 各个模块的细节和原理；其次分别给出预测模块和重构模块的损失函数并给出 MST-GAT 中异常分数的计算公式；最后对比实验和消融实验来进一步验证 MST-GAT 在多模态时间序列数据集上的效果。

第六章总结了本文提出的单模态和多模态时间序列异常检测算法，并展望了未来可以扩展的研究方向。

第二章　相关背景知识

上一章详细介绍了本文的研究背景及其意义，本章将介绍本文相关的背景知识，包括时间序列异常检测、多模态学习和图卷积网络。

2.1　时间序列异常检测

时间序列异常检测是数据挖掘领域的一个热门任务，用于检测时间序列中不符合当前数据分布的数据点或数据区间。该任务广泛存在于欺诈检测、安全关键系统故障检测、网络安全事件检测和网络入侵检测中[35-38]。

传统的异常检测方法可以分为聚类[39]、基于距离[40]、基于密度[41]和基于隔离[42]的方法。最近，由于深度神经网络具有良好的特征表示能力和强大的泛化性，深度学习方法成为了主流[36]。现有的深度学习方法可以分为两种范式，即基于重构和基于预测的方法。一般来说，基于重构的方法通过重构输入数据来学习整个时间序列的潜在分布[43]。例如，深度自编码高斯模型 (DAGMM)[44] 通过结合深度自编码器网络和高斯混合模型 (GMM) 获得低维特征和基于重构的异常分数。RAMED[45] 利用多分辨率网络来鼓励重建的输出与输入的全局时间形状相匹配。OmniAnomaly[46] 将 VAE 应用到端到端结构中来重构输入数据，并根据重构概率检测异常。MAD-GAN[47] 提出了一种基于生成对抗网络 (GAN) 的方法，该方法同时使用生成器和判别器来检测异常，该方法将重建损失和判别结果结合为异常度量。基于预测的方法通过预测值和真实值之间的预测误差来判断异常。Cheng 等人[48] 提出了一种用于检测异常互联网流量的多尺度 LSTM 模型。DeepAnT[49] 基于卷积神经网络（CNN），使用自回归方法检测时间序列异常。Hundman 等人[50] 证明了长短期记忆 (LSTM) 在检测航天器异常方面的可行性，并介绍了一种无需依赖注释即可动态设置阈值的方法。图偏差网络 (GDN)[51] 利用图注意力网络 (GAT) 在多元时间序列中执行结构学习，并通过注意力权重解释检测到的异常。

2.2 多模态深度学习

随着深度学习在许多领域取得的成功，深度学习领域经开始研究更复杂的多模态学习任务。多模态数据通常由用于监测不同模态的传感器所产生的时序数据组成，目标是以互补的方式使用这些数据来提高复杂任务上的性能。不同于使用手动设计或手工制作的特定于模态的特征，深度学习可以自动学习每个模态的特征表示，然后将这些特征输入机器学习模型。多模态学习旨在充分利用来自不同模态的信息[52-53]。具体来说，多模态学习有助于整合多模态数据中的互补信息，并有助于提取相似信息以提高模型的鲁棒性。与使用单模态数据的模型相比，利用多模态数据的模型在大多数情况下能够取得更好的表现。如何将多模态信息融合成统一的表示是一个主要挑战。

深度学习模型的体系结构的灵活性为早期、中期和晚期实现多模态融合提供了可能。多模态融合中的早期融合也被称为特征级融合，晚期融合被也称为决策级融合。早期融合涉及将多个数据源（如多个传感器数据）集成到一个单一的特征向量中，然后用作机器学习算法的输入。为了缓解原始数据的对齐问题，在特征级融合之前，往往会从每个模态中提取更高级别的表示。大多数早期的融合模型都简化了假设，即各种信息源的数据之间存在条件独立性，但这在实践中是不成立的。因为多种模态往往高度相关（例如，视频和音频）。比如，Sebe 等人[54] 认为，不同信息源包含的信息只与另一个信息源存在相关性。在 Owens 等人[55] 的假设允许以独立于其他模态的方式来处理每个模态的数据。早期融合方法中最简单的形式是将多模态特征进行拼接，这是由 Poria 等人[56] 提出并实现的。多模态数据的早期融合可能无法充分利用所涉及模态的互补性，并可能会产生一个高维的、包含大量冗余信息的特征向量，主成分分析（PCA）等降维技术通常被用来减少特征向量中的冗余信息。作为 PCA 的非线性推广，自动编码器被广泛用于深度学习，它从原始数据中提取特征表示[57]。近年来，自动编码器已经扩展到用于学习多模态数据的嵌入空间，它可以在公共特征空间中有效地表示多模态数据[58]。多模态数据早期融合需要明确不同模态数据之间的时间同步性。通常情况下，会以相同的采样率对这些信号进行重采样来解决这个问题。

晚期融合指的是来自多个分类器的决策的聚合，每个分类器在单独的模态下

进行训练[59]。这种融合体系结构通常比早期融合更受研究者们的青睐，因为来自多个分类器的错误往往是不相关的，并且此种方法与特征无关。晚期融合中可以使用不同的规则来确定如何组合来自不同分类器的决策。这些融合规则包括：最大融合、平均融合、基于贝叶斯规则的融合等。晚期融合在 21 世纪初到中期很流行，当时集成分类器在机器学习界受到广泛关注。毫无疑问，当输入模态显著不相关、维度和采样率差异很大时，对多模态学习问题实施后期融合方法要简单得多。另一种方法是中期融合，它在如何以及何时融合从多模态数据中学习到的表示方面具有很大的灵活性。

中期融合是指神经网络对不同模态的数据分别提取特征，从而得到每个模态数据的高维特征表示。在多模态上下文中，当所有模态都转换为特征表示时，就可以将不同模态的表示融合到单个隐藏层中，然后学习多模态数据的联合表示。基于深度学习的多模态融合方案大部分属于中期融合。与其他融合技术相比，中期融合更能发挥深度多模态融合模型强大的特征表示能力。Karpathy 等人[60] 证明了中期融合方式在大规模视频分类问题上始终产生更好的结果，相比与早期融合和后期融合模型。因此，本文提出的多模态学习方法使用了中期融合的多模态融合方法。

多模态学习可以通过多核学习模型[61]、概率图模型[62]、神经网络模型[63] 等来实现。其中，神经网络模型由于其出色的融合多模态数据的能力，在许多任务中实现了显着的性能提升。例如，Iwana 等人[64] 使用多模态 CNN 和基于局部距离的表示来执行时间序列分类。Yang 等人[65] 引入了一种自适应加权算法和一个多头共同注意网络来建模多模态机器翻译中文本和视觉表示之间的关联。最近，多模态机器学习被引入到异常检测中[66]。Park 等人[67] 使用来自非异常场景的多模态数据来优化隐马尔可夫模型（HMM），并对机器人操作进行了异常检测。Park 等人[68] 利用基于 LSTM 的自动编码器来检测辅助喂食机器人的异常情况。

2.3 图神经网络

图神经网络（GNNs）在社交网络[69]和医学[70]等图结构数据中取得了显着的成功。典型的 GNN 假设节点的表示受到图结构中其相邻节点的影响。图卷积网络（GCNs）包括谱方法和空间方法[71]。谱方法对于基 (base) 的选择比较敏感，而空间方法则受到平移不变性的限制[72-73]。

注意力机制已成为深度学习模型中被广泛使用的组件并被已经被应用到了许多领域[9]。最近，注意力机制被引入到图神经网络中。图注意网络（GAT）[34]利用注意机制将聚合权重分配给相邻节点。图注意力网络的相关变体在与时间序列建模相关的任务中取得了进展，例如交通流量预测[74]和时间序列预测[75]。图注意力网络能很好地提取空间特征，在有向图中表现出优于图卷积神经网络的性能[76]。GAT 使用注意力机制将输入特征向量 h 映射到聚合表示 h' 中。注意分数 a_{ij} 公式为：

$$a_{ij} = \frac{\exp\left(\hat{a}_{ij}\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\hat{a}_{ik}\right)}, \tag{2.1}$$

$$\hat{a}_{ij} = \text{LeakyReLU}(\pi\left(\mathbf{W}h_i, \mathbf{W}h_j\right)), \tag{2.2}$$

其中 $\mathcal{N}_i$ 是节点 i 的邻居集，$\hat{a}_{ij}$ 表示节点 i 和节点 j 在归一化之前的注意力分数，$\pi(\cdot)$ 表示节点间的相关函数，$\mathbf{W}$ 是权重矩阵，h_i 是节点 i 的特征表示，LeakyReLU 是激活函数。每个节点的输出特征可以计算为：

$$h'_i = \sigma\left(\sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{W}h_j\right), \tag{2.3}$$

其中 σ 表示 sigmoid 激活函数。

2.4 本章小结

本章主要介绍了时间序列异常检测、多模态机器学习和图神经网络的背景知识和一些重要的相关工作。在下一章中，我们将引入本文的第一个工作，针对单模态时间序列异常检测任务所提出的在线稀疏 Transformer 模型。

第三章　基于 Transformer 的单模态时间序列异常检测

正如 1.1节中所介绍的，Transformer 能够有效地处理大部分序列相关的任务。然而，由于 Transformer 无法处理概念漂移问题并且具有较高的计算复杂度，这些问题导致将 Transformer 直接应用于时间序列异常检测场景时，往往会得到不太理想的效果。本章提出在线稀疏 Transformer，它使用了概念漂移适应方法（CDAM）来动态调整学习率，这使得模型能够快速概念漂移，并且避免过度拟合异常点。此外，我们设计了方根（root square）稀疏自注意力，其时间复杂度从 $O(L^2)$ 降低到 $O(L\sqrt{L})$，与标准自注意力相比，其性能依然保持着竞争力。

3.1　研究动机

时间序列异常检测（TSAD）是现实场景中的一项重要任务，被广泛用于数据监测和网络安全检测等领域。异常检测的一个常见方法是使用序列模型。作为一个有效的序列模型，Transformer 可以捕获到时间序列的长期依赖性，有望更好地完成异常检测任务。然而，在使用 Transformer 进行异常检测时，仍有一些问题需要解决。(1) 未能适应概念漂移。标准 Transformer 假设训练和测试数据来自相同的分布。然而，由于时间序列数据的时变性，可能会导致概念漂移问题，实际情况往往会违反这一假设。(2) 计算复杂度高。推理阶段的标准 Transformer 的时间复杂度随着序列长度 L 的增加而呈二次方增长。为了解决第一个问题，我们提出了概念漂移适应方法（CDAM）来动态调整 Transformer 的学习率。CDAM 旨在通过在线学习策略，充分利用旧的概念数据来优化新的概念数据上的新模型。为了解决第二个问题，我们提出了方根稀疏 Transformer，它只需要 $O(L\sqrt{L})$ 的时间复杂性。在几个异常检测基准上的结果表明，提出的模型优于许多异常检测方法，特别是在带有概念漂移的时间序列中。

3.2 问题建模

3.2.1 单模态时间序列异常检测

对于一个时间序列 $\mathbf{x} = [x_1, x_2, \ldots, x_i, \ldots, x_t] \in \mathbb{R}^{1\times t}$。$\mathbf{x}$ 对应的标签定义为 $\mathbf{y} = [y_1, y_2, \ldots, y_i, \ldots, y_t]$，其中 $y_i \in \{0, 1\}$。y_i 表明 x_i 是否为异常点，即 0 为正常，1 为异常。异常点 x_i 指的是行为异常、与之前正常点明显不同的点。与可以访问时间序列中所有点的批量异常检测不同，在时间序列异常检测中，t 时刻之后的点是未见的。

3.2.2 漂移检测方法（DDM）

给定时间序列中 i 时刻的点的表示 $(\boldsymbol{u}_i, v_i)$，其中 $\boldsymbol{u}_i$ 是不同属性的向量，v_i 是类标签。漂移检测方法（DDM）[77] 被提出来用于时序上的监督学习任务。对于每个时刻，时间序列预测模块 $\mathcal{F}$ 根据 $\boldsymbol{u}_i$ 生成预测 $\hat{v}_i$，然后 DDM 将其与真实标签 v_i 进行比较来确定预测是正确（$\hat{v}_i = v_i$）或不正确（$\hat{v}_i \neq y_i$）。DDM 根据模块 $\mathcal{F}$ 得到的预测错误率做出决策，即模型在当前对做出错误预测的概率。具体来说，DDM 假设 u_i 的分布是稳定的，并且当时间 i 的值增加时，学习算法 p_i 的错误率将会降低。对于时间序列中的每个时间点，DDM 监控模型 $\mathcal{F}$ 的在线错误率 p_i。如果考虑当前时刻 DDM 的检测误差服从二项分布，即 $e \sim b(i, p_i)$，则错误率的标准差 $s_i = \sqrt{p_i(1-p_i)/i}$。DDM 在漂移检测期间维护两个变量：$p_{min}$ 和 s_{max}。对于每个时刻，i、p_{min} 和 s_{min} 都会在 $p_i + s_i < p_{min} + s_{min}$ 时更新它们的值。如果 $p_i + s_i \geq p_{min} + 2 \times s_{min}$ 成立，DDM 会将当前时刻视为漂移警告级别。如果 $p_i + s_i \geq p_{min} + 3 \times s_{min}$ 成立，DDM 会将当前时刻视为漂移级别。p_t、s_t、p_{min} 和 s_{min} 将在漂移级别之后触发新的警告级别时重置。

3.3 模型架构

我们提出的在线稀疏 Transformer 拥有基于预测模块的异常检测架构，同时使用 CDAM 模块来应对概念漂移问题。

图 3.1说明了我们方法的整体结构。所提出的在线稀疏 Transformer 由三个模块组成，即时间序列预测模块、CDAM 模块和异常检测模块。当发生概念漂移

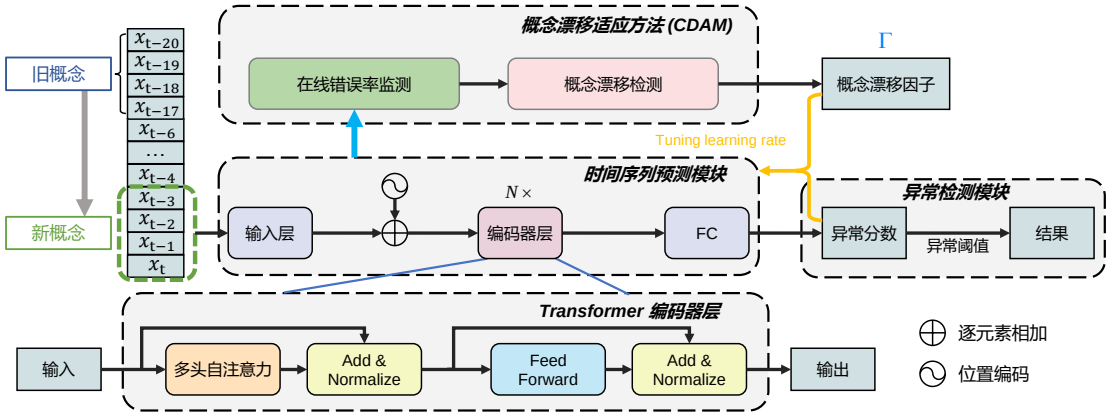


图 3.1 在线稀疏 Transformer 的整体结构图

时，在线稀疏 Transformer 根据 CDAM 的检测级别来动态地调整学习率，以尽快适应新的数据分布。具体来说，时间序列预测模块 $\mathcal{F}$ 由具有多头自注意力的 Transformer 编码器组成。我们利用这个模块来实现下一个时间戳的预测。CDAM 模块专为概念漂移检测而设计。它通过监测时间序列预测模块的预测错误率来检测概念漂移，然后通过检测结果动态调整 Transformer 的学习率。异常检测模块根据预测值计算异常分数，并使用预定义的阈值将数据点标记为正常或异常。在图 3.1中，我们还给出了 Transformer 编码器层的细节。每个 Transformer 编码器层由多头注意力、层归一化和前馈网络（FFN）组成。在以下小节中，Transformer 默认指的是 Transformer 编码器。

3.3.1 时间序列预测模块

我们采用自回归策略训练时间序列预测模块，将时间戳 t 处的当前元素 x_t 作为其预测目标。原始时间序列被转换成几个大小为 w 的重叠窗口序列。在每次 t，我们使用长度为 w 的历史窗口作为输入：

$$\mathbf{i}_t = [x_{t-w}, x_{t-w}, ..., x_{t-1}]. \tag{3.1}$$

Transformer 中的多头自注意力机制使其能够捕获时间序列中的长期和短期依赖关系，并且不同的注意力负责提取不同方面的特征。这些优势使 Transformer 成为顺序任务的良好候选者。在多头自注意力中，不同注意力头的最终表示是分别并行计算的。具体来说，令 $\mathbf{W}_h^Q \in \mathbb{R}^{w \times d_k}$, $\mathbf{W}_h^K \in \mathbb{R}^{w \times d_k}$ 和 $\mathbf{W}_h^V \in \mathbb{R}^{w \times d_v}$ 是计算

查询（**Q**）、键（**K**）和值（**V**）矩阵的权重，其中 w 表示滑动窗口大小，h 代表第 h 个注意力头，d_k 是 query 和 keys 的维度，d_v 是 value 的维度。多头注意力在 t 时刻处同时将输入 $\mathbf{i}_t$ 转换为 H 组 **Q**、**K**、**V** 矩阵。对于 head h ($h = 1, \ldots, H$)，查询、键和值矩阵计算为 $\mathbf{Q}_h = \mathbf{i}_t \mathbf{W}_h^Q$, $\mathbf{K}_h = \mathbf{i}_t \mathbf{W}_h^K$ 和 $\mathbf{V}_h = \mathbf{i}_t \mathbf{W}_h^V$。在前面的转换之后，使用 $\mathbf{Q}_h$、$\mathbf{K}_h$ 和 $\mathbf{V}_h$ 之间的缩放点积注意力来计算 head h 对应的输出矩阵：

$$\begin{aligned} \mathbf{O}_h &= \text{Attention}\left(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h\right) \\ &= \text{Softmax}\left(\frac{\mathbf{Q}_h \mathbf{K}_h^T}{\sqrt{d_k}}\right) \mathbf{V}_h. \end{aligned} \tag{3.2}$$

多头 attention 拼接各个 attention 的结果，表示如下：

$$\mathbf{Z} = \text{Concat}\left(\mathbf{O}_1, \ldots, \mathbf{O}_H\right) \mathbf{W}_{out}, \tag{3.3}$$

其中 **Z** 表示多头注意力的输出表示，concat$(\cdot)$ 是拼接操作，H 是 head 的个数，$\mathbf{W}_{out}$ 表示权重矩阵。

之后，前馈网络采用多头注意力的输出作为输入。前馈网络（FFN）由全连接网络组成，FFN 的输出表示为：

$$\mathbf{R} = \text{Relu}\left(\mathbf{Z}\mathbf{W}_1 + b_1\right) \mathbf{W}_2 + b_2, \tag{3.4}$$

其中 Relu 是非线性激活函数，$\mathbf{W}_1$ 和 $\mathbf{W}_2$ 是权重矩阵，b_1 和 b_2 是偏置向量。最后，FFN 的输出通过全连接（FC）层进行变换，以得到预测结果 x_t'。

总体而言，原始输入序列 i_t 在 t 时刻被输入到模型参数集为 $\mathbf{W}_{t-1}$ 的 Transformer 中，得到 t 时刻的预测值 x_t'：

$$x_t' = \mathcal{F}(i_t, \mathbf{W}_{t-1}) = \mathcal{F}(x_{t-w}, x_{t-w}, ..., x_{t-1}, \mathbf{W}_{t-1}), \tag{3.5}$$

其中 $\mathcal{F}$ 表示时间序列预测模块。

3.3.2 概念漂移适应方法（CDAM）

DDM 是一种监督学习算法，需要时间戳级别的异常标签来监控模型的在线错误率。然而，时序异常检测往往是无监督的。在 i 时刻，我们无法获得当前数据的异常标签。因此，我们提出了概念漂移适应方法（CDAM）来解决上述问题，该方法可以应用于没有注释数据的无监督异常检测任务。

假设我们有时间序列 $\mathbf{x} = [x_1, x_2, \ldots, x_t]$。对于 t 时刻的数据 x_t，时间序列预测模块 $\mathcal{F}$ 得到预测值 $\hat{x}_t$。预测结果可以为真 $(|\hat{x}_t - x_t| \leq \gamma_1)$ 或假 $(|\hat{x}_t - x_t| > \gamma_1)$，其中 γ_1 是从验证集中选择的概念漂移阈值。CDAM 通过在漂移检测窗口监测预测模型 $\mathcal{F}$ 的在线错误率来确定警告等级和漂移等级。设 p'_t 和 s'_t 为 t 时刻在线错误率的概率和标准差，p'_{min} 和 s'_{min} 是 CDAM 维护的两个变量。p'_{min} 和 s'_{min} 在 $p'_t + s'_t < p'_{min} + s'_{min}$ 时更新它们的值。如果 $p'_t + s'_t \geq p'_{min} + \frac{2}{3}s'_{min}$ 成立，CDAM 将判断为漂移警告。如果 $p'_t + s'_t \geq p'_{min} + 2 \times s'_{min}$ 成立，CDAM 将其标记为漂移。p'_t, s'_t, p'_{min} 和 s'_{min} 将在漂移级别之后出现的新的警告级别时重置。CDAM 首先生成一个概念漂移因子 Γ 来代表不同的级别，即 $\Gamma = 0$ 代表正常级别，$\Gamma = 1$ 是警告级别，$\Gamma = 2$ 表示漂移水平。然后 CDAM 根据不同级别动态调整时间序列预测模块 $\mathcal{F}$ 的学习率。

假设所提出模型的在线错误率在从 t_1 到 t_2 的时间内呈先增加后减小的趋势。在线错误率在时间 w_1 处上升到警告水平，在时间 d_1 处上升到漂移水平，在时间 d_2 处下降到漂移水平，在时间 w_2 处下降到警告水平 $(t_1 < w_1 < d_1 < d_2 < w_2 < t_2)$。其中，$w_1$ 到 d_1 和 d_2 到 w_2 的时间属于警告级别。d_1 到 d_2 的时间属于漂移级别，其他时间属于正常级别。在线错误率低于正常级别时，说明在线稀疏 Transformer 的性能比较稳定。所以我们保持学习率为 η_0 不变。动态学习率被应用于每个时间戳，当发生概念漂移时，CDAM 将使在线稀疏 Transformer 能够尽快适应新的数据分布。

动态学习率可以表示为：

$$\eta_t = \begin{cases} \eta_0, & \Gamma = 0 \\ \alpha_t \cdot \eta_0, & \Gamma = 1 \\ \tau \cdot \alpha_t \cdot \eta_0, & \Gamma = 2 \end{cases}, \tag{3.6}$$

其中 α_t 是 t 时刻的异常分数，将在下一小节中定义，η_0 是在线学习的初始学习率，τ 表示平衡超参数来调整学习率。总体而言，当 CADM 处于正常水平时，学习率设置为 η_0。当 CADM 达到警告级别时，模型参数会以 $\alpha_t \cdot \eta_0$ 的学习率进行更新。当 CADM 达到漂移水平时，我们利用 $\tau \cdot \alpha_t \cdot \eta_0$ 的学习率来更新模型参数。

3.3.3 在线优化和异常检测

所提出的模型通过在线学习在每个时间戳进行优化。在线学习有助于模型及时地更新模型参数以适应新的数据分布。模型参数集的更新过程可以表示为：

$$\mathbf{W}_t = \mathbf{W}_{t-1} - \eta_t \cdot \nabla G_t(\mathbf{W}_{t-1}), \tag{3.7}$$

其中 $\mathbf{W}_t$ 表示在 t 时刻更新的模型参数集，$\nabla G_t(\mathbf{W}_{t-1})$ 表示模型在 t 时刻的梯度，$G_t(\cdot)$ 是平方误差损失函数。给定 x_t 的预测值 x'_t，t 时刻的异常分数计算如下：

$$a_t = e_t^2 = (x_t - x'_t)^2, \tag{3.8}$$

其中 e_t 表示 x_t 和 x'_t 之间在 t 时刻的预测误差。假设 γ_2 是异常检测阈值，当 $\alpha_t > \gamma_2$ 时，检测模块会将当前数据标记为异常。

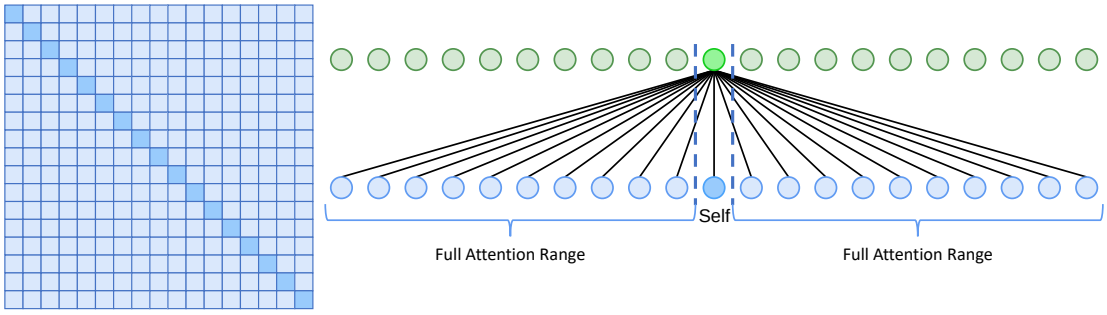
3.4 方根稀疏自注意力

3.4.1 完全自注意力

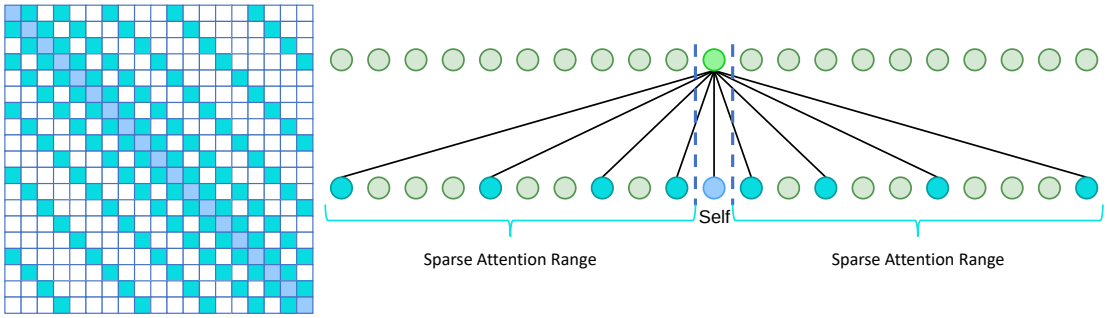
标准 Transformer 中的自注意力机制捕获输入序列中任意两个向量的相关性，并且易于并行。如图 3.2(a)，左边是标准的自注意力矩阵，右边是两个矩阵的关联图。对于长度为 L 的序列，自注意力计算序列中两个向量之间的相关性，从而产生一个 L^2 大小的相关矩阵。因此，标准自注意力的复杂度是 $O(N^2)$。这会导致计算速度变慢，并且难以有效地对具有长期依赖性的长期时间序列进行建模。时间序列异常检测任务对模型的实时性要求很高，计算速度慢、内存占用大，会使得模型在实际应用中难以部署。因此，我们提出一种用于异常检测的新型稀疏自注意力架构来解决这个问题。

3.4.2 方根稀疏自注意力

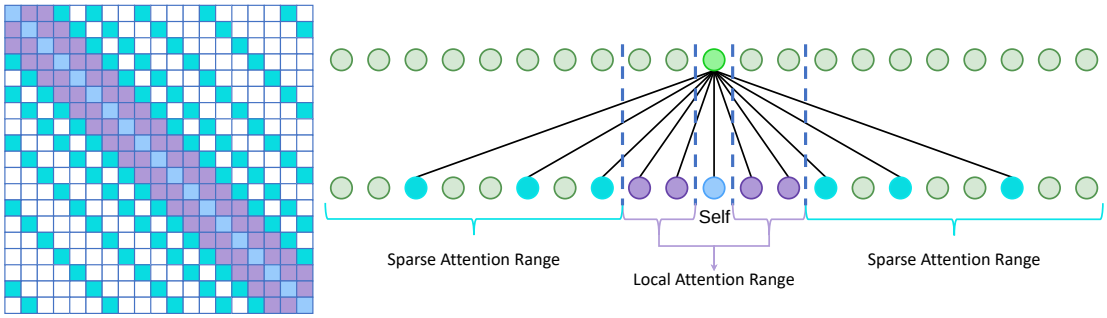
受 log-sparse 自注意力[78] 的启发，我们提出了方根（root square）稀疏自注意来解决上述问题。虽然现有的 log-sparse 自注意力只需要 $O(L\log L)$ 的时间和空间复杂度，但它过于稀疏，无法有效地捕获时间序列中的长期依赖关系。与 log-sparse 自注意力相比，我们的方根稀疏自注意力缓解了上述问题。它在算法



（a）完全自注意力



（b）方根稀疏自注意力



（c）局部 + 方根稀疏自注意力

图 3.2 Transformer 中不同的自注意力机制示意图

复杂性和捕获长期依赖关系的能力之间取得了平衡。它允许每个单元只关注与自身相距特定增量步长的单元。具体来说，对于感受野为 L 的方根稀疏自注意力，它关注具有以下索引的单元格：

$$I = \{L - \frac{1(1-1)}{2} + 1, L - \frac{2(2-1)}{2} + 1, \ldots, L - \frac{n(n-1)}{2} + 1\}. \tag{3.9}$$

这样，根据在线稀疏 Transformer 中的方根稀疏自注意力机制，时间复杂度从 $O(L^2)$ 减少到 $O(L\sqrt{L})$。如图 3.2(b)，左边是注意力矩阵，右边是关联图。在后续的实验中，我们会给出完全注意力（即标准注意力）和方根稀疏注意力之间的详细性能比较。

表 3.1 Yahoo 数据集的详细信息，它包含四个子集

子集	时间序列条数		值的范围	异常点个数
A1	67	min	0	0
		max	7,845,760	12
A2	100	min	-2,204	1
		max	128,420	3
A3	100	min	-7,988	1
		max	7,006	16
A4	100	min	-6,171	1
		max	6,324	16

3.4.3 结合局部自注意力

稀疏自注意力通过引入局部的概念来构建更精确的依赖关系。局部自注意力选择丢弃全局关联，重新引入局部关联。它允许每个单元格与其自身以及大小为 M 的左右窗口的单元格相关联，以便可以使用更多的局部信息进行预测。然而，局部注意力直接牺牲了长距离相关性，因此在 M 之外恢复方根稀疏注意力来解决这个问题。异常检测任务要求模型能够捕获时间序列的局部和长期依赖关系。稀疏自注意力和局部自注意力的结合可以更好地地满足这个要求。如图 3.2(c) 所示，注意矩阵在左边，关联图在右边。从注意力矩阵的角度来看，与当前单元格相对距离小于 M 的单元格属于局部注意力范围，而相对距离大于 M 的单元格属于稀疏注意力范围。

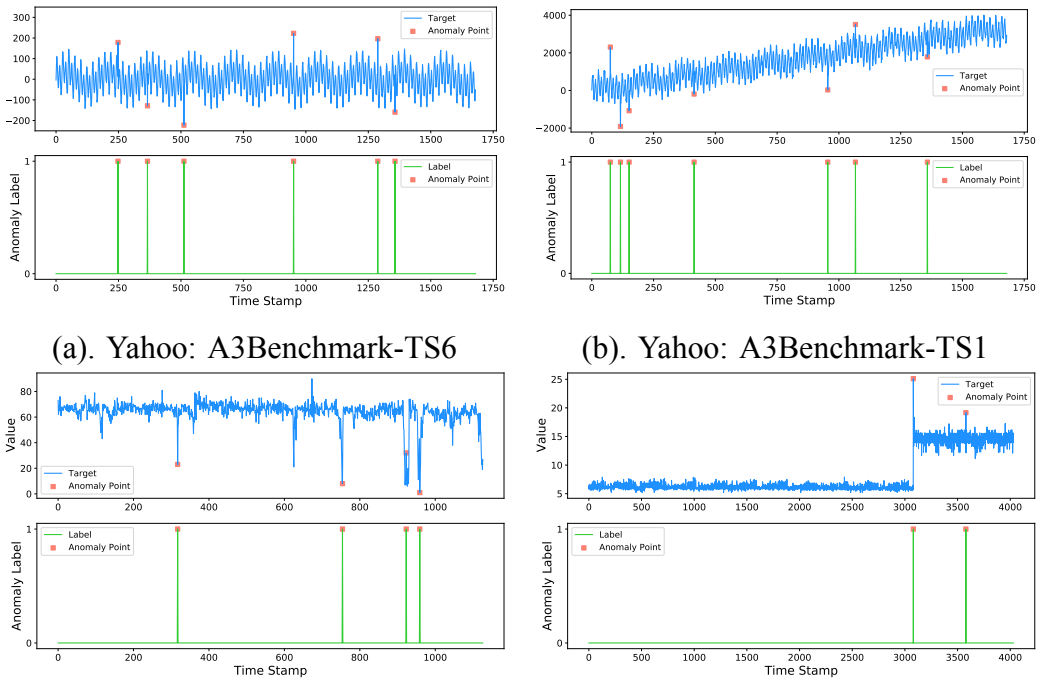
3.5 实验

3.5.1 数据集

我们在 Yahoo 数据集[79] 和 Numenta Anomaly Benchmark（NAB）数据集[80] 上评估提出的模型。这两个数据集的细节信息在表 3.1和表 3.2 中给出。图 3.3展示了这两个数据集的示例，其中（a）和（c）是没有概念漂移的时间序列，而（b）和（d）是带有概念漂移的时间序列。每个子图的上半部分是原始时间序列，下

表 3.2 NAB 数据集的详细信息，它包含六个子集

子集	时间序列条数		值的范围	异常点个数
ArtificialWithAnomaly	6	min	-22	1
		max	165	1
RealAdExchange	6	min	0	1
		max	16	4
RealAWSCloudWatch	16	min	0	0
		max	863,964,000	3
RealKnownCause	7	min	0	2
		max	39,197	5
RealTraffic	5	min	0	1
		max	5,578	4
RealTweets	10	min	0	2
		max	13,479	5



(a). Yahoo: A3Benchmark-TS6　　(b). Yahoo: A3Benchmark-TS1

(c). NAB: speed_7578　　(d). NAB: rds_cpu_utilization_cc0c53

图 3.3 来自雅虎和 NAB 数据集的时间序列示例

半部分是数据标签，其中标签“1”和“0”分别表示异常点和正常点。

雅虎数据集由 367 条真实和合成时序数据构成，每个时序数据有 1,420-1,680 个数据点。该数据集包含四个子集，即 A1、A2、A3 和 A4。请注意，A1 包含真

实世界的时间序列，而 A2、A3 和 A4 由合成时间序列组成。NAB 数据集由来自不同来源的传感器生成的 58 个时间序列组成，例如网络利用率、工业机器、社交媒体等。每个时间序列有 1,000-22,000 个数据点。NAB 数据集包含三种类型的时间序列，即人工生成的时间序列、没有任何异常的时间序列和真实世界生成的时间序列。

3.5.2 对比方法

我们对七种基线方法进行了详细分析，并将它们与我们在 Yahoo 和 NAB 数据集上的方法进行了比较。基线方法可以被分为两种类型，非深度学习方法（HTM[80]、Skyline[81]、Gaussian Process[5]、Context OSE[82]）和深度学习方法（MAD-GAN[47]、DeepAnT[49]、Online RNN-AD[6]）。

- **HTM：** Hierarchical Temporal Memory（HTM）是一种源自神经科学的时间序列异常检测算法，它对数据流中的空间和时间相关性进行建模。
- **Skyline：** 它是一种流行的开源异常检测算法。该方法使用投票机制来综合考虑来自不同简单检测器的检测结果，并获得最终的异常分数。
- **高斯过程：** 它采用稀疏高斯过程以自回归的方式检测时间序列异常。
- **Context OSE.** 它是一种基于无监督学习的时间序列异常检测方法，用于检测具有相似属性的一组时间序列并且它需要一个窗口大小参数。
- **MAD-GAN：** 它利用生成对抗网络来检测异常，并使用新颖的 DR 分数作为异常度量。MAD-GAN 的生成器和判别器均采用了 LSTM。
- **DeepAnT：** 它是一种基于深度卷积神经网络的方法，用于检测时间序列数据中的异常。DeepAnT 由两个模块组成，称为时间序列预测器和异常检测器。
- **Online RNN-AD：** 它是一个基于 GRU 的在线异常检测模型，具有动态学习率。它采用逐点预测误差进行异常检测。

3.5.3 实验设置

我们使用 Adam 优化器[83] 在 NVIDIA RTX 2080Ti GPU 上优化我们提出的方法。学习率从 0.001 开始，epoch 为 50。最优概念漂移阈值和异常检测阈值对我们

的方法很重要，很难找到适合所有时间序列的这两个阈值。一般来说，Yahoo 和 NAB 中的每个子集都具有相似的属性，因此我们采用 peaks-over-threshold（POT）算法[84] 来搜索每个子集的最佳阈值。我们在我们的方法中使用网格搜索来选择超参数。我们根据经验将 η_0、τ 和 M 分别设置为 0.001、1.5 和 20。时间序列预测模块包含一个三层的 Transformer 编码器，每层有 8 的注意力头和 512 的隐藏维度。为了进行公平比较，我们对一些超参数执行网格搜索。对于 MAD-GAN 和 Online RNN-AD，隐藏状态大小从 $\{32, 64, 128\}$ 中选择，层数从 $\{1, 2, 3\}$ 中选择。对于 DeepAnT，卷积层的隐藏维度选自 $\{32, 64, 128\}$。对于在线稀疏 Transformer，Transformer 编码器的层数从 $\{5, 3, 2\}$ 中选择。我们在最佳设置下评估基线方法。

值得注意的是，滑动窗口 w 是另一个参数。适当的历史窗口有助于 Transformer 在捕获时间序列中的局部依赖关系和长期依赖关系之间取得平衡。为了选择合适的 w，我们对 NAB 的两个子集进行了实验。经过初步实验，我们根据经验选择了 60、80、100 和 120 的窗口大小。表 3.3展示了我们模型在不同窗口大小下的 F1 分数，其中窗口大小定义为 w。通常，最好的结果出现在 $w = 100$ 时。

3.5.4 评价指标

我们采用精度、召回率和 F1 分数作为评估指标。精度是正确检测到的异常与检测到的异常数量的比率。召回率是正确检测到的异常与所有实际异常的数量之比。F1 分数联合考虑精度和召回率。异常点和正常点的数量通常是不平衡的，这表明 F1 分数将是一个很好的评估指标。它们的公式如下：

$$\text{precision} = \frac{TP}{TP + FP}, \tag{3.10}$$

$$\text{recall} = \frac{TP}{TP + FN}, \tag{3.11}$$

$$F1\text{分数} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{3.12}$$

其中 TP、FP、FN 分别表示真阳性、假阳性和假阴性。

3.5.5 实验结果

在本小节中，我们首先将我们的在线稀疏 Transformer 与其他方法进行比较，然后研究了概念漂移对不同方法性能的影响。表 3.4 和表 3.5 报告了不同方法对

表 3.3 不同历史窗口大小对模型性能的影响

子集	w=60	w=80	w=100	w=120
RealAWSCloudWatch	0.194	0.198	**0.203**	0.200
RealKnownCause	0.205	0.209	**0.212**	**0.212**

表 3.4 Yahoo 数据集上的 F1 分数比较

子集	HTM	Skyline	GP	C.OSE	MAD-GAN	DeepAnT	RNN-AD	Ours
A1	0.40	0.41	0.51	0.35	0.55	0.51	0.58	**0.62**
A2	0.83	0.71	0.74	0.77	0.82	0.79	0.85	**0.91**
A3	0.77	0.74	0.70	0.52	0.72	0.76	0.77	**0.79**
A4	0.58	0.51	0.59	0.50	0.61	0.58	0.63	**0.68**
Average	0.65	0.59	0.64	0.54	0.68	0.66	0.71	**0.75**

表 3.5 NAB 数据集上的 F1 分数比较

子集	HTM	Skyline	GP	C.OSE	MAD-GAN	DeepAnT	RNN-AD	Ours
ArtiWAnomaly	0.13	0.09	0.09	0.11	0.15	0.13	0.19	**0.22**
AdExchange	**0.54**	0.27	0.28	0.33	0.39	0.35	0.41	0.52
AWSCloudWatch	0.15	0.12	0.11	0.14	0.19	0.16	0.20	**0.25**
KnownCause	0.18	0.13	0.15	0.16	0.18	0.15	0.23	**0.26**
Traffic	0.41	0.25	0.36	0.26	0.42	0.38	0.45	**0.52**
Tweets	0.10	0.07	0.07	0.11	0.12	0.10	**0.16**	0.14
Average	0.25	0.16	0.18	0.19	0.24	0.21	0.27	**0.32**

Yahoo 和 NAB 数据集进行异常检测的结果，其中最好的分数以粗体显示。最后一行报告每种方法的平均 F1 分数。在表 3.4 中，我们观察到（1）大多数子集中，深度学习模型，即 MAD-GAN、DeepAnT、Online RNN-AD 和 online sparse Transformer，比非深度学习模型，即 HTM、Skyline、高斯过程和 Context OSE 表现更好；（2）在流式异常检测方法中，Online RNN-AD 比 HTM 有更好的性能，表明深度学习方法可以比传统模型捕获更复杂的时间序列依赖关系；（3）Online sparse Transformer 在所有子集上表现最好。与最佳基线相比，平均 F1 分数从 0.71 提高到 0.75。换句话说，在线稀疏 Transformer 比基线方法取得了更好的结果，因为它可以对时间序列之间的复杂关系进行建模，并有效地减少概念漂移对模型的影响。

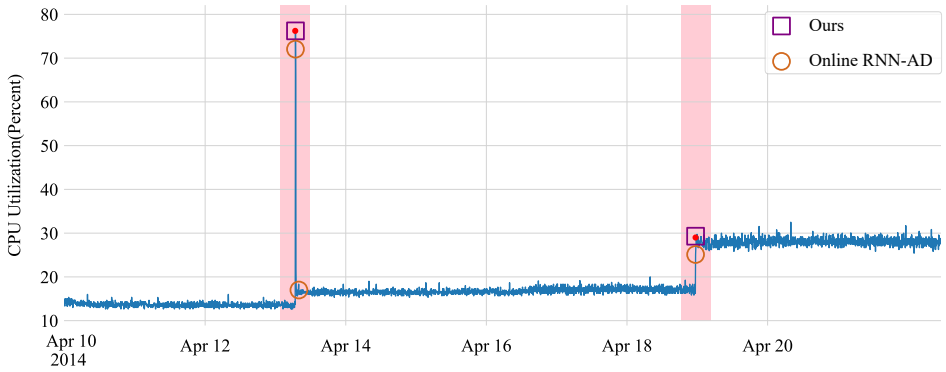


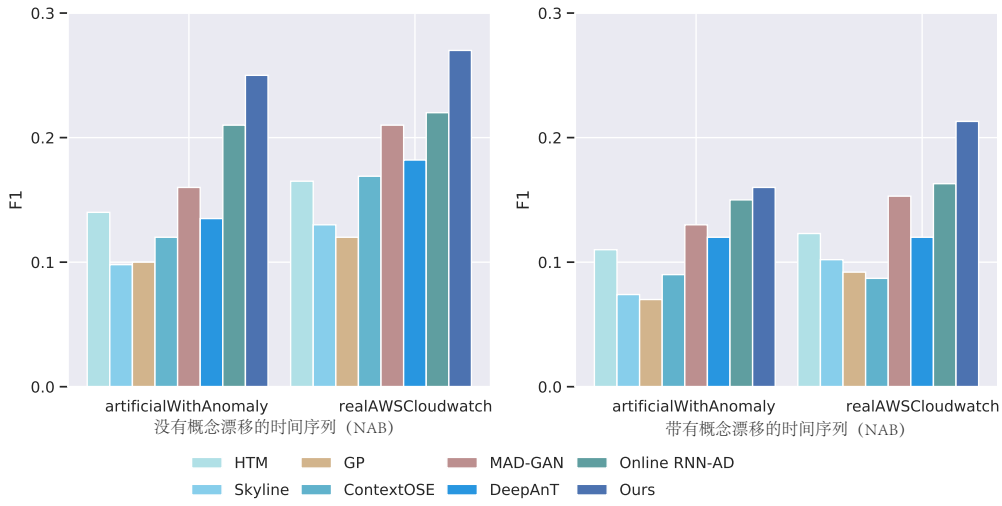
图 3.4 异常检测结果示例，红点表示异常

在表 3.5中，我们可以在表 3.4 中得到类似的观察结果。此外，我们还观察到（1）与其他深度学习基线方法（即 MAD-GAN 和 Online RNN-AD）相比，DeepAnT 的平均 F1 分数最差。我们推测基于 CNN 的 DeepAnt 比基于 RNN 的模型更难捕获数据中的时间依赖性；（2）与 Yahoo 数据集的表现相比，HTM 排名从第五到第三。因为 HTM 是专门为 NAB 数据集开发的流式异常检测算法；（3）总体而言，在线稀疏 Transformer 在四个子集上优于所有其他方法，在两个子集上排名第二。它表明在线稀疏 Transformer 是 TSAD 任务的良好候选者。

为了更详细地比较，图 3.4提供了在线稀疏 Transformer 和最佳基线方法，即在线 RNN-AD，在具有两个异常的真实时间序列中的案例研究. 我们可以观察到 Online RNN-AD 未能检测到所有异常，因为它在第一个异常附近产生了误报。同时，在线稀疏 Transformer 得到准确的结果，没有任何假阳性和假阴性。该案例证明我们的模型避免了对异常点的过度拟合，并且可以适应概念漂移，从而保证了准确的异常检测结果。

我们在 Yahoo 的子集 A1、A2 和 NAB 的子集 ArtificialWithAnomaly、RealAWSCloudWatch 上进行了另一组实验。对于每个子集，我们根据时间序列中是否存在概念漂移，将其分为两部分。如图 3.5所示，我们将提出的方法与其他基线方法进行了比较。可以观察到，与在有概念漂移的子集上的结果相比，大多数方法在没有概念漂移的相应子集上取得了更好的性能，这表明概念漂移增加了异常检测的难度。在 realAWSCloudwatch 子集上，我们的模型在具有概念漂移的数据的 F1 分数和没有概念漂移的数据上实现了 0.04 的相对改进。我们的方法在所有子集上的性能明显优于其他列出的方法。换句话说，我们的方法对于有概念

（a）Yahoo 的 A1 和 A2 子集



（b）NAB 的 artificialWithAnomaly 和 realAWSCloudwatch 子集

图 3.5 在子集中有概念漂移/没有概念漂移情况下模型性能

漂移和没有概念漂移的数据都是有效的。

我们进行消融研究以验证每个模块的作用。这是了解我们的模型如何工作的关键步骤。在第一个实验中，我们旨在验证 CDAM 模块的有效性。表 3.6 报告了我们模型在 Yahoo 的 10 个时间序列中的评估结果，其中“ours w/CDAM”和“ours w/o CDAM”分别表示在线稀疏 Transformer 是否使用了 CDAM 模块。数字 1-5 是带有概念漂移的时间序列，而 6-10 是没有概念漂移的时间序列。CDAM 模型在大多数情况下都能取得更好的结果。从概念漂移适应的角度来看，CDAM 可以检测概念漂移的出现并动态调整学习率。它比没有 CDAM 的在线 Transformer 模型更好更快地适应概念漂移。我们推测 CDAM 有助于我们提出的模型在自回归预测过程中避免过度拟合异常，从而保持预测模块的稳定性。综上所述，我们

表 3.6 在 10 个不同时间序列上的 F1 分数比较

时间序列	Ours w/ CDAM			Ours w/o CDAM		
	P	**R**	**F1**	**P**	**R**	**F1**
1	0.83	0.98	0.90	0.89↑	1.00↑	0.94↑
2	0.38	0.73	0.50	0.43↑	0.76↑	0.55↑
3	0.83	0.67	0.74	0.90↑	0.67	0.77↑
4	0.48	0.80	0.60	0.53↑	0.84↑	0.65↑
5	0.16	0.36	0.22	0.23↑	0.30	0.26↑
6	0.05	0.33	0.09	0.11↑	0.12	0.12↑
7	0.28	0.70	0.40	0.35↑	0.56	0.43↑
8	0.79	0.20	0.32	0.85↑	0.23↑	0.37↑
9	0.81	0.20	0.32	0.88↑	0.23↑	0.36↑
10	0.57	0.50	0.53	0.63↑	0.51↑	0.56↑

表 3.7 Yahoo 数据集上 “Sparse + CDAM” 及其变体之间 F1 分数的比较

数据集	Full	Sparse	Full + CDAM	Sparse + CDAM	LSTM + CDAM
A1	0.60	0.58	**0.64**	0.62	0.60
A2	0.90	0.88	**0.92**	0.91	0.87
A3	0.76	0.75	**0.79**	**0.79**	0.76
A4	0.62	0.64	0.67	**0.68**	0.65

可以看到 CDAM 模块在提出的在线稀疏 Transformer 中发挥了重要作用。

在第二个消融研究中，我们在 Yahoo 数据集上对比了使用稀疏自注意力的模型与使用全自注意力的模型表现，结果如表 3.7 所示。“Full” 表示 Transformer 使用了标准自注意力方法，而 “Sparse” 表示使用了提出的 sparse 自注意力。“+ CDAM” 表示模型使用 CDAM 模块。请注意，没有 CDAM 的模型的学习率设置为 $\eta_t = a_t \cdot \eta_0$。

正如我们预期的那样，在大多数情况下，完全注意力 Transformer 的性能略好于稀疏注意力 Transformer，无论它们是否配备了 CDAM 模块。然而，在 A3 子集上，Full+CDAM 和 Sparse+CDAM 模型实现了相同的结果。更有趣的是，它甚至比 A4 子集上的完全注意力模型略好，这意味着我们使用稀疏注意力的 Transformer 模型可以学习数据中的短期和长期依赖关系。稀疏自注意力不仅取得了竞争性的性能，而且降低了算法的复杂性。此外，与 Sparse 模型相比，Sparse+CDAM

表 3.8 不同注意力机制的比较

	复杂度	time/epoch (s)	F1 分数
完全自注意力	$O(L^2)$	3381	0.325
方根稀疏自注意力	$O(L\sqrt{L})$	2265	0.321

模型在四个子集上始终获得了 F1 分数增益，显示了 CDAM 模块的有效性。

此外，我们在 Yahoo 数据集上将在线稀疏 Transformer 与基于 LSTM 的变体模型（即 LSTM+CDAM）进行了比较。LSTM+CDAM 在我们的模型中用 LSTM 替换 Transformer 编码器来执行时间序列预测。为了公平比较，我们使用网格搜索来确定 LSTM 的超参数。在这个实验中，LSTM 包含三个隐藏层，每个隐藏层有 64 个单元。表 3.7 显示了实验结果。Ours+CDAM 在所有子集上都取得了比 LSTM+CDAM 更好的性能，这表明 Transformer 比 LSTM 更适合 TSAD 任务。

最后，我们在 NAB 数据集上进行了实验来比较方根稀疏自注意力和完全自注意力之间的训练时间。实际时间开销和理论时间复杂度的比较见表 3.8。在训练阶段，sparse+CDAM 可以显着减少 epoch 的平均时间，同时保证了具有竞争性的性能。这些验证了方根稀疏注意力的时间复杂度低于完全注意力。更小的时间复杂度保证了我们模型的实时性能，更有利于在现实场景中的部署。这些结果再次证明了方根稀疏自注意力的有效性。

3.6 本章小节

本章提出了一种基于概念漂移检测的时间序列异常检测框架，称为在线稀疏 Transformer。为了解决时间序列中的概念漂移问题，我们设计了 CDAM 模块来动态调整模型的学习率。CDAM 和在线学习共同促进在线稀疏 Transformer 及时地更新模型参数以适应新的数据分布。此外，由于 Transformer 中自注意力的时间复杂度很高，我们设计了方根稀疏自注意力来替换原来的自注意力，其复杂度仅为 $O(L\sqrt{L})$，这减少了模型的时间开销，更有利于模型在真实场景中的部署。我们在 Yahoo 和 NAB 数据集上比较了在线稀疏 Transformer 与基线方法的性能。实验结果表明，无论有没有概念漂移，所提出的模型在性能方面都优于基线。

第四章　多模态时序对抗攻击异常检测

在上一章中提出了在线稀疏 Transformer 用于解决单模态时间序列中的概念漂移问题。本章针对多模态时间序列数据提出了多模态深度融合 Transformer（MDFT）。不同于现有的时序异常检测模型，MDFT 不仅充分捕获了音频时间序列中的时间依赖关系，而且还利用了多模态数据具有一致性及互补性的特点，通过结合多模态学习以得到更优的特征表示。

4.1　研究动机

尽管深度神经网络在许多任务中显示出巨大的潜力，但它们很容易受到对抗攻击的影响，这些由对抗攻击产生的样本是通过向自然样本添加小扰动而生成的，我们将这些样本称之为对抗攻击异常。本章的研究对象是语音时间序列，结合了包括语音数据和对应的文本数据来进行多模态异常检测。最近的许多研究证明，充分利用不同的模态可以有效地增强深度神经网络的表示能力。在本章中，我们设计了多模态深度融合 Transformer（MDFT）。首先，音频特征和富有语义信息的文本特征分别由音频编码器和文本编码器提取。然后，我们建立了多模态注意机制来捕获音频和语言域之间的互补信息，以获得多模态联合表示。最后，将这个特征表示输入给密集层以生成检测结果。

4.2　对抗攻击方法

对抗攻击方法包括黑盒攻击和白盒攻击。黑盒攻击仅能得到受害者模型的输入和输出数据。Alzantot 方法[85] 是著名的语音数据黑盒攻击算法。它通过改变音频片段子集来提高攻击成功率，并且扰动不会改变人类听众对音频片段的 89% 的感知。白盒攻击能够访问受害者模型每一层的参数，并通过使用相应的梯度最小化扰动来最大化攻击的成功。Carlini & Wagner（C&W）方法[86] 是一种强大的白盒攻击方法。C&W 方法生成的对抗攻击异常样本与原始音频样本的相似度超过 99%。白盒攻击能够访问自动语音识别（ASR）模型所有层的参数，并且可以使用相应的梯度通过最小化扰动来最大化攻击的成功率。作为一种流行的白盒

攻击方法，C&W 方法利用连接主义时间分类（CTC）损失来进行扰动。

$$\begin{aligned} &\text{minimize } ||\delta||_2^2 + \alpha \cdot \ell\left(x+\delta, \pi\right), \\ &\text{such that } dB_x(\delta) < \tau, \end{aligned} \tag{4.1}$$

其中 δ 表示扰动，α 是平衡接近对抗攻击异常样本和保持原始样本之间相对重要性的参数，ℓ 表示 CTC 的损失函数，π 表示对齐方式，τ 是以分贝（dB）为单位的扰动阈值。Alzantot[85] 被选为生成对抗攻击异常样本的黑盒攻击方法。它是一种基于遗传算法的无梯度方法。与白盒攻击方法不同，Alzantot 只能访问目标语音识别模型的输入和输出。

4.3 多模态异常检测数据集生成

基于上述流行的音频攻击方法，我们分别使用它们生成两个数据集，即白盒攻击的 WiAd 数据集和黑盒攻击的 BlAd 数据集。选择 C&W 方法作为白盒攻击方法，并选择 DeepSpeech[87] 作为生成 WiAd 数据集的受害者模型。具体来说，C&W 方法攻击在 Mozilla Common Voice 数据集[88] 上执行语音识别任务的 DeepSpeech 模型。由于 Mozilla Common Voice 数据集中不同的语音时长存在较大差异，因此我们根据音频的长度将数据集分为三个子集，然后从这三个子集中随机选择一些音频作为原始音频生成的数据集。因此，我们将数据集中的音频按持续时间分为三类，以进行更详细的研究。（*i*）短音频（short audio）：持续时间在 1 到 2 秒之间。（*ii*）中等音频（medium audio）：持续时间在 3 到 4 秒之间。（*iii*）长音频（long audio）：持续时间超过 5 秒。

攻击目标是音频白盒攻击中必不可少的配置。因此，我们为三种不同的文本长度设置了攻击目标。（*i*）短目标（short target）：“This is for you”。（*ii*）中等目标（medium target）：“He needs three days answered the alchemist”。（*iii*）长目标（long target）：“It was faintly marked with transverse stripes and slightly flattened from the perfect round”。

对于三个不同持续时间的类别示例中的每一个，我们为每个类别选择了 100 个示例作为原始音频示例。然后，我们使用白盒攻击来攻击这些音频示例以获得正样本。即一个原始音频样本会被三个攻击目标中的每一个攻击一次，得到三个

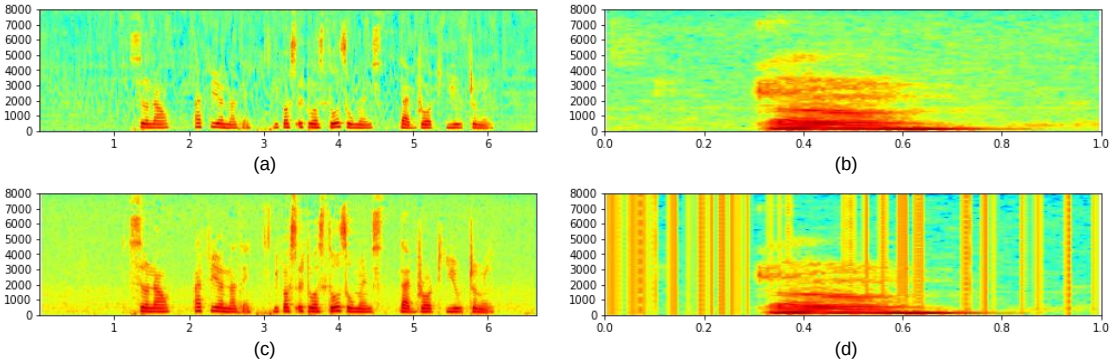


图 4.1 原始音频的音频频谱可视化及其对应的对抗攻击异常样本

正样本。

我们将使用 300 个对抗攻击异常样本作为正样本。我们为每个子集使用三个提到的目标生成了对抗攻击异常样本。我们获得了一个包含三个子集的数据集，每个子集包含 300 个正常音频样本，总共产生 300 个带有对抗攻击异常的音频样本。为了平衡正负样本，我们为每个类别选择 300 个音频示例，这与之前选择的 100 个负样本不同。据此，得到白盒攻击数据集的音频模态样本，其中包含 900 个原始音频和 900 个音频样本。然后我们依次将上述 1800 个音频示例被输入到 DeepSpeech 以生成转录集，用作输入音频的文本模态。因此，1800 个音频示例及其对应的文本构成了最终的白盒攻击数据集，即 WiAd。

我们从语音唤醒数据集（即 Google Speech Command）的 35 个关键词中选择了 10 个关键词作为 10 个类别。因为谷歌语音命令数据集中的音频长度都是一秒，所以我们使用了不同于白盒攻击的分类方法。对于每个关键字，我们从中选择了 20 个不同说话者的示例。与白盒方法攻击不同，黑盒攻击使用其他九个关键词作为攻击目标，从而为每个关键词生成九个对抗攻击异常样本。因此，黑盒攻击生成的数据集包含 1800 个带有对抗攻击异常的音频样本。同样，为了平衡正负样本，我们选择了与之前不同的 1800 个音频样本作为数据集中的负样本。数据集 BlAd 由 3600 个包含正、负音频样本的音频组成。与在 WiAd 中生成文本模态数据一样，我们随后将上述 3600 个音频示例依次被输入到 DeepSpeech 以获得转录集，将其视为输入音频的文本模态。因此，3600 个音频示例及其对应的文本构成了最终的黑盒攻击数据集，即 BlAd。

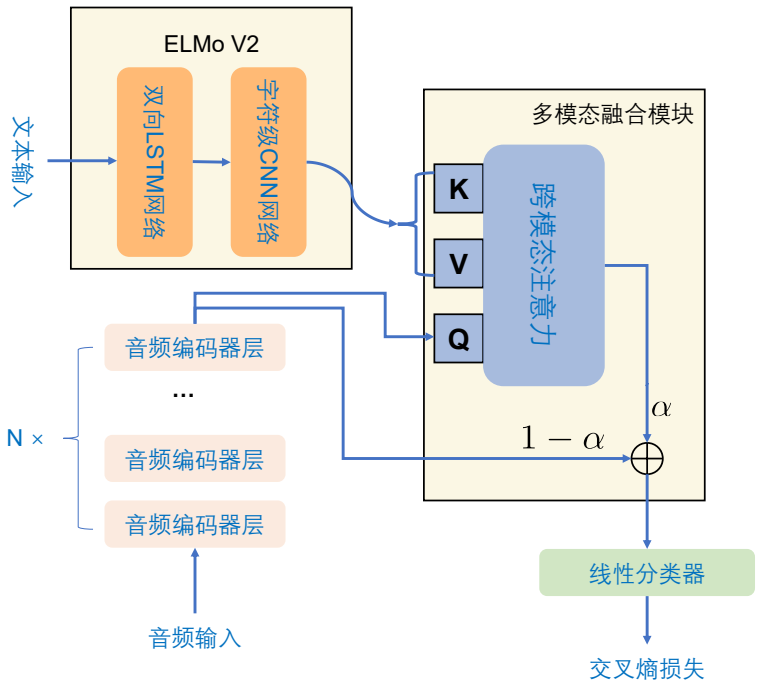


图 4.2 MDFT 的整体结构图

图 4.1通过四个频谱图说明了两种攻击方法的正常样本和带有对抗攻击异常的样本之间的差异。(a) 是原始音频，本文使用白盒攻击（C&W 方法）和长目标来攻击（a）以获得对抗攻击异常样本（b）。(b) 是另一个原始音频。黑盒攻击 Alzantot 方法结合目标词“stop”生成对抗攻击异常样本（d）。

为了生成数据集，我们使用具有 3 个 NVIDIA RTX 2080Ti GPU 的 Linux 服务器。对于白盒和黑盒攻击方法，我们保留开源代码中的默认配置来生成对抗攻击异常样本。在生成速度方面，针对一秒音频文件生成对应于白盒和黑盒攻击的对抗攻击异常样本的平均时间分别约为 1 分钟和 5 分钟。

4.4 模型架构

如图 4.2 所示，MDFT 使用 ELMo v2[89] 来生成文本的高维特征表示，并使用 Transformer 编码器来提取音频表示。

4.4.1 提取文本特征

为了利用 ASR 模型生成的转录文本中的信息，我们利用 AllenNLP 提出的 NLP 中著名的词嵌入方法 ELMo v2 来实现文本的词嵌入。ELMo v2 使用双向语言模型生成词嵌入。双向语言模型由两个堆叠的循环网络层组成，用于提取隐藏表示并使用 CNN 网络将单词转换为嵌入向量。双向语言模型的字符级输入有助于捕获单词间的潜在相关性。与传统的词嵌入不同，ELMo v2 通过参考输入句子中词的上下文来计算词嵌入，不同上下文中的相同词对应不同的词嵌入。

4.4.2 提取音频特征

给定音频频谱图 $X \in \mathbb{R}^{T\times F}$，它的长度为 T，频率维度为 F。音频频谱图通过线性矩阵 $W_{input} \in \mathbb{R}^{F\times d}$ 投影到维度为 d 的特征。在这项工作中，我们将 F 设置为 40。为了学习位置信息，我们添加了一个可学习的位置嵌入 X_p。最终的音频输入表示如下：

$$X_{input} = XW_{input} + X_{\text{position}}, \tag{4.2}$$

然后将音频输入特征输入到音频特征提取模块，该模块由 L 层 Transformer 编码器组成。每个编码器由多头注意力（M-Att）和多层感知器（MLP）组成。在 l 层的 Transformer 编码器中，让 $W_h^Q, W_h^K, W_h^V \in \mathbb{R}^{d\times d_h}$ 是计算查询（Q）、键（K）和值（V）的权重矩阵，其中 d_k 是查询和键的维度，d_v 是值的维度。多头注意力在时间 t 处同时将输入 Z 转换为 Q、K、V 矩阵的 H 组。对于 head h ($h = 1, \cdots, H$)，查询、键和值矩阵计算分别为 $Q_h = X_l W_h^Q, K_h = X_l W_h^K$ 和 $V_h = X_l W_h^V$。经过前面的变换，Q_h、K_h 和 V_h 之间的缩放点积注意力用于计算每个 head h 对应的输出矩阵：

$$\text{Att}_h(X_l) = \text{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right) V_h. \tag{4.3}$$

多头自注意力机制通拼接联运算和线性投影整合不同 head 的输出，

$$\text{M-Att}(X_l) = [\text{Att}_1(X_l); \text{Att}_2(X_l); ...; \text{Att}_H(X_l)]W_P, \tag{4.4}$$

其中 $W_P \in \mathbb{R}^{H\times d_h\times d}$ 是权重矩阵。

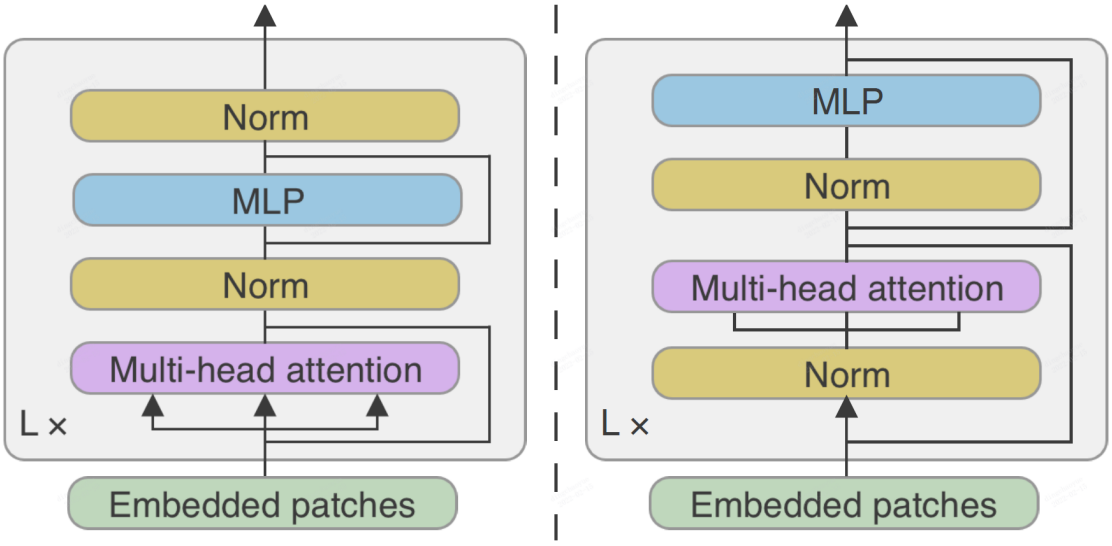


图 4.3 PostNorm Transformer **（左）和** Prenorm Transformer **（右）的编码器架构**

如图 4.3 所示，PostNorm Transformer 和 PostNorm Transformer 是两种不同的 Transformer 架构。MDFT 使用 PostNorm Transformer 作为默认设置。PostNorm Transformer 的多头注意力和 MLP 块放置在层归一化（LN）之前。与 PreNorm Transformer 不同，LN 位于前面。与大多数研究一样，我们在 MLP 中使用非线性激活函数 GELU。

综上所述，第 l 层的 Transformer 编码器输出可以表示为：

$$\tilde{X}_l = \text{Linear}(\text{M-Att}(X_{l-1}) + X_{l-1}), \tag{4.5}$$

$$X_l = \text{Linear}(\text{MLP}(\tilde{X}_l) + \tilde{X}_l), \tag{4.6}$$

我们将音频特征谱视为视觉 Transformer 中的图像，然后将其切分为多个 patch。与视觉 Transformer 不同，这里的自注意力是基于具有时域信息的频谱 patch 之间计算的，而不仅仅是具有空间信息的图像 patch，以更好地捕获音频中的时间依赖性。

4.4.3 融合文本和音频模态

我们引入了跨模态注意，旨在将特征从音频模态映射到文本模态。与点积注意力一样，Q、K 和 V 分别表示查询、键和值的矩阵。K 和 V 是使用文本模态的

特征构建的，Q 是从音频模态的特征转化而来的。跨模态注意力被表述为：

$$Y = \text{Softmax}(KQ^T)V, \tag{4.7}$$

其中 Y 是跨模态注意力得到的表示。然后我们融合 Y 和音频表示 X_{audio}，即 Transformer 编码器最后一层的输出，结合平衡因子 α，得到融合后的多模态表示：

$$Z = \text{Concat}(\alpha \cdot Y, (1-\alpha) \cdot X_{audio}), \tag{4.8}$$

其中 “Concat” 表示拼接操作。然后将多模态特征 Z 被输入到分类器得到检测结果。根据一些实验，我们凭经验将 α 设置为 0.3。我们使用主流的交叉熵损失结合 Adam 优化器来训练 MDFT。

4.5 实验

我们通过实验来证明 MDFT 在两个生成的数据集（WiAd 和 BlAd）上的有效性，并将其与 MDFT 的变体进行比较，即 MDFT（音频模态），它删除了与文本相关的组件，并且仅使用音频特征而不是多模态特征. 此外，我们还进行了几项消融研究，以进一步分析提出的方法。

4.5.1 实验设置

数据集 WiAd 包含来自 Mozilla Commonvoice 数据集的三个类别的正常示例子集。C&W 方法白盒攻击方法对 Mozilla Common Voice 数据集执行白盒攻击。数据集 BlAd 包含来自 Google Speech Command 数据集的正常示例的子集。Alzantot 作为黑盒攻击方法对 Google Speech Command 数据集的执行黑盒攻击。请注意，对于 WiAd 和 BlAd 数据集，我们以 7:3 的比例来划分所生成的数据样本，从而得到训练和测试集，训练和测试集中没有来自相同源音频的音频样本。

如表 4.1 所示，我们设计了六种不同的实验配置，旨在研究检测模型在不同训练环境下对可见和不可见对抗攻击的性能。此外，我们希望验证用多种类型的攻击数据训练模型可以提高检测模型对不同攻击的性能。

4.5.2 实验结果

在本次实验中，设计了一些评估设置以观察白盒和黑盒攻击模型之间的差异。我们使用单独或联合的方式来使用 WiAd 和 BlAd 数据集，以训练和评估我们的模型。表 4.1 说明了不同的配置并显示了模型准确率。

实验结果表明，与 MDFT（音频模态）相比，所提出的方法显着提高了检测性能。多模态模型取得了优于单模态模型的实验结果，归功于充分使用了来自音频和文本模态的信息，并且多模态模型比单模态模型具有更好的鲁棒性。训练和测试阶段使用相同攻击类型的数据（第 1 行和第 4 行）可以获得更高的准确度（超过 98.5%）。在此基础上将训练集扩展到联合训练集（第 5 行和第 6 行）会导致准确度略有下降（超过 95.5%）。这些结果表明，即使在训练示例中存在随机噪声的情况下，MDFT 也可以很好地学习以抵消训练集中攻击的扰动。可以看出，MDFT 模型在训练和测试集属于相同攻击类型时，展现出较高的准确率。与黑盒攻击产生的对抗扰动相比，白盒攻击产生的对抗扰动具有比黑盒攻击更容易学习的特定模式。从这些实验中，我们可以得出结论，当训练和测试数据集具有相同的攻击类型时，所提出的 MDFT 可以获得较好的性能。

我们研究了不匹配的训练集和测试集（第 2 行和第 3 行），它们表现出相反的结果。（Train WiAd, Test BlAd）的准确率低于相同攻击类型的训练和测试集，但仍然达到 83% 以上的准确率。然而，（Train BlAd, Test WiAd）的准确率只有 50% 左右。这一结果表明，白盒攻击和黑盒攻击具有不同的性质。多条件训练（Train WiAd & BlAd, Test WiAd）和（Train WiAd & BlAd, Test BlAd）的结果也非常高。因此，MDFT 模型显示了学习更多种类的扰动的能力。可以从这部分结果中提取的另一个知识是，如果我们通过来自不同攻击方法的对抗攻击异常样本来训练检测模型，MDFT 就有能力检测不同攻击类型的对抗攻击异常样本。

实验（训练 WiAd，测试 WiAd）的详细结果如表 4.2 所示。很明显，当我们得到源音频长度与攻击目标长度之间的差值最大时，检测准确度较高。此外，当这两个长度更接近时，我们的准确度较低。

我们进行了实验来研究 MDFT 在未知攻击目标样本上的表现。如表 4.3 所示，MDFT 通过使用 WiAd 的训练集和测试集的不同组合始终表现出较高的准确

表 4.1 不同训练和测试配置下的 F1 比较

Train	Test	MDFT (音频模态) %	MDFT %
WiAd	WiAd	99.33 ± 0.37	99.45 ± 0.39
WiAd	BlAd	82.05 ± 0.33	83.27 ± 0.35
BlAd	WiAd	49.69 ± 0.48	50.39 ± 0.42
BlAd	BlAd	98.41 ± 0.36	98.60 ± 0.38
WiAd, BlAd	WiAd	95.28 ± 0.36	95.77 ± 0.29
WiAd, BlAd	BlAd	96.49 ± 0.34	97.13 ± 0.35

表 4.2 在 WiAd 数据集的准确率评估

Targets \ Length	Short	Medium	Long
Short	98.72	99.73	99.97
Medium	99.63	98.50	99.65
Long	99.89	99.26	98.85

表 4.3 使用未知目标长度生成的异常样本测试 MDFT 的准确率

实验	Accuracy %
Train Short & Medium, Test Long	99.56
Train Short & Long, Test Medium	99.87
Train Medium & Long, Test Short	99.34

率，这证明了 MDFT 模型结合多模态数据对不同类型对抗攻击具有鲁棒性。

4.5.3 消融实验

我们通过改变输入到 Transformer 编码器（Train BlAd，Test BlAd）的音频频谱特征的 patch 大小来研究准确率的变化。结果说明模型在时域宽度为 1 时取得了最佳的性能，如图 4.4所示。这表明时域卷积和时域注意力机制有助于捕获频谱中的时间依赖性。音频编码器的第一个线性投影层可以被认为是一个时域卷积，卷积核大小为 (40, 1)，步幅大小为 1。因此 (40, 1) 的内核大小是我们模型的默认设置。我们还比较了 PreNorm 和 PostNorm 的性能，实验结果表明 PostNorm 更有利于提高对抗攻击异常样本检测的性能。这与之前关于关键字识别任务的工作得出的结论相同[90]。

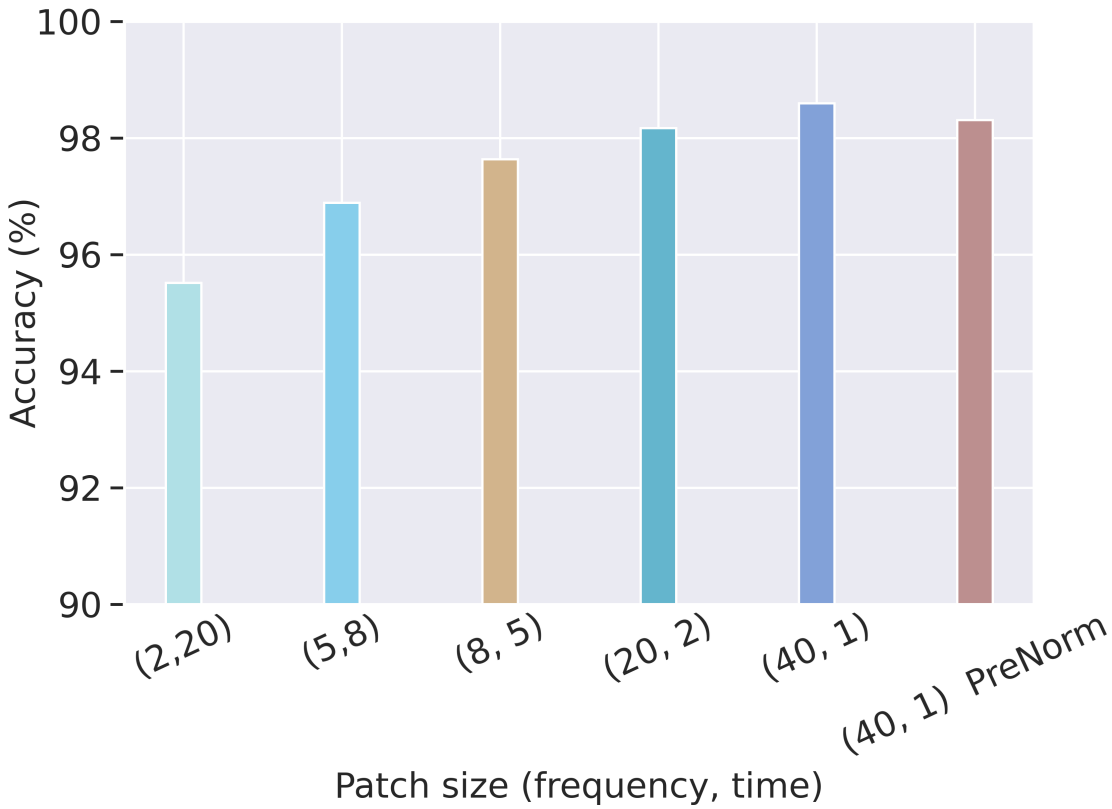


图 4.4 Blad 数据集上 MDFT 在不同 patch 大小下的准确率

4.6 本章小节

本章引入了一种新颖的方法来使用最先进的白盒和黑盒对抗攻击方法生成时序语音数据上的异常样本。然后，我们设计了一个基于视觉 Transformer 的多模态模型（MDFT）来检测由对抗攻击产生的异常样本。此外，我们通过不同的训练和测试设置使用两个生成的数据集来评估模型的性能。结果显示 MDFT 在两个数据集上都优于其对应的单模态变体模型。

第五章　基于图注意力网络的多模态时间序列异常检测

上一章研究了在语音和文本这两种非对齐模态上的多模态时间序列异常检测，本章将研究在对齐模态上的时间序列异常检测，这涉及到多个模态之间空间依赖关系的构建。多模态时间序列异常检测通常用于监控工业设备和信息技术系统（即实体）中传感器的多种模态（例如温度、速度和功率），并且来自每个传感器的数据流被视为单变量时间序列。多模态时间序列数据可以用于检测具有复杂时空依赖关系的异常，这在独立监测每个模态时并不明显[67]。此外，在被监测的实体部分或完全中断之前，及时检测异常有助于用户进行故障排除。

5.1　研究动机

多模态时间序列（MTS）异常检测对于保持工作设备（例如水处理系统和航天器）的安全性和稳定性至关重要。尽管最近的深度学习方法在异常检测方面取得了不错的表现，可是它们没有明确地捕获不同模态间的时空关系，从而导致更多的假阴性和假阳性。在本文中，我们提出了一种多模态时空图注意力网络（MST-GAT）来解决这个问题。MST-GAT 首先采用多模态图注意力网络（M-GAT）和时间卷积网络来捕获多模态时间序列中的时空相关性。具体来说，M-GAT 使用一个多头注意力模块和两个关系注意力模块（即模态内和模态间注意力）来显式地建模模态相关性。此外，之前的工作已经证明，基于重建和基于预测的模型在大多情况下具有互补性[32]。因此，我们提出了一个联合网络来整合这两种范式的优点。实验结果表明，MST-GAT 优于最先进的对比方法。进一步的分析表明，MST-GAT 通过定位最可能导致异常的单变量时间序列来增强检测到的异常的可解释性。

5.2　问题建模

多模态时间序列由属于同一实体的多个模态的时间序列组成，每个模态可以包含一个或多个时间序列。MTS 异常检测模型旨在检测时间戳级别的异常。时间序列异常检测通常被认为是一项无监督的任务，我们假设训练阶段的数据不

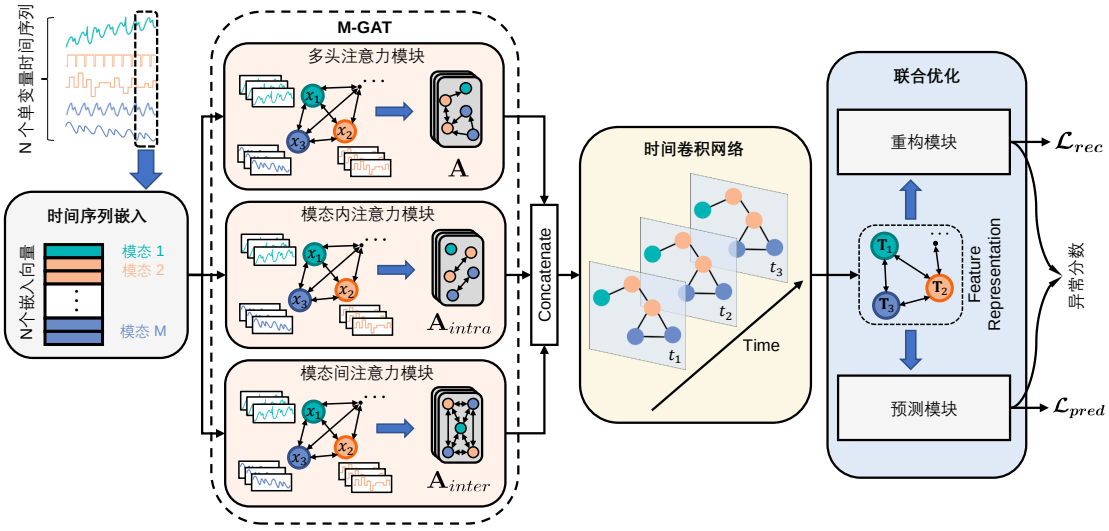


图 5.1 MST-GAT 模型的整体架构

存在异常。我们将这个问题表述如下。在训练阶段，多模态时间序列数据由具有 T 个时间戳的 N 个单变量时间序列组成，即 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T] \in \mathbb{R}^{N \times T}$，每个单变量时间序列包括 M 个模态（$1 \leq M \leq N$）。在时间 t ($t \leq T$) 的多模态时间序列观测被记为 $\mathbf{x}_t = [x_{1,t}^{m_1}, x_{2,t}^{m_2}, \cdots, x_{N,t}^{m_N}]^\top$，其中 $x_{i,t}^{m_i}$ ($m_i \in \{1, 2, \cdots, M\}$) 表示数据来自第 i 个单变量时间序列，属于第 m_i 个模态。

在推理阶段，模型旨在检测多模态时序数据 $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \cdots, \tilde{\mathbf{x}}_{T'}] \in \mathbb{R}^{N \times T'}$ 中的异常，这个序列属于同样的 N 个单变量时序传感器产生的。多模态时间序列的长度表示为 T'。模型需要产生一个检测结果 $\tilde{\mathbf{y}} = [\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_{T'}] \in \mathbb{R}^{T'}$，其中 $\tilde{y}_i \in \{0, 1\}$ 和 $\tilde{y}_i = 1$ 表示 $\tilde{\mathbf{x}}_i$ 是一个异常。

5.3 模型架构

MST-GAT 被表述为一种图结构，将每个传感器视为一个节点，将它们的关系视为边，它为整个多模态时间序列之间的复杂多模态相关性和时空关系建模。如图 5.1所示，MST-GAT 的架构涉及四个部分：

- **图结构学习**。它使用时间序列嵌入（表征每个时间序列的固有属性）来学习空间维度上的图结构；
- **多模态图注意力网络（M-GAT）**。它通过多头（multi-head）注意力模块和

附加的关系注意力模块（模态内（intra-modal）和模态间（inter-modal）注意力）显式捕获模态内和模态间关系；

- **时间卷积网络**。它利用时间轴上的卷积结构来捕获时间序列的时间依赖性；
- **联合优化和异常分数**。MST-GAT 优化重建和预测目标，然后用异常分数识别异常。异常分数进一步用于解释检测到的异常。

5.3.1 图结构学习

在多模态时间序列中，不同的模态可以表现出多种属性，并且它们以复杂的方式相互关联。因此，我们希望对每个时间序列使用灵活的表示来捕获多种模态之间的潜在相关性。在本文中，我们引入时间序列嵌入来构建多头注意力模块的灵活图结构。给定一个用于多模态时序数据的图结构 $\mathcal{G}$，它包含 N 个节点，其中每个节点存储当前时刻单变量时间序列的表示。节点之间的边表示不同时间序列之间的依赖关系。节点 i 的邻居节点集合记为 $\mathcal{N}_i = \{j \mid \mathbf{A}_{ij} > 0\}$。我们为节点 i 定义嵌入 $\mathbf{v}_i \in \mathbb{R}^d$ 的时间序列来表征其固有属性，其中 $i \in \{1, 2, \cdots, N\}$，$d$ 是嵌入维度。时间序列嵌入进一步用于构建多头注意力模块的邻接矩阵 $\mathbf{A}$。邻接矩阵可以表示为：

$$\mathbf{A}_{ij} = \mathbb{1} \quad \{j \in \text{TopK}\left(\{e_{ik} | k \in \mathcal{C}_i\}\right)\}, \tag{5.1}$$

$$e_{ij} = \text{sim}(\mathbf{v}_i, \mathbf{v}_j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \times \|\mathbf{v}_j\|}, \tag{5.2}$$

其中公式 5.1 表示如果 $j \in \mathbf{C}_i$，则 $\mathbf{A}_{ij} = 1$，否则 $\mathbf{A}_{ij} = 0, \mathcal{C}_i = \{1, 2, \cdots, N\}$ 是候选集，sim$(\cdot)$ 是余弦相似度，TopK 是从候选集中选出值最大的 K 个索引的操作。具体来说，我们首先计算 e_{ij}，即时间序列嵌入向量间的余弦相似度。接着，我们从候选集中选择前 K 个相似的节点来构造一个稀疏有向图，参数 K 控制图结构的稀疏度。

5.3.2 空间维度中的 M-GAT

对于用于训练的时间序列 $\mathbf{X}$，我们使用长度为 w 的滑动窗口在每个时间步产生一个固定长度的输入。我们将 $\hat{\mathbf{X}}$ 定义为 M-GAT 在 t 时的输入：

$$\hat{\mathbf{X}} = [\mathbf{x}_{t-w+1}, \mathbf{x}_{t-w+2}, \cdots, \mathbf{x}_t] \in \mathbb{R}^{N \times w}, \tag{5.3}$$

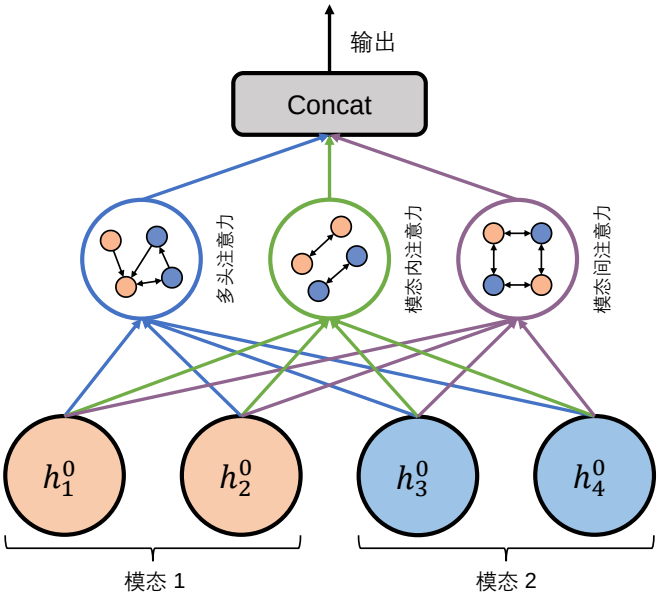


图 5.2 M-GAT 网络的架构

我们以同样的方式处理测试时间序列 $\tilde{\mathbf{X}}$。

假设 $\mathbf{H}^l$ 表示 M-GAT 在 l 层的特征表示。M-GAT 的初始输入为 $\mathbf{H}^0 = (\hat{\mathbf{X}}\mathbf{W}_{in}) \,||\, \mathbf{V}$，其中 $\mathbf{W}_{in} \in \mathbb{R}^{w \times d}$ 是输入数据的可学习变换，$||$ 表示拼接操作。所提出的 M-GAT 的架构如图 5.2 所示。它由三个注意力模块组成，即多头注意力、模态内和模态间注意力。多头注意力模块专注于对多模态时间序列之间的空间关系进行建模，而模态内和模态间注意力模块专注于捕获不同时间序列之间的多模态相关性。

多头注意力模块通过聚合其邻居的表示来更新每个节点的特征表示，公式如下：

$$h_{att_i}^{l+1} = ||_{s=1}^{S} \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{ls} \mathbf{W}_{att}^{ls} h_j^l, \tag{5.4}$$

$$\alpha_{ij}^{ls} = \text{attention}(i, j), \tag{5.5}$$

其中 $h_j^l \in \mathbf{H}^l$ 是第 j 个节点在第 l 层的表示，$h_{att_i}^{l+1}$ 是 $l+1$ 层第 i 个节点的特征，S 表示 attention head 的个数，$||$ 表示拼接操作，α_{ij}^{ls} 表示由节点 i 和节点 j 之间在第 l 层第 s 个注意力头处计算的注意力分数，$\mathbf{W}_{att}^{ls}$ 表示 l 层第 s 个注意力 head 的权重矩阵，$\text{attention}(i, j)$ 表示缩放的点积注意力[9]。

以前的方法使用 GAT 根据邻接矩阵来聚合相邻节点的表示[51]。然而，这些方法未能考虑时间序列的多模态依赖性，这可能会丢失一些关于多模态相关性

的重要信息。直观地说，具有不同关系的相邻节点应该对中心节点有不同的影响。我们扩展了多头注意力模块，增加了两个关系注意力，即模态内和模态间注意力，以更有效地整合多模态时序特征。这两个关系注意力模块的邻接矩阵定义如下：

$$\mathbf{A}_{intra}^{ij} = \mathbb{1}\left\{j \in \mathbf{C}_{intra}^{i}\right\}, \tag{5.6}$$

$$\mathbf{A}_{inter}^{ij} = \mathbb{1}\left\{j \in \mathbf{C}_{inter}^{i}\right\}, \tag{5.7}$$

其中 $\mathbf{C}_{intra}^{i} = \{j|m_i = m_j\}$ 和 $\mathbf{C}_{inter}^{i} = \{j|m_i \neq m_j\}$ 是候选集。也就是说，$\mathbf{C}_{intra}^{i}$ 包含与节点 i 属于相同模态的节点，而 $\mathbf{C}_{inter}^{i}$ 由属于与节点 i 不同的模态的节点。一般来说，$\mathbf{C}_{intra}^{i}$ 和 $\mathbf{C}_{inter}^{i}$ 由公式 5.6和公式 5.7计算，但如果 $|\mathbf{C}_{intra}^{i}| > K$，则通过 TopK 操作选择最大的 K 个余弦相似度值的索引。类似地，如果 $|\mathbf{C}_{inter}^{i}| > K$，TopK 运算也将应用于 $\mathbf{C}_{inter}^{i}$：

$$\mathbf{C}_{intra}^{i} = \left\{\text{TopK}\left(\left\{e_{ik}|k \in \mathbf{C}_{intra}^{i}\right\}\right)\right\}, \tag{5.8}$$

$$\mathbf{C}_{inter}^{i} = \left\{\text{TopK}\left(\left\{e_{ik}|k \in \mathbf{C}_{inter}^{i}\right\}\right)\right\}. \tag{5.9}$$

然后，我们使用这两个关系注意力模块来明确捕获时序数据之间的多模态相关性。我们将模态内注意力模块的特征计算为：

$$h_{intra_i}^{l+1} = \sum_{j \in \mathcal{N}_{intra_i}} \beta_{intra_i}^{lj} \mathbf{W}_{intra}^{l} h_j^{l}, \tag{5.10}$$

$$\beta_{intra_i}^{lj} = \frac{\exp(g_{intra_i}^{lj})}{\sum_{k \in \mathcal{N}_{intra_i}} \exp(g_{intra_i}^{lk})}, \tag{5.11}$$

$$g_{intra_i}^{lj} = \sigma(\text{ReLU}((\mathbf{V}_i||\mathbf{V}_j)\mathbf{W}_{intra1}^{l} + b_{intra1}^{l})\mathbf{W}_{intra2}^{l}), \tag{5.12}$$

其中 $h_{intra_i}^{l+1}$ 是 $l+1$ 层第 i 个节点的特征，$\mathcal{N}_{intra_i} = \{j \mid \mathbf{A}_{intra}^{ij} > 0\}$ 表示节点 i 的模态内邻居集，$\beta_{intra_i}^{lj}$ 表示节点 i 和节点 j 之间的第 l 层的注意力分数，$\mathbf{W}_{intra}^{l}$，$\mathbf{W}_{intra1}^{l}$ 和 $\mathbf{W}_{intra2}^{l}$ 是第 l 层的权重矩阵，b_{intra1}^{l} 是第 l 层的偏置向量。$h_{inter_i}^{l+1}$ 的计算类似于 $h_{intra_i}^{l+1}$ 的计算方式，$\mathcal{N}_{inter_i} = \{j \mid \mathbf{A}_{inter}^{ij} > 0\}$ 是节点 i 的模态间邻居集。我们将 $h_{att_i}^{l+1}$、$h_{intra_i}^{l+1}$ 和 $h_{inter_i}^{l+1}$ 合并到最终表示 h_i^{l+1}：

$$h_i^{l+1} = \text{ReLU}(\mathbf{W}_{out}^{l+1} o_i^{l+1} + b_{out}^{l+1}), \tag{5.13}$$

$$o_i^{l+1} = h_{att_i}^{l+1} \,||\, h_{intra_i}^{l+1} \,||\, h_{inter_i}^{l+1}, \tag{5.14}$$

其中 h_i^{l+1} 是节点 i 在第 $l+1$ 层的最终表示，$\mathbf{W}_{out}^{l+1}$ 是第 $l+1$ 层的权重矩阵，b_{out}^{l+1} 是第 $l+1$ 层的偏置向量，$||$ 表示拼接，o_i^{l+1} 是通过拼接第 $l+1$ 层的中间特征 $h_{att_i}^{l+1}$、$h_{intra_i}^{l+1}$ 和 $h_{inter_i}^{l+1}$ 而得到的。

5.3.3 时间维度中的卷积

多模态图注意力在空间维度上捕获每个节点的邻居信息，而时间卷积网络在时间维度上应用时间卷积来捕获时序数据间的时间依赖关系。时间卷积网络的输入是图级别表示 $\mathbf{H}^{L_{gat}}$，其中 L_{gat} 是 M-GAT 中的层数。时间级别表示计算如下：

$$\mathbf{T}^{l+1} = \text{ReLU}\left(\Phi * \left(\text{ReLU}\left(\mathbf{T}^{l}\right)\right)\right), \tag{5.15}$$

其中 $\mathbf{T}^{l+1}$ 表示第 $l+1$ 层的时间级别表示，$*$ 是卷积操作，Φ 是内核大小，ReLU 是激活函数。时间卷积网络通过合并来自相邻时间片的信息来更新特征，因此它可以很好地捕获时序数据中的时间依赖关系。

5.3.4 联合优化

重建和预测模块的输入是时间卷积网络的输出。为清楚起见，我们设置 $\mathcal{X}_t = \mathbf{T}^{L_{tem}}$ 作为重建和预测模块在时间 t 的输入，其中 L_{tem} 表示时间卷积中的层数。MST-GAT 结合了重构和预测模块的优点。重构模块捕获整个时间序列的数据分布，预测模块预测下一个时间戳的观测值。我们用两个任务优化 MST-GAT，即重建和预测任务。损失函数包含两个优化目标，定义为：

$$\mathcal{L} = \gamma_1 \mathcal{L}_{rec} + (1 - \gamma_1) \times \mathcal{L}_{pred}, \tag{5.16}$$

其中 $\mathcal{L}_{rec}$ 表示重建模块的损失函数，$\mathcal{L}_{pred}$ 表示预测模块的损失函数，γ_1 是平衡重建和预测模块的超参数。

重构模块的目标是学习输入数据的重构概率。受 OmniAnomaly[46] 的启发，本文利用变分自编码器（VAE）来得到 $\mathcal{G}_t$ 的重构表示。给定输入 $\mathcal{X}_t$，VAE 使用条件分布 $p_\psi(\mathcal{X}_t|z_t)$ 来重构 $\mathcal{X}_t$，其中 z 是潜在表示。训练重建模块的目标是最大化 z_t 的后验分布：

$$p_\psi(z_t|\mathcal{X}_t) = p_\psi(\mathcal{X}_t|z_t)p_\psi(z_t)/p_\psi(\mathcal{X}_t), \tag{5.17}$$

其中 $p_\psi(\mathcal{X}_t)$ 是 $\mathcal{X}_t$ 的重构概率。令 $p_\psi(\mathcal{X}_t) = \{p_i | i = 1, 2, \cdots, N\}$，其中 p_i 表示第 i 个单变量时间序列的重建概率。重建概率 $p_\psi(\mathcal{X}_t)$ 可以定义如下：

$$p_\psi(\mathcal{X}_t) = \int p_\psi(z_t) p_\psi(\mathcal{X}_t | z_t) dz_t. \tag{5.18}$$

上面的方程很难计算，我们需要一个新的模型 $q_\rho(z_t|\mathcal{X}_t)$ 来逼近 $p_\psi(z_t|\mathcal{X}_t)$。给定编码器模型 $q_\rho(z_t|\mathcal{X}_t)$ 和解码器模型 $p_\psi(\hat{\mathcal{X}}_t|z_t)$，重建损失定义为：

$$\begin{aligned} \mathcal{L}_{rec} = &-\mathrm{E}_{q_\rho(z_t|\mathcal{X}_t)}[log p_\psi(\mathcal{X}_t|z_t)] \\ &+ D_{KL}(q_\rho(z_t|\mathcal{X}_t) || p_\psi(z_t)), \end{aligned} \tag{5.19}$$

其中 $\mathrm{E}_{q_\rho(z_t|\mathcal{X}_t)}[log p_\psi(\mathcal{X}_t|z_t)]$ 表示 $\mathcal{X}_t$ 的对数似然期望。D_{KL} 代表 KL 散度。负的 $\mathcal{L}_{rec}$ 是对 $\log p_\psi(\mathcal{X}_t)$ 下限的估计。

预测模块使用 $\mathcal{X}_t$ 来预测下一个时间戳的观察结果。我们使用多层感知机（MLP）网络作为时间卷积网络后的预测模块。预测损失可以定义为：

$$\mathcal{L}_{pred} = \frac{1}{T-w} \sqrt{\sum_{i=1}^{N} (x_{i,t+1} - \hat{x}_{i,t+1})^2}, \tag{5.20}$$

其中 $x_{i,t+1}$ 表示第 i 个时间序列在 $t+1$ 时刻的真实值，$\hat{x}_{i,t+1}$ 是在 $t+1$ 时刻第 i 个时间序列的预测值。

5.3.5 异常分数和推理

在每个时间戳，重建模块和预测模块分别生成重建概率 p_i 和预测 $\hat{\mathbf{x}}_i$，其中 $\hat{\mathbf{x}}_i$ 表示第 i 个单变量时间序列的预测值。MST-GAT 的异常分数平衡了这两个模块的权重。每个时间戳的最终异常分数是每个时间序列的异常分数之和。具体来说，异常分数表示为：

$$\text{score} = \sum_{i=1}^{N} \frac{(1 - p_i) + \gamma_2 \times (\mathbf{x}_i - \hat{\mathbf{x}}_i)^2}{1 + \gamma_2}, \tag{5.21}$$

其中 $\mathbf{x}_i$ 和 $\hat{\mathbf{x}}_i$ 分别是真实值和预测值，γ_2 是为平衡两个模块而引入的超参数，由验证集选择。在推理阶段，检测规则是如果某个时间戳的异常分数大于定义的异常阈值，则将该时间戳标记为“异常”，否则标记为“正常”。我们采用 peaks-over-threshold（POT）算法[84] 来选择验证集上的异常阈值。最后，在 algorithm 1 中总结了 MST-GAT 的整体训练和推理流程。

Algorithm 1 MST-GAT 的训练和推理过程

训练流程

输入: 多模态训练时间序列 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]$，训练 epoch I, 批量大小 M 和超参数 γ_1, γ_2。

随机初始化参数 W_{model}（W_{model} 包括 MST-GAT 中所有可学习的参数）;
for epoch $i \in 1, 2, \ldots, I$ **do**
    通过 M-GAT 在空间维度上计算 $\mathbf{H}^{L_{gat}}$; // 公式 5.13
    通过时间卷积计算时间维度的 $\mathbf{T}^{L_{tem}}$; // 公式 5.15
    通过重建模块计算重建概率; // 公式 5.18
    通过预测模块计算预测值;
    最小化联合损失函数以优化 W_{model}; // 公式 5.16
end for
return 优化后的模型参数 W_{model}。

推理流程

输入: 多模态测试时间序列 $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \cdots, \tilde{\mathbf{x}}_{T'}]$、模型参数 W_{model} 和超参数 γ_1、γ_2。

for each $\tilde{\mathbf{x}}_i$ **do**
    计算 $\tilde{\mathbf{x}}_i$ 的异常分数; // 公式 5.21
    if 异常分数 > 阈值 **then**
        $\tilde{\mathbf{x}}_i =$ “一个异常点”;
    else
        $\tilde{\mathbf{x}}_i =$ “一个正常点”;
    end if
end for
return 预测的标签列表 $\tilde{\mathbf{X}}$。

表 5.1 四个多模态时间序列数据集的详细统计信息

数据集	特征个数	模态个数	训练	测试	异常点 (%)
MSL	27	8	58317	73729	10.72
SMAP	55	12	135183	427617	13.13
SWaT	51	8	496800	449919	11.98
WADI	123	8	1048571	172801	5.99

5.4 实验

在本节中，我们设计了多个实验来证明 MST-GAT 的有效性。我们首先介绍四个常用的公共数据集。接下来，我们在这些数据集上评估 MST-GAT，并表明 MST-GAT 的性能优于现有的异常检测方法。然后，我们对所提出模型的关键组件进行消融研究。最后，我们通过案例研究证明了 MST-GAT 的可解释性。

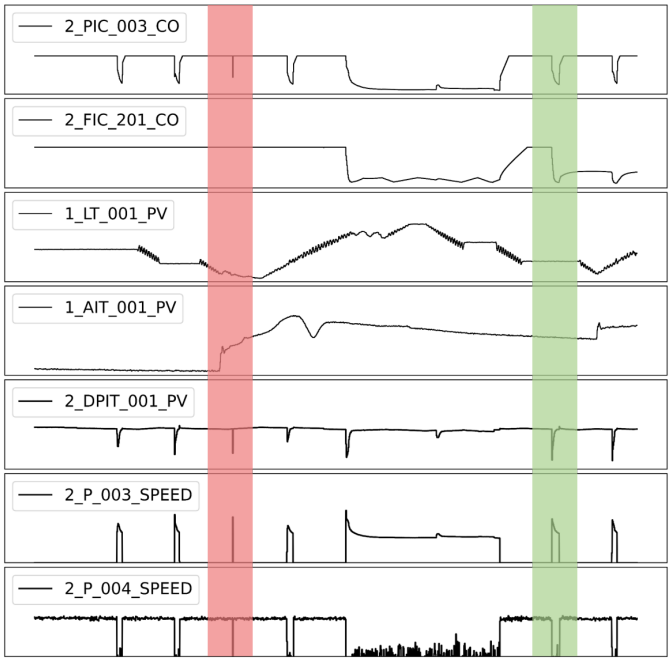


图 5.3 多模态时间序列数据的示例，红色阴影区域（左）表示异常值，绿色阴影区域（右）表示正常值，相同后缀的时间序列属于同一个模态

5.4.1 数据集介绍

本实验涉及四个基准数据集。我们在表 5.1 中展示统计数据，并在下面简要介绍它们。火星科学实验室漫游车（MSL）[50] 和土壤水分主动被动卫星（SMAP）[50] 是从航天器获取的真实世界数据集。这些数据集由 NASA 的专家注释。每个数据集都包括预先分割的训练和测试集。训练集是从正常数据中收集的，测试集包括标记的异常。安全水处理（SWaT）[91] 数据集是从具有 51 个传感器的按比例缩小的水处理试验台收集的，包括 7 天的正常运行和 4 天的模拟攻击场景。这些模拟攻击包括不同的持续时间和不同的攻击目标。配水系统（WADI）[92] 是从包含 123 个传感器的简化配水试验台获取的数据集。训练集包含正常操作下两周产生的数据，测试集包括攻击场景下两天产生的数据。

图 5.3 展示了 WADI 数据集上的多模态时间序列示例。在绿色阴影区域（右），除 2_FIC_201_CO 和 1_LT_001_PV 外，所有传感器值都有明显波动，但系统仍处于正常状态，这些时间序列保持一致趋势。然而，在红色阴影区域（左）部分，传感器 1_AIT_001_PV 与其他单变量时间序列相比表现出不一致的模式，表明

该传感器可能存在潜在的问题。

5.4.2 对比模型

我们将 MST-GAT 与八种流行的 MTS 异常检测方法进行比较，包括：

- **PCA:** PCA 旨在将高维特征投影到低维表示，投影的重构误差用于计算异常分数[93]。
- **AE:** 自动编码器包括一个编码器和一个解码器，并使用重构误差来检测异常[94]。编码器将输入数据压缩成一个隐藏向量，解码器使用该向量重构输入数据。
- **DAGMM:** 深度自编码高斯模型融合自编码器和高斯混合模型来得到低维表示[44]。DAGMM 是一种经典的基于重建的方法，它使用重建误差作为异常分数。
- **LSTM-VAE:** LSTM-VAE 用 LSTM 代替变分自编码器中的全连接网络，可以更好地捕获时间依赖性[68]。
- **MAD-GAN:** 使用 GAN 的多元异常检测策略利用 LSTM-RNN 作为时序异常检测模型的生成器和判别器[47]。
- **OmniAnomaly:** OmniAnomaly 采用随机循环神经网络进行时间序列异常检测，并利用重建概率来解释检测到的异常[46]。
- **USAD:** USAD 是一个基于自动编码器的框架，并以对抗方式进行训练[25]。USAD 中的自动编码器使对抗训练来提高鲁棒性。
- **GDN:** GDN 是一种基于预测的模型，通过图注意力网络在多变量时间序列中进行结构学习，并通过注意力权重解释检测到的异常[51]。

5.4.3 评价指标

我们使用精确率、召回率、F1 分数和 ROC 曲线下面积（AUC）作为评估指标。ROC 曲线表示真阳性率与假阳性率之间的曲线图，AUC 定义为 ROC 曲线下的面积。

表 5.2 不同模型在多模态数据集上的结果，粗体和下划线分别代表最优和次优的结果

	MSL			SMAP			SWaT			WADI		
Method	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
PCA	29.37	24.14	26.50	28.84	19.93	23.57	24.92	21.63	23.16	39.53	5.63	9.86
AE	71.66	50.08	58.96	72.16	79.95	75.86	72.63	52.63	61.03	34.35	34.35	34.35
DAGMM	49.11	55.62	52.16	58.45	90.58	71.05	27.46	69.52	39.37	54.44	26.99	36.09
LSTM-VAE	52.57	95.46	67.80	85.51	63.66	72.98	96.24	59.91	73.85	87.79	14.45	24.82
MAD-GAN	85.17	89.91	87.48	80.49	82.14	81.31	98.97	63.74	77.54	41.44	33.92	37.30
OmniAno.	88.67	91.17	89.90	74.16	97.76	84.34	98.25	64.97	78.22	99.47	12.98	22.96
USAD	93.08	89.17	<u>91.08</u>	90.96	85.29	88.03	98.51	66.18	79.17	64.51	32.20	42.96
GDN	91.35	86.12	88.66	89.32	88.72	<u>89.02</u>	99.35	68.12	<u>80.82</u>	97.50	40.19	<u>56.92</u>
MST-GAT	95.06	89.10	**91.98**	91.26	89.83	**90.54**	98.73	72.41	**83.55**	98.24	43.51	**60.31**

5.4.4 实验设置

我们使用 PyTorch 1.6.0 以及配备了 NVIDIA 2080ti GPU 的 Ubuntu 服务器来训练提出的模型。整个网络以 32 的批大小和总共 60 的 epoch 进行训练。所有数据集的嵌入维度 d 设置为 128。我们根据经验将滑动窗口大小设置为 32，时间卷积的内核大小设置为 16，并将每个数据集的注意力头数设置为 4。我们将 MSL、SMAP、SWaT 和 WADI 的 K 分别设置为 15、30、30 和 30。通过网格搜索选择模型超参数 γ_1 和 γ_2 分别为 0.5 和 0.8。我们利用 POT 算法[84] 来设置验证数据集的异常阈值。在推理阶段，任何异常分数超过阈值的时间戳都将被视为“异常”。

5.4.5 实验结果和分析

表 5.2总结了 MST-GAT 和对比模型在基准数据集上的性能比较。结果表明，就 F1 分数而言，MST-GAT 在四个基准上始终优于现有基线。我们可以观察到，大多数基线在 MSL 和 SMAP 数据集上表现更好，因为它们具有相对简单的异常模式和时空动态性，并且 MST-GAT 在 F1 分数方面仍然比最佳基线高出 0.9（%）和 1.52（%）分别在 MSL 和 SMAP 数据集上。此外，大多数基线在 SWaT 和 WADI 数据集上显示出较差的结果，但 MST-GAT 在 F1 分数（%）方面显著优于其他模型。多模态图注意力网络和时间卷积网络的结构有效地整合了多模态时间序列的信息，使 MST-GAT 能够捕获多模态时间序列中复杂的时空动态。与大

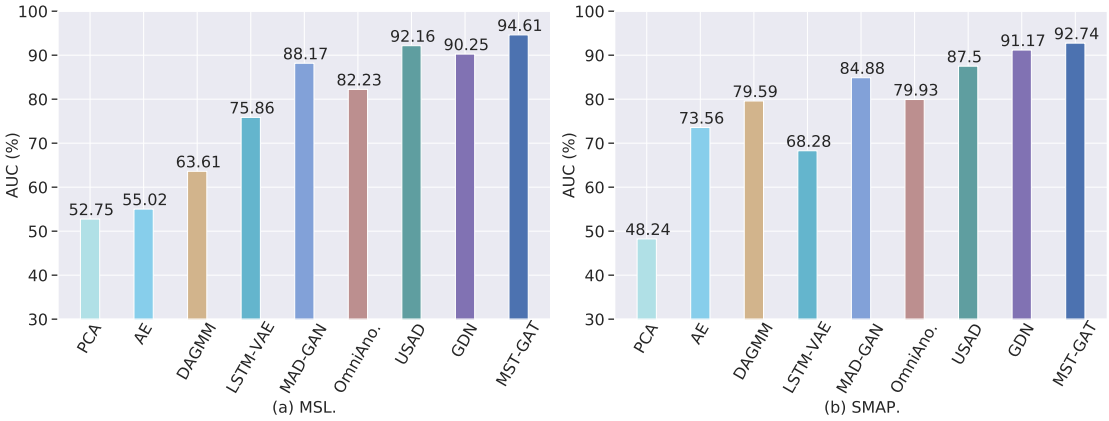


图 5.4 MSL 和 SMAP 数据集上的 AUC（%）结果

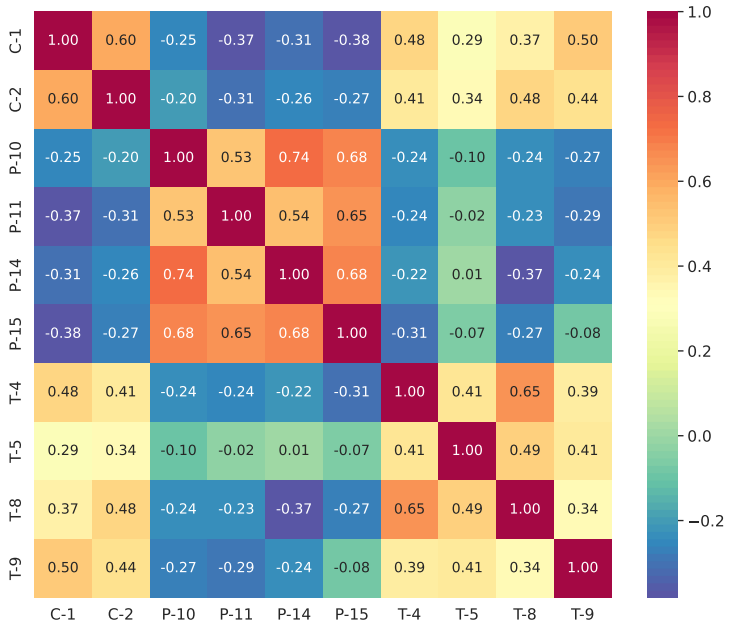


图 5.5 MSL 数据集中不同模态时间序列嵌入之间的余弦相似度，相同前缀属于同一模态

多数深度学习方法相比，传统方法（例如 PCA 和 DAGMM）表现不佳，因为它们难以在时间序列中编码全局信息，并且缺乏对空间和时间依赖性的充分考虑。此外，最近的 USAD 和 GDN 实现了比其他基线更好的性能。然而，GDN 并不擅长从时间序列中获取时间特征，USAD 忽略了多模态时间序列中的空间相关性。MST-GAT 优于同样采用图注意力网络的 GDN，显示了使用多模态图注意力网络和时间卷积网络的可行性。

我们进一步测试了 MSL 和 SMAP 数据集的 AUC 结果，如图 5.4 所示。提出的 MST-GAT 始终优于其他强大的基线。我们将性能优势归因于多模态时间序

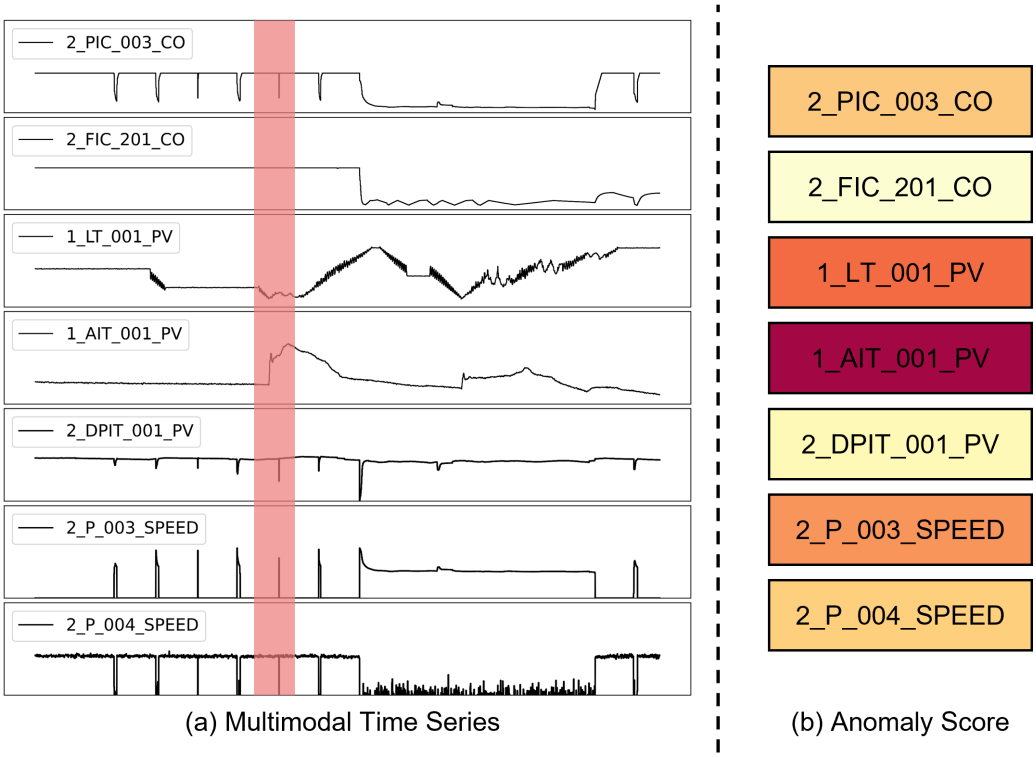


图 5.6 WADI 数据集上异常的可解释性分析，具有相同后缀的时间序列属于相同的模态

列中空间和时间信息的有效利用。通过使用 M-GAT 和时间卷积网络，MST-GAT 考虑模态内、模态间和时间信息的相互依赖关系，以捕获多模态时间序列之间的时空相关性。结果表明，多模态图注意力网络和时间卷积网络的使用有助于 MST-GAT 在异常检测中实现更高的真阴性率和更低的假阳性率。

MSL 数据集上时间序列嵌入的混淆矩阵如图 5.5 所示。可以看出，属于同一模态的单变量时间序列具有较高的相似性，这证明了 MSL 数据集上的模态内具有一致性。由于不同的模态可能表现出不同程度的相关性，模态 C 和模态 T 在 $[0.29, 0.50]$ 范围内的相似度具有较高的相关性，而模态 P 和模态 T 在 $[0.01, 0.37]$ 范围内的绝对相似度表现出较低的相关性。总体而言，时间序列嵌入强烈反映了模态内和模态间的相关性，揭示了使用 M-GAT 对不同时间序列之间的模态依赖性进行建模的可行性。

我们还进行了实验来展示 MST-GAT 的可解释性。图 5.6 (a) 展示了 WADI 数据集上多模态时间序列的异常示例，阴影区域是多模态时间序列中的异常区间。MST-GAT 给出异常区间内每个传感器的平均异常分数，红色阴影区域中平均异

表 5.3 SWaT 数据集上的超参数分析（F1 分数-%）. 最好的结果用粗体显示

γ_1 \ γ_2	0.4	0.6	0.8	1
0.2	82.07	82.84	83.26	82.84
0.5	82.03	83.16	**83.55**	83.45
0.8	81.97	82.50	83.35	83.01

表 5.4 MST-GAT 及其四种不同变体的性能比较，最好的结果以粗体突出显示

Method	Prec	Rec	F1
MST-GAT	**98.73**	**72.41**	**83.55**
- Modal	97.36	70.12	81.52
- Temp	96.73	69.54	80.91
- TopK	91.62	65.10	76.12
- Att	70.21	67.76	68.96

常分数最高的传感器被定位为最有可能导致该异常的传感器。如图 5.6（b）所示，颜色越深表示异常得分越高，选择 1_AIT_001_PV 作为最有可能导致该异常的传感器，因为它显示出与其他时间序列不一致的趋势。MST-GAT 提供符合人类直觉的可解释结果。MST-GAT 解释异常的能力很大程度上归功于 MST-GAT 的多模态图注意力，它正确地捕获了特征之间的相关性。

此外，我们对 SWaT 数据集的超参数进行敏感性分析。我们关注两个重要的超参数 γ_1 和 γ_2，它们用于提出的损失函数和异常分数。表 5.3 报告所有 γ_1 组合的 F1 分数（%），从 0.2 到 0.8，增量为 0.3，γ_2 从 0.4 到 1 ，增量为 0.2 。结果表明，MST-GAT 对 γ_1 和 γ_2 不敏感，对不同的超参数设置表现出鲁棒性。

5.4.6 消融实验

我们进行消融实验，这对于了解 MST-GAT 的每个组件的作用非常重要。我们逐渐移除不同的模块来观察性能的变化。首先，我们删除了 M-GAT 中的模态内和模态间注意力模块。其次，我们进一步去除了 MST-GAT 中的时间卷积。第三，为了研究图结构学习的必要性，我们用完全图代替了 TopK 实现的稀疏有向图。在一个完全图中，所有节点都相互连接。最后，我们丢弃了多头注意力模块中的注意力机制，并通过为每个邻居分配相等的权重来聚合信息。

MST-GAT 及其变体在 SWaT 数据集上的结果总结在表 5.4 中，我们发现：(***i***)

在实验中，移除模态内和模态间注意力模块会造成性能的下降，这说明在多模时序中显式捕获模态内和模态间依赖关系是有利于提高性能。我们推测多模态图注意力网络有利于获得更好的 MTS 异常检测特征表示。(***ii***) 配备时间卷积的 MST-GAT 优于没有时间卷积的模型，这表明在多模态时间序列中建模时间依赖性的必要性。(***iii***) 不使用注意力机制的 MST-GAT 变体比其他变体表现最差。由于每个单变量时间序列具有非常不同的属性，因此为每个邻居分配相同的权重会引入额外的噪声，并且无法对多模态时间序列中的复杂依赖关系进行建模。(***iv***) 我们可以观察到，移除每个组件都会不断降低性能，这证明了 MST-GAT 中每个组件的合理性。

5.5 本章小节

在本章中，我们设计了 MST-GAT，它是一种多模态时空图注意力网络，用于多模态时间序列异常检测。MST-GAT 利用多模态图注意力网络和时间卷积网络来捕获多模态时间序列之间的空间和时间相关性。MST-GAT 通过联合训练利用了重建和预测模块。此外，我们提出了一种基于重建概率和预测值的检测异常的有效异常解释方法。实验结果显示了 MST-GAT 优于最先进的基线，并且能够提供与人类直觉一致的可解释结果。

第六章　总结与展望

6.1　本文总结

本文研究了时间序列异常检测模型，针对概念漂移问题及检测音频时序中的对抗攻击异常分别提出了相应的单个模态数据上的异常检测模型在线稀疏 Transformer 和两个模态数据上的异常检测模型 MDFT。同时还对多个模态数据间的时空依赖性进行了初步探索，提出了 MST-GAT 模型。

首先，我们在单个模态的数据集上提出了一种基于概念漂移检测的时间序列异常检测框架，称为在线稀疏 Transformer。为了适应时间序列中的概念漂移，我们设计了 CDAM 模块来动态调整模型的学习率。CDAM 和在线学习共同促进在线稀疏 Transformer 及时更新模型参数来适应新的数据分布。此外，由于 Transformer 中 self-attention 的时间复杂度很高，我们设计了方根稀疏自注意力来替换标准自注意力，从而降低了时间复杂度，更有利于模型在真实场景中的部署。

其次，我们通过使用白盒和黑盒对抗攻击方法生成了带有时序对抗攻击异常的两个的数据集，即 WiAd 和 BlAd。我们设计了一个基于视觉 Transformer 的异常检测模型（MDFT）通过融合包含音频和文本的多模态数据来检测对抗攻击异常。此外，我们尝试了不同的实验设计来从多个方面测试模型效果。实验结果表明，MDFT 在两个数据集上都优于其对应的单模态模型。

最后，我们提出多模态时空图注意网络 MST-GAT，用于对齐数据上的多模态时间序列异常检测。MST-GAT 利用多模态图注意力网络和时间卷积网络来捕获多模态时间序列之间的空间相关性和时间依赖性。重建模块和预测模块被用来联合优化模型性能。在测试阶段，重建概率和预测值被用来解释被检测到的异常。在基准多模态数据集上的结果证明了 MST-GAT 的有效性，并且展示了 MST-GAT 对检测到的异常具有良好的可解释性。

6.2 未来工作

本文在不同类型的多模态时序数据上开展了异常检测的研究，包括非对齐的音频和文本数据以及对齐的多传感器时序数据。针对不同数据集的特点，本文设计了几个新颖的模型用于检测多模态时序数据中的异常。基于本文的工作，未来还可以从以下几个角度来研究多模态时序异常检测：

1) 深度学习模型凭借其强大的时序建模能力，在时间序列异常检测任务上已经得到了广泛的应用。但是，由于受到模型结构的限制，现有的时序模型无法被直接用于非对齐的多模态时间序列数据，例如自动驾驶汽车或无人机上采集的多模态传感器数据，这些数据往往是使用不同的采样率收集的。现有的方法主要在数据预处理阶段使用线性插值和下采样等方法来获得对齐的多模态数据，然后将其作为多维时序数据来输入给模型。但是，此种方法会带来一定的信息损失。如何有效地融合非对齐的多模态时序数据是未来值得研究的一个方向。
2) 在存储开销和推理速度方面，本文没有去重点考虑，模型参数量和推理速度仍然存在可以优化的空间。在实际部署中，多模态时序异常检测模型需要实时地处理来自多模态传感器的信号，并且及时地反馈检测结果。这就要求异常检测模型具有较低的处理延迟和较小的模型尺寸。因此，未来可以开展有关加快推理速度和降低模型存储开销方面的研究。
3) 近些年，自动驾驶领域发展迅速，自动驾驶汽车中包含大量相互关联的多模态数据，如雷达、图像、音频以及其他传感器的数据。充分融合这些多模态数据来及时发现自动驾驶过程中出现的异常情况，对于保障乘客的安全至关重要。因此，未来可以研究适用于自动驾驶系统的多模态时间序列异常检测模型。

参考文献

[1] Ahmed M, Mahmood A N, Hu J. A survey of network anomaly detection techniques[J]. Journal of Network and Computer Applications, 2016, 60: 19-31.

[2] Erhan L, Ndubuaku M, Di Mauro M, et al. Smart anomaly detection in sensor systems: A multi-perspective review[J]. Information Fusion, 2021, 67: 64-79.

[3] Ma X, Wu J, Xue S, et al. A comprehensive survey on graph anomaly detection with deep learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2021.

[4] Lu J, Liu A, Dong F, et al. Learning under concept drift: A review[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 31: 2346-2363.

[5] Gu M, Fei J, Sun S. Online anomaly detection with sparse Gaussian processes[J]. Neurocomputing, 2020, 403: 383-399.

[6] Saurav S, Malhotra P, TV V, et al. Online anomaly detection with concept drift adaptation using recurrent neural networks[C]//ACM India Joint International Conference on Data Science and Management of Data. 2018: 78-87.

[7] Cho K, Van Merriënboer B, Bahdanau D, et al. On the properties of neural machine translation: Encoder-decoder approaches[J]. arXiv preprint arXiv:1409.1259, 2014.

[8] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9: 1735-1780.

[9] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems. 2017: 5998-6008.

[10] Wang H, Wu Z, Liu Z, et al. Hat: Hardware-aware Transformers for efficient natural language processing[C]//Annual Meeting of the Association for Computational Linguistics. 2020: 7675-7688.

[11] Zhou H, Zhang S, Peng J, et al. Informer: Beyond efficient Transformer for long sequence time-series forecasting[C]//AAAI Conference on Artificial Intelligence.

2021: 11106-11115.

[12] Gulati A, Qin J, Chiu C C, et al. Conformer: Convolution-augmented Transformer for Speech Recognition[C]//Proc. Interspeech. 2020: 5036-5040.

[13] Schneider S, Baevski A, Collobert R, et al. wav2vec: Unsupervised pre-training for speech recognition[J]. arXiv preprint arXiv:1904.05862, 2019.

[14] Wu L, Zong D, Sun S, et al. A sequential contrastive learning framework for robust dysarthric speech recognition[C]//IEEE International Conference on Acoustics, Speech and Signal Processing. 2021: 7303-7307.

[15] Zhang Q, Lu H, Sak H, et al. Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss[C]//IEEE International Conference on Acoustics, Speech and Signal Processing. 2020: 7829-7833.

[16] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[17] Gain B, Haque R, Ekbal A. Not all contexts are important: The impact of effective context in conversational neural machine translation[C]//International Joint Conference on Neural Networks. 2021: 1-8.

[18] Sucholutsky I, Schonlau M. Soft-label dataset distillation and text dataset distillation[C]//International Joint Conference on Neural Networks. 2021: 1-8.

[19] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks [J]. arXiv preprint arXiv:1312.6199, 2013.

[20] Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples[J]. arXiv preprint arXiv:1412.6572, 2014.

[21] Khalid F, Ali H, Hanif M A, et al. Fadec: A fast decision-based attack for adversarial machine learning[C]//International Joint Conference on Neural Networks. 2020: 1-8.

[22] Samizade S, Tan Z H, Shen C, et al. Adversarial example detection by classification for deep speech recognition[C]//IEEE International Conference on Acoustics, Speech and Signal Processing. 2020: 3102-3106.

[23] Wang W, Park Y, Lee T, et al. Utilizing multimodal feature consistency to detect adversarial examples on clinical summaries[C]//Proceedings of the Clinical Natural Language Processing Workshop. 2020: 259-268.

[24] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[C]//ICLR. 2020.

[25] Audibert J, Michiardi P, Guyard F, et al. USAD: Unsupervised anomaly detection on multivariate time series[C]//Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 3395-3404.

[26] Ren H, Xu B, Wang Y, et al. Time-series anomaly detection service at microsoft [C]//ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 3009-3017.

[27] Li Z, Zhao Y, Han J, et al. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding[C]//Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021: 3220-3230.

[28] Kromanis R, Kripakaran P. Support vector regression for anomaly detection from measurement histories[J]. Advanced Engineering Informatics, 2013, 27: 486-495.

[29] Zhang G P. Time series forecasting using a hybrid ARIMA and neural network model[J]. Neurocomputing, 2003, 50: 159-175.

[30] Shipmon D T, Gurevitch J M, Piselli P M, et al. Time series anomaly detection; detection of anomalous drops with limited features and sparse examples in noisy highly periodic data[J]. arXiv preprint arXiv:1708.03665, 2017.

[31] Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32: 4-24.

[32] Zhao H, Wang Y, Duan J, et al. Multivariate time-series anomaly detection via graph attention network[C]//IEEE International Conference on Data Mining. 2020: 841-850.

[33] Chung J, Gulcehre C, Cho K, et al. Empirical evaluation of gated recurrent neural

networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.

[34] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.

[35] Yamanishi K, Takeuchi J I, Williams G, et al. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms[J]. Data Mining and Knowledge Discovery, 2004, 8: 275-300.

[36] Zhang C, Song D, Chen Y, et al. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data[C]//AAAI Conference on Artificial Intelligence. 2019: 1409-1416.

[37] Li J, Di S, Shen Y, et al. Fluxev: a fast and effective unsupervised framework for time-series anomaly detection[C]//Proceedings of the 14th ACM International Conference on Web Search and Data Mining. 2021: 824-832.

[38] Yu M, Sun S. Policy-based reinforcement learning for time series anomaly detection[J]. Engineering Applications of Artificial Intelligence, 2020, 95: 103919.

[39] Kiss I, Genge B, Haller P, et al. Data clustering-based anomaly detection in industrial control systems[C]//IEEE International Conference on Intelligent Computer Communication and Processing. 2014: 275-281.

[40] Chaovalitwongse W A, Fan Y J, Sachdeo R C. On the time series k-nearest neighbor classification of abnormal brain activity[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2007, 37: 1005-1016.

[41] Ma J, Perkins S. Time-series novelty detection using one-class support vector machines[C]//Proceedings of the International Joint Conference on Neural Networks. 2003: 1741-1745.

[42] Puggini L, McLoone S. An enhanced variable selection and isolation forest based methodology for anomaly detection with OES data[J]. Engineering Applications of Artificial Intelligence, 2018, 67: 126-135.

[43] Borghesi A, Bartolini A, Lombardi M, et al. A semisupervised autoencoder-based approach for anomaly detection in high performance computing systems[J]. Engi-

neering Applications of Artificial Intelligence, 2019, 85: 634-644.

[44] 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: 1-14.

[45] Shen L, Yu Z, Ma Q, et al. Time series anomaly detection with multiresolution ensemble decoding[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021: 9567-9575.

[46] Su Y, Zhao Y, Niu C, et al. Robust anomaly detection for multivariate time series through stochastic recurrent neural network[C]//ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 2828-2837.

[47] Li D, Chen D, Jin B, et al. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks[C]//International Conference on Artificial Neural Networks. 2019: 703-716.

[48] Cheng M, Xu Q, Lv J, et al. MS-LSTM: A multi-scale LSTM model for bgp anomaly detection[C]//International Conference on Network Protocols. 2016: 1-6.

[49] 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.

[50] Hundman K, Constantinou V, Laporte C, et al. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding[C]//ACM SIGKDD international conference on knowledge discovery & data mining. 2018: 387-395.

[51] Deng A, Hooi B. Graph neural network-based anomaly detection in multivariate time series[C]//AAAI Conference on Artificial Intelligence. 2021: 4027-4035.

[52] Poria S, Cambria E, Bajpai R, et al. A review of affective computing: From unimodal analysis to multimodal fusion[J]. Information Fusion, 2017, 37: 98-125.

[53] Liu X, Zhao J, Sun S, et al. Variational multimodal machine translation with underlying semantic alignment[J]. Information Fusion, 2021, 69: 73-80.

[54] Sebe N, Cohen I, Garg A, et al. Machine learning in computer vision: volume 29 [M]. 2005.

[55] Owens A, Wu J, McDermott J H, et al. Ambient sound provides supervision for visual learning[C]//European Conference on Computer Vision. 2016: 801-816.

[56] Poria S, Cambria E, Gelbukh A. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis[C]//Proceedings of the conference on Empirical Methods in Natural Language Processing. 2015: 2539-2544.

[57] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786): 504-507.

[58] Wang D, Cui P, Ou M, et al. Learning compact hash codes for multimodal representations using orthogonal deep structure[J]. IEEE Transactions on Multimedia, 2015, 17: 1404-1416.

[59] Wu D, Pigou L, Kindermans P J, et al. Deep dynamic neural networks for multimodal gesture segmentation and recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38: 1583-1597.

[60] SravyaPranati B, Suma D, ManjuLatha C, et al. Large-scale video classification with convolutional neural networks[C]//International Conference on Information and Communication Technology for Intelligent Systems. 2020: 689-695.

[61] Wen H, Liu Y, Rekik I, et al. Multi-modal multiple kernel learning for accurate identification of tourette syndrome children[J]. Pattern Recognition, 2017, 63: 601-611.

[62] Zhen Y, Yeung D Y. A probabilistic model for multimodal hash function learning [C]//Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2012: 940-948.

[63] Wang Y, Huang W, Sun F, et al. Deep multimodal fusion by channel exchanging [C]//Advances in Neural Information Processing Systems. 2020: 4835-4845.

[64] Iwana B K, Uchida S. Time series classification using local distance-based features

in multi-modal fusion networks[J]. Pattern Recognition, 2020, 97: 107024.

[65] Yang P, Chen B, Zhang P, et al. Visual agreement regularized training for multi-modal machine translation[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020: 9418-9425.

[66] Nedelkoski S, Cardoso J, Kao O. Anomaly detection from system tracing data using multimodal deep learning[C]//IEEE International Conference on Cloud Computing. 2019: 179-186.

[67] Park D, Erickson Z, Bhattacharjee T, et al. Multimodal execution monitoring for anomaly detection during robot manipulation[C]//IEEE International Conference on Robotics and Automation. 2016: 407-414.

[68] 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: 1544-1551.

[69] Yuan W, He K, Guan D, et al. Graph kernel based link prediction for signed social networks[J]. Information Fusion, 2019, 46: 1-10.

[70] Wang S H, Govindaraj V V, Górriz J M, et al. Covid-19 classification by FGC-Net with deep feature fusion from graph convolutional network and convolutional neural network[J]. Information Fusion, 2021, 67: 208-229.

[71] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[72] Duvenaud D, Maclaurin D, Aguilera-Iparraguirre J, et al. Convolutional networks on graphs for learning molecular fingerprints[J]. Advances in Neural Information Processing Systems, 2015: 2224-2232.

[73] Monti F, Boscaini D, Masci J, et al. Geometric deep learning on graphs and manifolds using mixture model CNNs[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 5115-5124.

[74] Zhang X, Huang C, Xu Y, et al. Spatial-temporal convolutional graph attention networks for citywide traffic flow forecasting[C]//Proceedings of the ACM Interna-

tional Conference on Information & Knowledge Management. 2020: 1853-1862.

[75] Cirstea R G, Guo C, Yang B. Graph attention recurrent neural networks for correlated time series forecasting–full version[J]. arXiv preprint arXiv:2103.10760, 2021.

[76] Zhu L, Wan B, Li C, et al. Dyadic relational graph convolutional networks for skeleton-based human interaction recognition[J]. Pattern Recognition, 2021, 115: 107920.

[77] Gama J, Medas P, Castillo G, et al. Learning with drift detection[C]//Brazilian Symposium on Artificial Intelligence. 2004: 286-295.

[78] Li S, Jin X, Xuan Y, et al. Enhancing the locality and breaking the memory bottleneck of Transformer on time series forecasting[C]//Advances in Neural Information Processing Systems. 2019: 5244-5254.

[79] Laptev N, Amizadeh A, Billawala Y. Yahoo labs news: Announcing a benchmark dataset for time series anomaly detection[Z]. 2015.

[80] Lavin A, Ahmad S. Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark[C]//International Conference on Machine Learning and Applications. 2015: 38-44.

[81] Stanway A. Etsy skyline[C/OL]//Online Code Repos. 2013. https://github.com/etsy/skyline.

[82] Ahmad S, Lavin A, Purdy S, et al. Unsupervised real-time anomaly detection for streaming data[J]. Neurocomputing, 2017, 262: 134-147.

[83] Kingma D P, Ba J. Adam: A method for stochastic optimization[C]//International Conference on Learning Representations. 2015: 1-15.

[84] Siffer A, Fouque P A, Termier A, et al. Anomaly detection in streams with extreme value theory[C]//Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017: 1067-1075.

[85] Alzantot M, Balaji B, Srivastava M. Did you hear that? adversarial examples against automatic speech recognition[J]. arXiv preprint arXiv:1801.00554, 2018.

[86] Carlini N, Wagner D. Audio adversarial examples: Targeted attacks on speech-to-text[C]//IEEE Security and Privacy Workshops. 2018: 1-7.

[87] Battenberg E, Chen J, Child R, et al. Exploring neural transducers for end-to-end speech recognition[J]. IEEE Automatic Speech Recognition and Understanding Workshop, 2017: 206-213.

[88] Mozilla common voice dataset[EB/OL]. 2019[2019-04-05]. https://voice.mozilla.org/en.

[89] Peters M, Neumann M, Iyyer M, et al. Deep contextualized word representations. arxiv 2018[J]. arXiv preprint arXiv:1802.05365, 1802, 12.

[90] Berg A, O'Connor M, Cruz M T. Keyword transformer: A self-attention model for keyword spotting[J]. arXiv preprint arXiv:2104.00769, 2021.

[91] Goh J, Adepu S, Junejo K N, et al. A dataset to support research in the design of secure water treatment systems[C]//International Conference on Critical Information Infrastructures Security. 2016: 88-99.

[92] 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 International Workshop on Cyber-Physical Systems for Smart Water Networks. 2017: 25-28.

[93] Shyu M L, Chen S C, Sarinnapakorn K, et al. A novel anomaly detection scheme based on principal component classifier[C]//Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop. 2003: 172-179.

[94] Provotar O I, Linder Y M, Veres M M. Unsupervised anomaly detection in time series using lstm-based autoencoders[C]//IEEE International Conference on Advanced Trends in Information Theory. 2019: 513-517.

硕士期间发表的学术论文以及学术成果

学术论文和出版物

[1] **Chaoyue Ding**, Shiliang Sun, Jing Zhao. Multi-Task Transformer with Input Feature Reconstruction for Dysarthric Speech Recognition[C]// IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2021: 7318-7322. (CCF B, Accepted)

[2] **Chaoyue Ding**, Shiliang Sun, Jing Zhao. Multi-Modal Adversarial Example Detection with Transformer[C]// International Joint Conference on Neural Networks (IJCNN). 2022. (CCF C, Accepted)

致　　谢

时光飞逝，日月如梭，不知不觉我已在华东师范大学大学度过了三年的研究生时光。无论在学习还是成长方面，我都感到收获满满。在实验室平时的讨论和交流中，我从老师和同学身上学到了许多。

首先，我想感谢我的导师孙仕亮教授。孙仕亮老师严谨的治学作风和不断探索的精神令我印象深刻。他在我的科研道路上提供了充分的指导和大量建设性的意见。孙仕亮老师对于论文逻辑与论文细节有着独到且深刻的见解。论文投稿之前，孙仕亮老师会一句句地带我推敲论文的语句逻辑和文章框架，并针对性地指出论文中存在的逻辑和细节问题。我也要感谢赵静老师，她在我小论文的撰写和发表上都付出了很多心力。每当我请赵静老师帮忙修改论文时，她就算很忙也会尽快帮忙阅读我的论文，第一时间反馈给我修改建议，我非常感谢赵静老师给我的帮助和关心。

其次，感谢我在 PRML 实验室遇到的所有同学。感谢师兄师姐，以及实验室的同学们。感谢他们在科研和生活上对我的帮助，陪我度过宝贵的研究生时光。感谢宗道明、张楠、吴丽丹、刘啸等师兄师姐，在我论文写作和实验遇到困难的的时候给予了我及时且重要的帮助。

最后，我想感谢我的家人，感谢你们尊重我人生道路上一次次的选择并赋予我追求理想的勇气，是你们的不离不弃和默默守护，才能让我全身心地投入到学校的学习和生活中去。