北方工业大学

硕 士 学 位 论 文

基于时间序列的工业互联网在线异常检测技术研究

学　生　姓　名　黄雅晴

学　　号　2021312110130

学科(专业学位)　控制科学与工程

研　究　方　向　智能信息处理

导　　师　董 哲

2024 年 6 月 3 日

Research on Online Anomaly Detection Technology of Industrial Internet based on Time-Series

By

HuangYaqing

A Dissertation Submitted to

North China University of Technology

In partial fulfillment of the requirement

For the degree of

Master of Engineering

North China University of Technology

June, 2024

基于时间序列的工业互联网在线异常检测技术研究

摘　要

随着工业互联网的发展，数字化和网络化的工业设备和生产线越来越多，大量的传感器和设备产生的数据形成了复杂的时间序列。这些数据中蕴含着设备运行状态、生产效率等重要信息，同时也可能包含潜在的设备故障和生产异常情况。传统的异常检测方法主要基于统计学和机器学习技术，通常需要对数据进行复杂的预处理和特征提取，并且对于异常的定义和检测策略往往需要人工干预。随着深度学习技术的发展，基于深度神经网络的异常检测方法逐渐成为研究热点。深度神经网络可以通过自动学习和提取数据中的特征，提高异常检测的准确性和效率。此外，深度强化学习作为一种结合了深度学习和强化学习的新兴技术，可以通过智能体与环境的交互学习最优策略，进一步优化异常检测的性能。基于此，本文尝试使用深度强化学习算法来弥补传统时间序列异常检测领域的缺陷，取得的研究成果如下。

针对工业互联网中复杂时间序列数据的异常检测问题，本文首先提出了一种融合注意力机制的 SR-ACNN 算法，其次将该算法中的谱残差方法作为奖励函数的一部分引入到基于深度强化学习的时间序列异常检测算法中，最后基于该异常检测算法设计了一个异常监测服务系统。融合注意力机制的 SR-ACNN 算法将谱残差方法变换的结果作为 ACNN 网络的输入，最后通过 ACNN 网络模型判别异常。基于弱监督的深度强化学习算法利用有限的异常标记数据，对未标注数据进行探索进而发现新异常类别，其主要内容如下：首先，通过设计谱残差方法、变分编码器和外在环境相结合的奖励机制，帮助智能体学习更好的策略；其次，通过引入一种采样方法，有效解决了异常检测领域数据不平衡的问题；最后，将自注意力机制和长短记忆网络结合，更好地捕捉时间序列的有效特征，使得强化学习模型更加稳健。基于上述研究内容，采用前后端分离技术，设计并实现了一个基于工业互联网的电力能源检测与异常诊断平台。该系统目前可实现时序数据实时监控、异常检测与报警等功能。

本研究针对工业互联网的时间序列异常数据在线异常检测问题，设计了两种算法。SR-ACNN 算法使用融合注意力机制的 CNN 网络学习异常判别规则，提高了算法的高效性和灵活性。基于现实中获取大量的有标签数据耗时且成本高的

问题，本文提出了基于弱监督的深度强化学习算法，通过设计外部奖励和内部奖励相结合的奖励机制，提高了算法的自适应性和有效性。基于弱监督的深度强化学习算法，开发了一个可在工业互联网领域实现时间序列在线异常检测的系统。

关键词：时间序列异常检测，谱残差，ACNN 网络，深度强化学习，混合奖励函数，自动编码器，注意力机制

Research on Online Anomaly Detection Technology of Industrial Internet based on Time-Series

Abstract

With the development of the industrial Internet, more and more industrial equipments and production lines are digitized and networked, and the data generated by a large number of sensors and equipment forms a complex time series. These data contain important information such as equipment operating status and production efficiency, and may also contain potential equipment failures and production anomalies. Traditional anomaly detection methods are mainly based on statistics and machine learning technologies, which usually require complex data preprocessing and feature extraction, and the definition of anomalies and detection strategies often require manual intervention. With the development of deep learning technology, anomaly detection methods based on deep neural networks have gradually become a research hotspot. Deep neural networks can improve the accuracy and efficiency of anomaly detection by automatically learning and extracting features from data. In addition, as an emerging technology that combines deep learning and reinforcement learning, deep reinforcement learning can further optimize the performance of anomaly detection through the interaction between the agent and the environment. Based on this, this thesis tries to use deep reinforcement learning algorithm to make up for the defects in the traditional time series anomaly detection field, and the research results are as follows.

To solve the anomaly detection problem of complex time series data in the industrial Internet, this thesis first proposes an SR-ACNN algorithm integrating attention mechanism, and then introduces the spectral residual method in the algorithm as a part of the reward function into the time series anomaly detection algorithm based on deep reinforcement learning. Finally, an anomaly monitoring service system is designed based on the anomaly detection algorithm. The SR-ACNN algorithm with fused attention mechanism takes the result of spectral residual method transformation as the input of the convolution attention module, and finally identifies the anomaly through the convolution attention module. The weak supervised deep reinforcement

learning algorithm uses limited anomaly labeling data to explore unlabeled data and find new anomaly categories. The main contents are as follows: First, by designing a reward mechanism that combines spectral residual method, variational encoder and external environment, the agent can learn better strategies; Secondly, by introducing a sampling method, the problem of data imbalance in anomaly detection field is effectively solved. Finally, the combination of self-attention mechanism and long and short memory network can better capture the effective features of time series, making the reinforcement learning model more robust. Based on the above research content, a power energy detection and anomaly diagnosis platform based on industrial Internet is designed and implemented by using the front-end separation technology. At present, the system can realize real-time monitoring of time series data, anomaly detection and alarm.

In this thesis, two algorithms are designed for online anomaly detection of time series anomaly data in industrial Internet. The SR-ACNN algorithm uses the CNN network with integrated attention mechanism to learn the anomaly discrimination rules, which improves the efficiency and flexibility of the algorithm. Due to the time-consuming and high cost of acquiring a large amount of labeled data in reality, this thesis proposes a deep reinforcement learning algorithm based on weak supervision. By designing a reward mechanism that combines external and internal rewards, the adaptability and effectiveness of the algorithm are improved. Based on weakly supervised deep reinforcement learning algorithm, a time series online anomaly detection system is developed in the field of industrial Internet.

Key words: time-series anomaly detection, spectral residuals, convolutional attention modules, deep reinforcement learning, mixed reward functions, autoencoders, attention mechanisms

目 录

第一章　绪论

1.1 研究背景及意义

当今世界信息技术迅速发展，贯穿人们生活的方方面面。随之而来的是每天都会产生大量的数据，特别是时间序列数据。时间序列数据能够捕捉到时间维度上的动态变化与趋势，因此逐渐成为学术界与行业从业者关注的热点[1]。时间序列有着非常广泛的应用场景[2-4]，如网络服务运维、工业生产设备监测[5]、医学的心电图[6]和血压检测、金融中的股票和期货价格走势、环境监测等。这些应用场景里，时间序列的异常现象通常作为衡量系统运作状况的关键指标，它往往揭示了系统可能出现的问题。比如，网络服务的瘫痪、工业生产设备的故障、人体的心率不规则或血压异常以及其他系统的异常情况，都可以借助对时间序列异常的分析来检测和识别。有效的时间序列异常监测可以及时识别异常并发出警报，这为工作人员提供了宝贵的响应时间，减轻了潜在的损失。其不仅有效地降低了潜在风险与成本，而且能够传递至关重要的信息。

信息技术的飞速发展同样促进了互联网行业的蓬勃，现如今互联网技术已经逐渐渗透到人们生活的各个方面，通过网络连接的计算机、手机等通信设备极大地改善了其功能，并深刻改变了人们的生活方式。在这一趋势下，“工业互联网”概念应运而生，它拓宽了网络接入设备的范畴，为电子装备、钢铁、采矿、电力等传统工业制造业带来了前所未有的便捷，朝着“万物互联”的宏伟目标迈进。工业互联网的本质是工业和网络技术的结合，它的出现不仅推动了工业智能化、数字化、信息化，而且使行业间的联系更加紧密，资源更易共享。随着各种生产设备和大型机械大规模接入工业互联网，工业大数据的利用价值日益凸显。实时监控与分析工业设备产生的时间序列数据，对于工业设备安全稳定运行十分重要。在此背景下，异常检测成为实现工业智能化的重要任务，构成了工业大数据分析领域的核心研究内容。

异常检测的核心任务是识别数据中未预料的突兀变化或罕见现象。该技术在工业各领域备受青睐，成为数据挖掘领域的研究热点。精准而快速的异常检测能够及时提醒工作人员进行故障排查，有效减少经济损失，并维护企业的声誉与品牌形象。为此，众多大型企业都具备简单的异常检测条件，用以监控其业务流程、产品及服务的健康状况。异常检测系统检测到异常情况时，会立刻向相关工作人员发出预警通知，以便工作人员能迅速采取与事故相关的应对措施。迄今为止，

众多学者已经在时间序列异常检测领域进行了深入的研究，并提出了一系列具体的解决方案。然而，现有研究大多使用传统的统计模型和机器学习方法，这些方法通常存在检测精度不高、泛化能力有限等问题。

机器学习使得计算机能够通过分析历史行为和经验来模仿人类的思维方式，实现自我优化和性能提升。在传统机器学习方法中，特征工程是模型训练前的关键步骤，其核心目标在于构建一个特征提取器，能够高效提取易于分类的特征向量。特征提取器的构建过程较为繁琐，还需要操作人员拥有专业的技能和丰富经验。深度学习[7]的一个分支，因为它构建的深层神经网络能够有效解决特征提取的问题，所以它在处理高维数据方面性能较强。在深度学习框架下，无需关注神经网络内部的详细计算过程，只需输入原始数据，神经网络会通过多层隐藏层的抽象和变换自动完成分类或其它复杂任务。

而机器学习与心理学的融合，推动了强化学习领域的发展。强化学习是一种学习模式，它通过智能体与环境的互动来优化自身的行为策略。通过这种方式，智能体旨在掌握自主学习和决策的技能，以实现自我提升并更好地适应环境的变化。这一方法的核心思想在于，智能体通过不断的尝试与环境的互动，利用反馈信息来调整自己的行为，从而在实践中不断优化策略。AlphaGo 在围棋比赛中击败冠军李世石，展示了其在决策和适应复杂环境方面的优秀性能，强化学习也因此成为研究热点。强化学习作为机器学习的一个关键领域，通过试错和探索，在奖励信号的指导下进行学习。近年来，深度学习（Deep Learning，DL）技术因其深度神经网络在特征提取方面的优势，已在多个领域取得显著成效。这种技术的运用，对强化学习领域产生了积极影响，推动了深度强化学习的发展。相较于传统强化学习方法，深度强化学习在处理复杂和高维度数据时更具优势，提高了算法的适应性和准确性。

Mnih 等人[8]提出了深度强化学习（Deep Reinforcement Learning, DRL）算法，将深度学习的感知功能和强化学习的决策功能相结合，创造出了一种高效的智能体学习新模式。深度学习使得智能体能够通过神经网络处理和理解高维度的输入数据，从而在感知层面上超越传统的强化学习方法。

随着工业互联网的快速发展，工业生产过程中产生的海量时序数据为设备状态监测和生产效率优化提供了丰富的信息资源。然而，如何有效地从这些数据中识别出异常信号，以预防潜在的设备故障和生产事故，是当前工业领域面临的一大挑战。传统的异常检测方法依赖于复杂的特征工程和人工设定的阈值，这不仅效率低下，而且难以适应多变的工业环境。因此，研究一种能够自动学习、适应性强、准确率高的异常检测方法具有重要的理论和实际意义。

因此，本文首先提出 SR-ACNN 算法，其在 SR 变换的基础上采用融合注意

力机制的卷积神经网络学习异常规则的判定，在高效的基础上提高了算法的泛化性。随后将 SR 方法作为深度强化学习模型奖励机制的一部分，提出基于弱监督的深度强化学习异常检测算法，提高异常检测方法的自适应性和准确性。本文的研究不仅在理论上推动了深度学习和强化学习在异常检测领域的交叉融合，而且在实践中为工业互联网环境下的智能监控和故障预警提供了有效的技术支持。通过设计并实现基于弱监督的深度强化学习异常检测系统，对提高生产安全性、减少意外损失、促进智能制造具有一定的现实意义。

1.2 国内外研究现状

异常通常指的是数据集中偏离常规分布的离群点或新颖点。异常检测技术的主要目的是利用特定的方法识别出这些与大多数数据明显不同的偏差实例[9]。鉴于异常现象在众多研究领域和应用场景中的普遍存在，各类针对特定领域的异常检测方法应运而生。近年来，随着信息技术的高速发展，特别是大数据和人工智能的发展，工业系统与信息技术的融合日渐紧密。工业系统运维涉及多个关键环节，包括设备状态个性化评估、异常检测、故障诊断和故障预测。广泛使用的传感器能够定期监控系统状态指标，帮助揭示潜在异常。结合大数据和人工智能技术，可以更有效地分析这些传感器数据，支持工业系统的智能化和自动化运维。基于时间序列的工业互联网异常检测逐渐成为异常检测领域的热门话题，受到了越来越多的学者的广泛关注。根据这些异常检测方法的不同特性，可以将其分类为多种类型，以便于深入研究和应用。总体而言，这些方法大致可分为两大类：一类是基于统计学的检测方法，另一类则基于机器学习原理。而基于机器学的方法又分为有监督、半监督、无监督和强化学习。

对时间序列数据的异常检测，基于统计学的方法可划分为两大子类别：一、依赖于数据分析的方法，二、依赖于预测模型的方法。数据分析类方法常采用 3sigma 准则[10]、Z-score、假设检验、傅里叶变换（Fast Fourier Transform，FFT）、箱线图、小波分析等方法来识别异常，但这些方法在模式异常和集合异常的检测上效果有限。而基于预测的方法包括滑动窗口[11]、滑动平均（Moving Average）[12]、奇异值分解（Singular Value Decomposition，SVD）、移动平均自回归（Autoregressive Integrated Moving Average model，ARIMA）[13-14]等方法。例如，Yu 等[15]研究人员采用了优化过的 ARIMA 模型来对真实世界的交通流量进行预测和分析，并通过比较预测结果与实际观测值的残差差异来辨识出异常情况。然而，此类方法主要适用于捕捉数据之间的线性关系，对于处理更为复杂的非线性

数据类型，其能力则显得相对不足。对于具体的工业应用而言，若能发现目标变量的分布模式，使用基于统计的方法是一种简单的发现异常的方式，可以快速发现异常点，适合工业系统数据分析对实时性的要求，但简单的基于统计的方法很难迁移到其他应用，另外这种方法不容易捕获复杂的时间依赖关系。尽管如此，在设计工业系统时间序列异常检测算法时仍可以借鉴基于统计的思路，以改善异常检测的实时性。

近年来，机器学习在异常检测领域大放异彩。大多数基于机器学习的方法首先从数据中提取特征，然后构建分类模型以辨识异常。在时间序列异常检测领域，机器学习方法可分为监督学习、半监督学习、无监督学习和强化学习。

（1）监督学习算法

在有监督学习中，模型的训练需要有完全标记的数据集。有监督学习方法包括随机森林、支持向量机（Support Vector Machine，SVM）[16]、贝叶斯网络[17]、基于规则和神经网络等方法[18]。例如，Malhotra 等人[19]创新性地提出了一个融合了循环神经网络（Recurrent Neural Network，RNN）和长短记忆网络（Long Short Term Memory，LSTM）的异常检测的算法，该算法专门用于处理时间序列数据。其通过学习正常数据集来构建预测模型，再利用模型生成的预测值与实际观测值之间的差异来判别数据点是否正常。这种方法能够有效地提取时间序列数据的动态特征，从而提高检测的准确性和稳定性。有监督学习方法在时间序列异常监测中具有较高的准确率，但在实际应用中，标记数据往往较为稀缺。

（2）半监督学习算法

半监督学习是有监督学习和无监督学习的结合，它在训练过程中使用的是有标签和无标签数据混合的训练集。半监督方法既减少了标记数据的需求，又利用了无监督数据中的潜在信息。例如，2023 年 Chen 等[20]提出了一种基于 VAE（Variational Autoencoders，VAE）的半监督异常检测模型，将 LSTM 网络引入到 VAE 编码器和解码器中，有效提取多元时间序列的依赖性特征，大大缩短了模型训练时间。半监督学习在数据标注成本较高的情况下，需要更少的人力资源，所以在实际应用中备受青睐。

（3）无监督学习算法

无监督学习是一种在没有标签数据的前提下，对数据进行聚类和特征提取。这类方法主要针对未标记的数据，其目标是发现数据中的潜在结构和分布规律。无监督学习方法不需要标签数据，因此在数据量巨大而标记数据有限的情况下具有较高的应用价值。常用的无监督机器学习的异常检测方法包括基于距离、基于密度、基于聚类、基于树等四种方法。基于距离的方法通过计算数据点与周围数据的距离来识别异常值。如果一个数据点的距离大于其它数据点，它是异常值的

概率就会大大增加。这类方法中常用的算法有局部异常因子和 k 最近邻。基于密度的方法是通过分析数据点的密度来完成异常值的检测。如果某个数据点周围的密度与其他区域明显相差较大，那么它可能是一个异常值。高斯混合模型和局部密度估计都是这种方法的实现。基于聚类的方法通过将数据点分为若干个簇，然后分析这些簇的特性来识别异常值。如果某个数据点不适合任何一个簇，或者它属于一个异常的簇，那么它可能是一个异常值。聚类算法如 k-均值层次聚类、DBSCAN、基于密度的聚类和高斯混合模型。基于树的方法使用决策树来构建模型，并基于模型对数据点的预测概率来衡量其是否能够被认作异常值。如果一个数据点的预测概率与其它数据点相比明显较低，那么其大概率是一个异常值。随机森林（Random Forest）和孤立森林（Isolation Forest）是这类方法的典型代表。另外，Li 等人[21]提出了一种基于生成对抗网络（Generative Adversarial Networks，GAN）的无监督多变量异常检测方法，创新性地使用 DR-score 的异常评分方法方法来充分利用 GAN 的生成器和鉴别器，通过重构和提取变量之间的潜在特征来对异常进行检测。Xu 等人[22]创造性地提出了一种基于无监督的变分自编码器异常检测算法，命名为 Donut。该算法在性能上显著超越了当时的最先进监督集成方法和基准 VAE 方法。Donut 是首个具有坚实理论基础的基于 VAE 的异常检测算法，为无监督异常检测领域带来了新的视角和方法。在时间序列异常检测中，无监督学习方法能够发现潜在的异常模式，但由于缺乏标签信息，其准确率可能较低。

（4）强化学习算法

强化学习是计算机学习生物环境互动策略的一种算法，通过不断探索和自身优化来寻找最佳行为模式。它从环境反馈中学习，通过不断调整策略以获得更高的长期利益。强化学习方法包括 Q-learning、深度 Q 网络（Deep Q Network，DQN）、策略梯度（Policy Gradient，PG)等。2020 年 Yu 等人[23]为解决时间序列异常检测领域的难点，提出了一个基于策略的强化学习架构。这一架构借助于 A3C 算法对时间序列异常检测其进行建模，其计算复杂度低，检测性能好。2021 年 Wu 等人[24]实现了一种半监督的时间序列异常检测算法，该算法将深度强化学习和主动学习相结合，无需手动调整参数，其 F1-score 比最佳的无监督学习方法高出 1.58 倍。2023 年 He 等人[25]提出了一种新颖的无监督学习强化算法，该算法使用核密度估计作为强化学习算法的奖励函数，在真实世界获取的人工注入异常的数据集上展现了 precision 为 90%，recall 为 80%，F1-score 为 85%的卓越性能。最近，Oh 和 Iyengar[26]将 IRL 应用于无监督环境中的顺序异常检测。强化学习的核心思想是通过奖励引导学习过程，通过与环境的交互来学习更好地识别异常的行为策略。强化学习可以在时间序列异常检测领域实现对异常数据的监测，

具有广泛应用前景。

1.3 主要研究内容

在本研究中，本文旨在提出一种应用于工业互联网的时间序列数据异常检测算法，并将其应用于实践中。主要研究内容如下：

（1）提出了一种融合注意力机制的 SR-ACNN 算法，该算法使用谱残差方法放大异常特征，并使用融合了注意力机制的卷积神经网络学习异常判别规则，提高了异常检测的效率。

（2）提出了一种基于弱监督的深度强化学习算法，将谱残差方法的输出作为奖励函数的一部分。该算法利用有限的异常标记数据，通过智能体与环境的交互，自主学习并发现新的异常类别。为了应对异常检测中的样本不平衡问题，引入了一种新的采样方法，以提高模型对异常样本的识别能力。提出了一个具有外部奖励的奖励机制，同时结合 SR 方法和变分编码器作为内部奖励，引导智能体学习有效的异常检测策略。通过结合自注意力机制和长短记忆网络，使得该算法能够更有效地捕捉时间序列数据的关键特征。

（3）开发了一个基于弱监督的深度强化学习异常检测系统，该系统能够实现时序数据的实时监控和异常报警。

1.4 章节安排

本文的具体章节安排如下：

第一章：绪论。详细阐述了时间序列异常检测的研究背景、重要性。介绍时间序列异常检测算法的分类和国内外的时间序列异常检测技术研究现状。随后，对本文主要研究内容进行了概略阐述。

第二章：融合注意力机制的 SR-ACNN 异常检测算法。首先对谱残差方法的基本原理进行了介绍，随后使用 ACNN 网络模型替代单一阈值检测异常点的传统方法，提高了算法的灵活性，最后给出实验和分析结果。

第三章：基于弱监督的深度强化学习时间序列异常检测算法。首先阐述了研究问题，接着介绍了数据预处理方式，随后详细描述了弱监督深度强化学习模型各部分及算法改进，最后通过在多种数据集上做实验证明了该算法的有效性。

第四章：基于上述研究内容，采用前后端分离技术，设计并实现了一个基于弱监督的深度强化学习异常检测系统。该系统目前可实现时序数据实时监控、异

常检测与报警等功能。

第五章：结论与展望。在本章中，对本文所开展的主要研究结论进行了归纳，并对所提出的异常检测算法以及系统中存在的问题进行了深入分析。同时，提出了未来的改进设想和系统演进的方向，以指导后续的研究和开发工作。

第二章　融合注意力机制的 SR-ACNN 异常检测算法

2.1 研究背景

异常检测的目的在于识别数据中的异常或不符合预期模式的数据点。该技术在众多工业应用中广受欢迎，是数据挖掘领域的一项关键研究。准确的异常检测能够及时发现生产过程中的异常情况，从而减少设备故障、提高生产效率和产品质量。一旦系统检测到异常现象，便会向相关工作人员发出警报，以便他们能够迅速做出更有效的决策。然而，在设计面向工业互联网的时序数据异常检测时，面临着多重挑战：

（1）算法泛化能力不够强。时间序列异常检测算法需要监控来自不同业务场景的多样化时间序列数据，这些数据通常表现出季节性、平稳性或非平稳性等典型的时间序列模式。因此，工业异常检测算法必须能够在这些不同的模式下都发挥良好的性能。然而，当前异常检测方法通用性不足，因此亟需寻找更具普适性的解决方案。

（2）效率较低。在业务应用中，监控系统必须能够在近乎实时的条件下处理数以百万计甚至数十亿计的时间序列。特别是对于分钟级的时间序列，必须在有限的时间内完成异常检测流程。因此，高效性是在线异常检测服务的核心要求之一。尽管计算复杂度高的模型可能具有更高的精确度，但它们在实时应用场景中的实用性有限。

为了解决上述问题，本章旨在开发一种准确、高效且泛化能力较强的异常检测方法。尽管传统的统计模型易于获取且应用广泛，但它们的准确性往往无法满足工业应用的要求。此外，还存在一些无监督方法，如 Luminoll[27]和 DONUT，但这些方法存在耗时过多或对参数调整过于敏感的问题。时间序列中的异常点与视觉显著性检测中的“突出”物体在本质上存在相似性，而谱残差方法在视觉显著性检测方面展现出了高效性和稳健性。因此，本章将视觉显著性检测领域中的谱残差（Spectral Residual，SR）方法引入到时间序列异常检测中。显著性描述的是图像或场景中那些能够迅速吸引人类注意力的显眼部分，引导人类聚焦于最重要的区域。类似地，当时间序列曲线出现异常时，异常总是视觉上最突出的部分。近期的显著性检测研究表明，当有足够的有标签数据时，卷积神经网络（Convolutional Neural Networks，CNN）可以进行有效的“端到端”训练。但是，CNN 在捕捉时间序列长期依赖关系方面性能较差，且常常难以捕捉时间序列数

据中的重要特征。为此，本章使用将 CNN 网络与注意力机制相结合的 ACNN 网络，帮助模型对时间序列数据中的重要特征进行加权，并更好地捕捉时间序列数据中的长期依赖关系，从而更好地识别异常点，提高模型的泛化能力。本章提出了一种名为 SR-ACNN 的异常检测算法，将 SR 方法的输出作为 ACNN 网络的输入，ACNN 在此过程中负责学习异常判别规则以取代 SR 模型原采用的单一阈值方法。

2.2 异常检测算法整体框架

给定一段长度为 T 的时间序列 $x \subseteq R^{T\times k}$：

$$x = \{x_1, x_2, ..., x_T\} \tag{2-1}$$

由于时间序列数据可以是单变量或多变量的，因此在单变量时间序列中 x_i 是一个标量，而在多变量时间序列中 x_i 是一个矢量，k 为时间序列的特征或维度数量。时间序列异常检测的任务就是输出一个对应的序列 $y = \{y_1, y_2, ..., y_T\}$。其中，$y_i \in \{0,1\}$ 表示 x_i 是否为异常点，如果 $y_i = 1$，则 x_i 为异常点；如果 $y_i = 0$，则 x_i 为正常点。

本章模型如图 2-1 所示，按顺序由以下模块组成，每个模块的详细机理将在本节以后依次叙述：

（1）数据预处理：分为滑动窗口取值和数据规范化两个步骤。原始时间序列数据首先进入数据预处理模块，随后被处理成谱残差模型可以接受的样本数据。

（2）谱残差模型：对经过数据预处理的原始时间序列进行 SR 变换，将 SR 变换后的结果作为 ACNN 网络模型的输入。

（3）ACNN 网络模型：在经过 SR 变换后的时间序列数据的基础上学习异常判别规则，检测异常点。

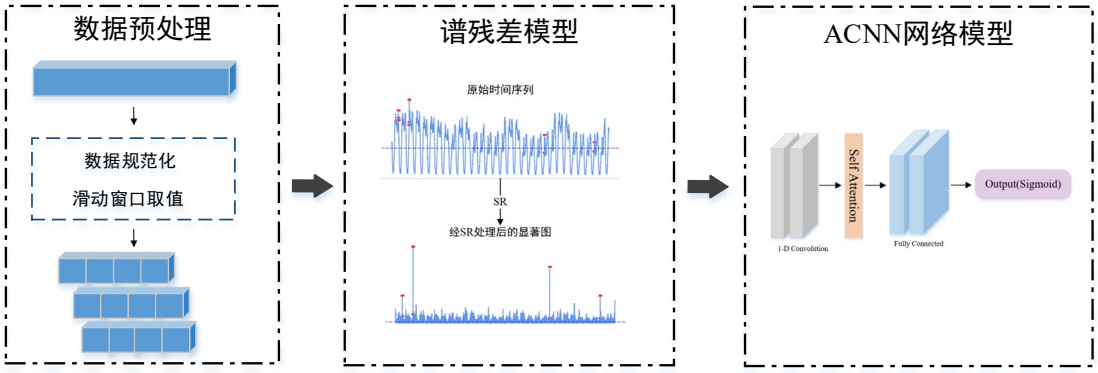

图 2-1　SR-ACNN 模型总体架构

2.3 数据预处理

（1）滑动窗口取值

在时间序列数据的预处理阶段，本文采用了滑动窗口的方法进行重采样，如图 2-2 所示。其具体策略是选取一个固定长度 m 的滑动窗口 $\{x_{i-m+1},x_{i-m+2},...,x_{i-1},x_i\}$，其中 $i=m,m+1,...,n-m+1$。在整个时间序列中以一定的步长从起始位置开始逐步向右移动。每次滑动窗口移动后，覆盖的数据区域构成一个小样本 s_i，这样就将原始的长时间序列划分为多个子序列 $\{s_m,s_{m+1},...,s_n\}$。这些子序列 s_i 可以被看作是新的数据集，用于模型的训练和分析。

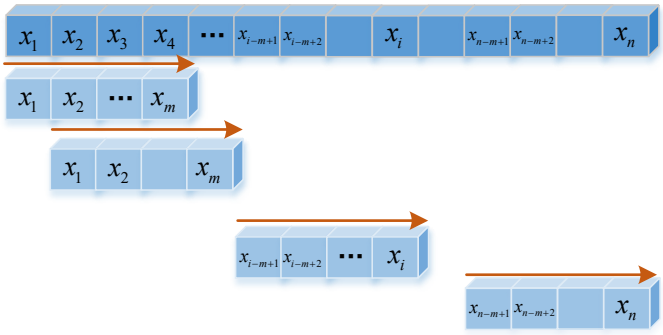

图 2-2　滑动窗口取值法示意图

（2）数据规范化方法

对于序列 $X=\{x_1,x_2,...,x_n\}$ 进行变换：

$$y_i=\frac{x_i-\min\limits_{1\le i\le n}(X)}{\max\limits_{1\le i\le n}(X)-\min\limits_{1\le i\le n}(X)} \tag{2-2}$$

则新序列 $Y=\{y_1,y_2,...,y_n\}\in[0,1]$ 且无量纲。该数据规范化方法又称 min-max 法。当 X 为矢量时，其 min 和 max 值为其各个空间的最小和最大值。

2.4 谱残差模型

受到计算视觉领域的启发，本章采用了一种基于快速傅里叶变换的谱残差方法[28]，该方法在视觉检测应用中表现出高效的性能。谱残差方法可以概括为以下四个关键步骤：

（1）对输入的序列数据进行傅立叶变换，得到其对数振幅谱；

（2）获取对数平均频域振幅谱；然后，计算出谱残差；

（3）通过傅立叶逆变换将频域的谱残差转换回时域。

数学上，给定一个序列数据 x，则有：

$$A(f) = Amplitude(\xi(x)) \tag{2-3}$$

$$P(f) = Phrase(\xi(x)) \tag{2-4}$$

$$L(f) = \log(A(f)) \tag{2-5}$$

$$AL(f) = h_q(f) \bullet L(f) \tag{2-6}$$

$$R(f) = L(f) - AL(f) \tag{2-7}$$

$$S(x) = \left\| \xi^{-1}(\exp(R(f) + iP(f))) \right\| \tag{2-8}$$

可以通过公式（2-3）到（2-8）来描述谱残差方法的全过程。其中，本章采用部分傅里叶变换[29]和部分傅立叶逆变换，分别用符号 ξ 和 ξ^{-1} 表示。x 是 $n \times 1$ 的输入时间序列，$Amplitude(f)$ 代表输入序列数据 x 的振幅谱，$P(f)$ 为相位谱，$L(f)$ 是 $A(f)$ 的对数形式，而 $AL(f)$ 是 $L(f)$ 的平均谱，可通过 $h_q(f)$ 卷积来逼近。其中，$h_q(f)$ 是一个 $q \times q$ 的矩阵，定义如下：

$$h_q = \frac{1}{q^2}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \tag{2-9}$$

$R(f)$ 即为谱残差，它通过对数谱 $L(f)$ 减去平均对数谱 $AL(f)$ 得到。谱残差实质上是输入序列在频域中的压缩表示，代表了序列中的异常部分。最终，通过傅立叶逆变换将处理后的序列重构到时域，并用 $S(x)$ 表示输出的显著映射后的序列。

图 2-3 展示了一个原始时间序列及其经过 SR 处理后的显著性图示例。从图中可以看出，显著性图中的异常点（红色部分）相较于原始输入中的异常点更为显著。

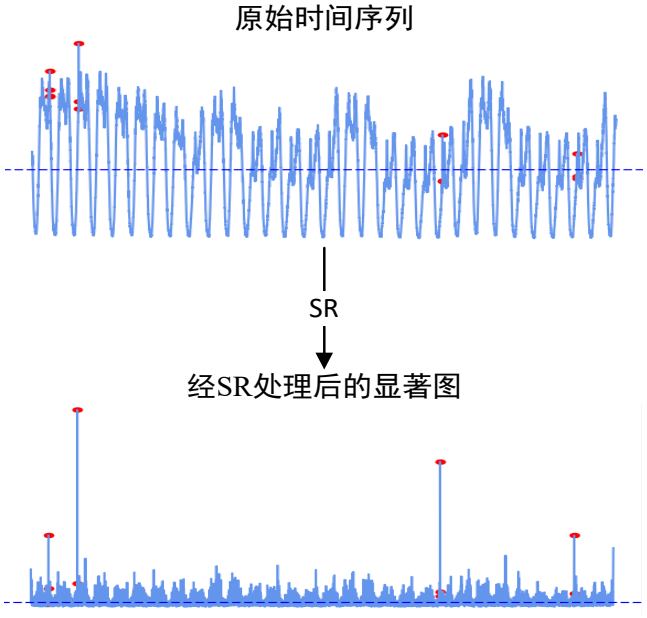

图 2-3　SR 变换后的显著性图示例

通过显著性映射，可以利用一个简单的规则来正确地识别异常点。可采用一个简单的简单阈值 τ 来标注异常点。给定显著性映射 $S(x)$，可计算输出序列 $O(x)$：

$$O(x_i)=\begin{cases}1, & \text{如果}\dfrac{S(x_i)-\overline{S(x_i)}}{S(x_i)}>\tau\\ 0, & \text{其他}\end{cases} \tag{2-10}$$

其中，x_i 表示序列 x 中的任意点，$\overline{S(x_i)}$ 是 $S(x_i)$ 之前 z 点的局部平均值。

在实践中，FFT 操作是在一个滑动窗口的序列内进行的。此外，期望该算法能以低延迟覆盖异常点。给定一段时间序列 $\{x_1,x_2,...,x_n\}$，其中 x_n 是最近的点，需要判断 x_n 是否是一个异常点。如果 x_n 位于滑动窗口的中心，那么 SR 方法效果更好。因此，在将序列输入到 SR 模型之前，在 x_n 之后加入几个估计点。估算点 x_{n+1} 的值计算公式如下：

$$\overline{g}=\frac{1}{m}\sum_{i=1}^{m}g(x_n,x_{n-i}) \tag{2-11}$$

$$x_{n+1}=x_{n-m+1}+\overline{g}\bullet m \tag{2-12}$$

其中，$g(x_i,x_j)$ 表示点 x_i 到 x_j 直线的梯度，g 表示上述点的平均梯度。m 是前面考虑的点数，在实验中设置 m 为 5。由于第一个估计点起着决定性的作用。因此，只需复制 x_{n+1} 进行 κ 次，并将这些点添加到序列的尾部。

2.5 ACNN 网络模型

2.5.1 注意力机制

注意力（Attention）机制最早是由谷歌的 Mnih 团队[30]提出的。自 2017 年以来，注意力机制[31]被广泛应用于神经网络领域，其本质思想如下图 2-4 所示。

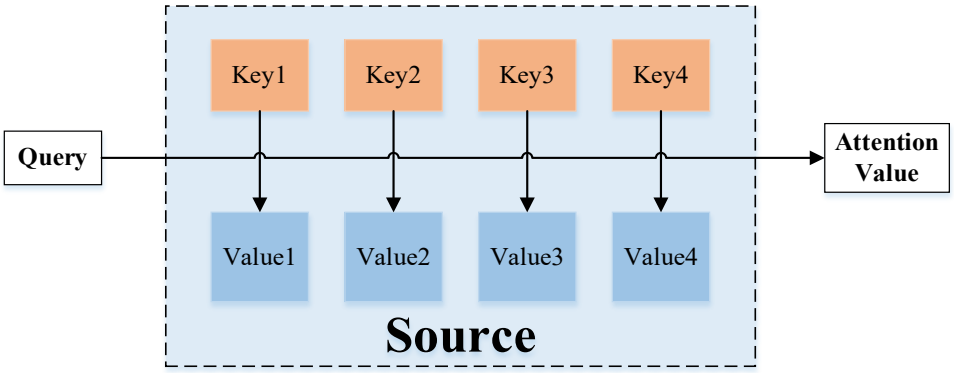

图 2-4　Attention 机制本质思想

人工神经网络中的注意力机制模仿了人类大脑的选择性视觉注意力。在人类

的视觉感知过程中，倾向于对整个场景中的特定区域给予更多的关注，这些区域被视为重点关注区域，而其他区域则可能被轻微关注或者完全忽视。因此，注意力机制在本质上是一种强调关注特定目标细节，同时忽略或抑制不相关信息的机制。

注意力机制的核心理念是将源信息 *Source* 视为一系列键值对 $\langle key, value \rangle$ 的集合。在处理目标信息 Target 中的特定元素（记为查询 *Query* ）时，注意力机制会评估查询与源信息中所有键之间的相关性或相似性，从而确定一个权重系数矩阵，这个矩阵体现了每个键 *key* 对于其对应值 *value* 的贡献程度。简而言之，注意力机制通过查询与键之间的相似度来获取权重系数，进而计算源信息 *Source* 中每个键值对中的值 *value* 的加权求和，最终得到注意力值的度量。可以用公式（2-13）来表示这一过程：

$$Attention(Query, Souce) = \sum_{i=1}^{L_x} Similarity(Query, key_i) * Value_i \tag{2-13}$$

关于 Attention 机制的具体运算过程，若对目前众多方法进行概括，可以将其精炼为三个主要步骤。

第一阶段：可以采用多种函数和机制来计算 *Query* 与某个 key_i 之间的相似性和相关性时。目前经常用到的方法有：计算两者向量的点积、衡量两者向量的余弦相似性，或者利用额外的神经网络来确定值。如下所示：

（1）点积的公式如下：

$$Similarity(Query, key_i) = Query \bullet Key_i \tag{2-14}$$

（2）余弦相似性的公式如下：

$$Similarity(Query, Key_i) = \frac{Query \bullet key_i}{\| Query \| \bullet \| key_i \|} \tag{2-15}$$

（3）MLP 网络的公式如下：

$$Similarity(Query, key_i) = MLP(Query, key_i) \tag{2-16}$$

第二阶段：在第一阶段，原始数值经过归一化处理后，具有可比性。随后，对得分采用类似于 SoftMax 的计算方法进行数值转换，将原始得分映射为一个概率分布，其权重和为 1。这种方法不仅实现了归一化，而且通过 SoftMax 函数的性质，自然地赋予了关键元素更高的权重。计算公式如下所示：

$$a_i = Softmax(Sim_i) = \frac{e^{Similarity_i}}{\sum_{j=1}^{L_x} e^{Similarity_j}} \tag{2-17}$$

第三阶段：根据第二阶段计算得来的权重系数对 *Vaule* 值进行加权求和：

$$Attention(Query, Source) = \sum_{i=1}^{L_x} a_i \bullet Value_i \tag{2-18}$$

由此可得 *Query* 所对应的 Attention 数值，这三阶段的计算流程如图 2-5 所示。

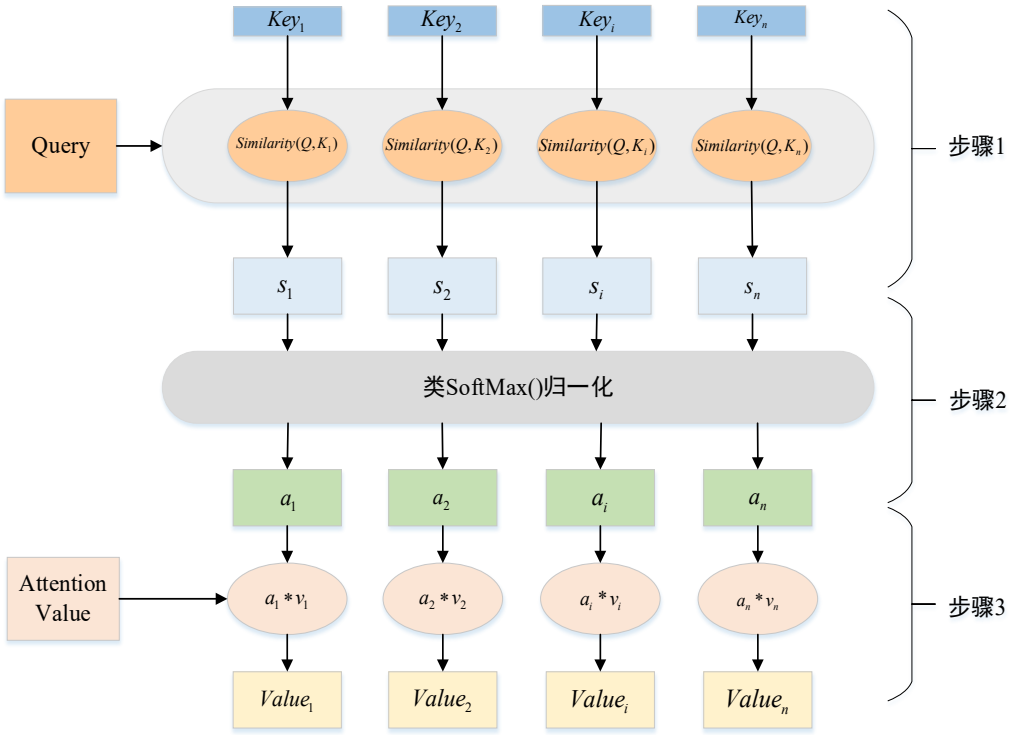

图 2-5　计算 Attention 值的三个阶段

2.5.2 卷积神经网络

卷积神经网络是一种在传统前馈神经网络基础上发展而来的深度学习模型，它通过引入卷积层和池化层，显著提升了在计算机视觉任务中的性能，如目标检测、图像识别和分类等。CNN 网络的具体结构如图 2-6 所示：

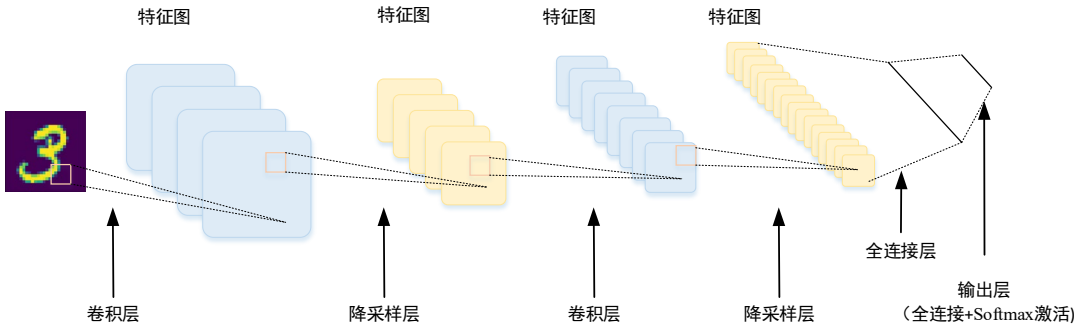

图 2-6　卷积神经网络结构图

CNN 的基本架构[32]包括输入层、多个卷积层、池化层、全连接层、输出层以及激活函数[33-35]。根据具体任务的需求，这些层次结构可以进行增减或调整。卷积层是 CNN 的核心组成部分，负责从输入数据中提取局部特征。一个卷积层可以包含多个卷积核，这些卷积核通过与输入数据进行卷积操作（即点积运算），从而生成特征图。每个卷积核能够捕捉到数据中的某种特定模式或特征，随着卷积核数量的增加，网络能够学习到更多样化的特征表示。为了控制模型的复杂度和参数数量，卷积层之后通常会接一个池化层。池化层的作用是对卷积层的输出进行降维处理，减少后续层的计算负担。常见的池化操作有最大池化和均值池化操作[36]。

最大池化是一种在卷积神经网络中实施的空间降维技术，它通过对卷积层输出的特征图进行局部区域内的极值提取，以实现数据压缩和特征筛选。如图 2-7 所示，最大池化操作会逐个检视特征图中定义好的池化窗口内的所有元素，并选择其中的最大值作为该窗口的代表。

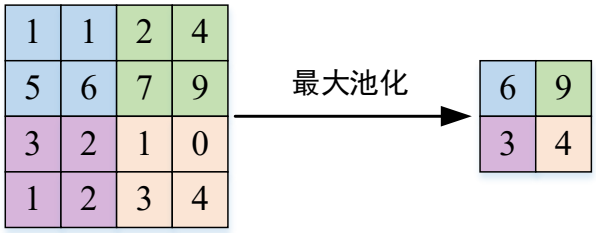

图 2-7　最大池化示意图

均值池化是一种深度学习中常用的池化策略，它通过计算池化窗口内所有数据元素的平均值来实现降维处理。这种方法的核心在于提取区域内的统计平均特征，从而在减少数据复杂度的同时，保留整体的背景信息。均值池化能够平滑局部特征，有助于消除噪声，并强化数据的泛化能力。如图 2-8 所示，在执行均值池化时，每个池化窗口覆盖特征图的一小块区域，然后计算该区域内所有像素点的平均值，以此作为该区域的代表。

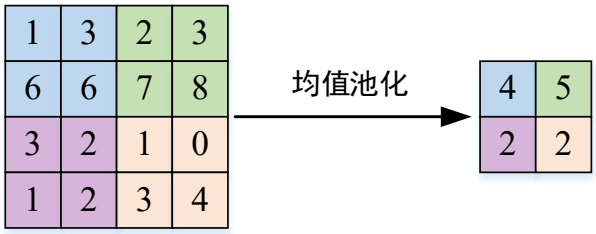

图 2-8　均值池化示意图

在经过多个卷积层和池化层的处理后，数据被送入全连接层。全连接层的每个神经元都与前一层的所有神经元相连接，这样设计的目的是为了整合局部特征，

形成全局的特征表示。全连接层之后通常会接一个 Softmax 层将网络的输出转换为概率分布，表示样本属于各个类别的置信度。其计算公式如下：

$$Softmax(z_i)=\frac{e^{z_i}}{\sum_{c=1}^{C}e^{z_C}} \tag{2-19}$$

其中 i 表示输出节点的编号，该函数常用于多分类问题，输出样本分别属于每个类别的置信度。

2.5.3 融合注意力机制的 ACNN 网络

原始的 SR 方法利用显著图上的单一阈值来检测异常点，如式（2-10）所示。但该规则过于原始，需要寻找更复杂的决策规则。将 ACNN 网络直接应用于 SR 方法的输出，ACNN 网络模型负责学习异常判别规则来代替原来 SR 所采用的单一阈值解决方案。

本章选择 CNN 作为基础模型，CNN 可以通过卷积层有效地提取时间序列数据中的局部特征，如趋势、周期性和模式。如果异常主要表现为局部特征的突变，CNN 会更加有效。对于需要快速响应的实时系统，CNN 通常具有更快的处理速度，因此是更好的选择。CNN 更适合数据量较小或中等的情况，因为它们通常需要较少的参数并且训练速度较快。但 CNN 在处理时间序列数据时存在一些局限性，尤其是在捕捉长期依赖性和重要特征方面。

ACNN 网络模型可以学习时间序列数据中不同时间点的重要性，以便更好地捕捉异常的特征。ACNN 网络模型通常由两个主要部分组成：卷积层和注意力机制层。卷积层可以提取时间序列数据中的局部特征，例如趋势、周期性和模式等。注意力机制可以对卷积层的输出进行加权，以便突出时间序列数据中最重要的特征。这样，模型可以更加关注时间序列数据中与异常相关的特征，而忽略其他不重要的特征。最后，将 ACNN 网络模型的输出传递给一个全连接层，以进行异常检测。在不改变 CNN 优势的前提下，结合注意力机制可以进一步提高异常检测的效果。具体来说，可以使用注意力机制来增强 CNN 网络对时间序列数据中重要特征的关注。ACNN 网络模型如图 2-9 所示。

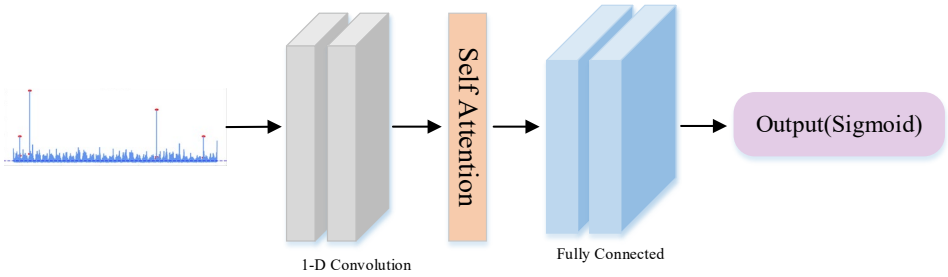

图 2-9　ACNN 网络模型

2.6 实验与分析

2.6.1 数据集

本章采用了以下三个工业数据集来评估 SR-ACNN 模型，其数据统计信息如表 3-1 所示。

表 3-1　数据集信息

	KPI	Yahoo	UIPD
总点数	3004066	572966	235860
异常点数	79554(2.65%)	3896(0.68%)	2122(0.9%)

KPI：由 AIOPS 数据竞赛发布，由多个 KPI 曲线和异常标签组成，这些异常标签来自不同的互联网公司，包括搜狗、腾讯、eBay 等，时间间隔大多为 1 分钟，部分为 5 分钟。该数据集集成了多个真实场景下互联网公司 KPI 时间序列的数据，并附有准确的实地标签。KPI 数据分为服务 KPI 和机器 KPI 两大类，分别代表了 Web 服务的性能指标和机器设备的运行状况。服务 KPI 涵盖了页面响应时间、页面浏览量、连接错误次数等指标，而机器 KPI 则包括 CPU 利用率、内存利用率、磁盘 IO、网卡吞吐量等。

Yahoo Benchmark：是 Yahoo 实验室发布的一个开放的异常检测数据集。部分时间序列曲线是合成的，而另一部分则来自雅虎服务的实际流量。合成数据集中的异常点由算法生成，真实曲线中的异常点为手工标记。

University Laboratory Power Dataset（UIPD）：电力数据的采集和分析在能源转型的当今世界显得尤为重要。其数据集来源为工业互联网中某高校实验室电力设备线上真实数据，如相电压数据、相电流数据、频率数据、有功功率数据和功率因素数据等，其采样间隔为 1 分钟。

2.6.2 评价指标与策略

（1）评价指标

由于异常检测问题本质上是一个样本不平衡的分类问题，因此使用分类准确率来衡量模型性能并不具有很大的实际意义。相反，通常使用精确率和召回率来评估异常检测的效果。在本章中，将异常序列数据定义为正样本，用 TP 、 FP 、 TN 和 FN 分别表示模型正确检测为正样本的数量、错误检测为正样本的数量、正确检测为负样本的数量以及错误检测为负样本的数量。基于这些指标，可以定义精确率、召回率和 F1 分数，具体计算方法如下：

在异常检测领域中，精确率（precision）这一指标可以被理解为，在所有被算法识别为异常的样本中，实际上确实是异常的样本所占的比例。精度越高，表示误报越少，其公式如下所示：

$$precison = \frac{TP}{TP + FP} \tag{2-20}$$

召回率（recall）是正确检测到的异常点与所有实际异常点的个数比率。召回率越高表示遗漏检测越少，其公式如下所示：

$$recall = \frac{TP}{TP + FN} \tag{2-21}$$

F1 分数（F1-score）是通过计算精确率和召回率的调和平均数得到的。这使得 F1 分数在分类任务中评估模型方面更加全面，其公式如下所示：

$$F1 - score = \frac{2 \times precision \times recall}{precision + recall} \tag{2-22}$$

（2）评估策略

在实际应用中，相关工作人员关注的是整个时间序列的异常情况，而不是每个数据点的细节。因此，本章采用以下评价策略，即对于连续的异常点时间段，在允许延长范围内，只要模型能够在其中任意一个时间点触发警报，就可以认为该异常段被成功检测到了。图 2-10 是这种评价策略的示意图。其中第一行红色框内的点表示异常点，第二行是模型的预测结果，假设阈值 k 设为 2，蓝色框内的点被判断为异常点；黄色框内检测到的异常点延迟 3 个单位，超过阈值，所以整段被判断为正常点。最终调整后的异常检测结果在第三行显示，在此基础上可以计算相应评价指标。在实验中，根据实际应用的要求，对于分钟级别的时间序列设 k 为 7，小时级别的时间序列设 k 为 3，日级别的时间序列设 k 为 1。

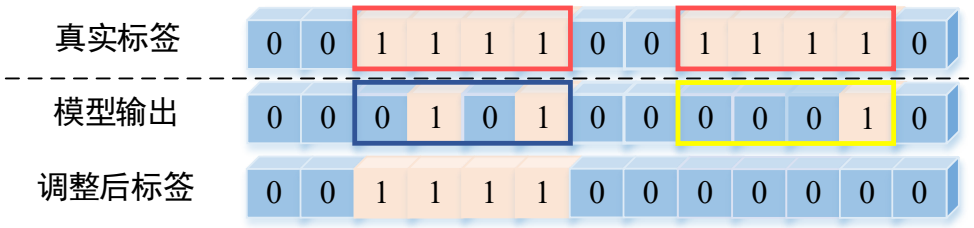

图 2-10　评估策略示意图

2.6.3 实验结果与分析

为证明 SR-ACNN 在时间序列异常检测任务中的性能，本章将其与以下几种时间序列异常检测方法进行比较。

Lumino：它是一个轻量级和可配置的 Python 库，配合多种统计异常检测算

法。它不需要任何训练，可以直接在测试数据上返回异常分数。

DSPOT[37]：它们被提出用于检测流单变量时间序列中的异常值，并能够自动选择阈值。需要少量数据进行初始化或校准。与传统的异常检测方法相比，DSPOT 具有自动选择阈值的能力，这意味着它能够在没有人工干预的情况下确定何时将数据点标记为异常。

DONUT：是一种基于分布的时间序列异常检测算法。它通过构建时间网络并利用时间序列数据的分布特性来检测异常。DONUT 算法的核心思想是将时间序列中的每个时间点视为一个节点，并根据数据点之间的相似性构建边，从而形成一个时间网络。

MAD-GAN：该方法是一种无监督的多变量异常检测方法，其基于生成性对抗网络，并在 GAN 的结构中嵌入了 LSTM-RNN 作为核心模型，利用其同时捕捉时间序列数据的时间相关性和分布特征。

VAE-LSTM：该方法一种结合了深度生成模型表示学习能力（采用变分自动编码器 VAE 实现）与长短期记忆网络（LSTM）时间建模能力的混合异常检测方法。该方法旨在通过融合两种模型的优势，从而提高对时序数据中异常模式的识别能力。

将数据集分为 80%的训练集和 20%的测试集，需要训练数据的模型在训练集上训练。通过过度采样异常来训练 SR-ACNN 模型，并保持样本数据正负比例为 1:2。实验中，$h_q(f)$ 中 q 值设置为 3，前面点的局部平均值个数 z 设为 21，估计 κ 的点数设为 5，滑动窗口大小 w 在 KPI 上设为 1440，在 Yahoo 上设为 64，在 UIPD 上设为 32。ACNN 网络的损失函数采用交叉熵损失函数，其训练过程中使用 SGD 优化器。本章算法与其他基准算法在不同数据集上的精确率、召回率和 F1 分数如表 3-2、表 3-3 和表 3-4 所示。

表 3-2　SR-ACNN 与其他方法在 KPI 数据集上的比较

数据集	模型	Precision	Recall	F1-score
KPI	Luminol	0.431	0.359	0.392
	DSPOT	0.624	0.439	0.515
	DONUT	0.368	0.394	0.380
	MAD-GAN	0.834	0.752	0.791
	LSTM-VAE	0.791	0.763	0.777
	SR-ACNN	0.892	0.749	0.814

表 3-3　SR-ACNN 与其他方法在 Yahoo 数据集上的比较

数据集	模型	Precision	Recall	F1-score
Yahoo	Luminol	0.316	0.763	0.447
	DSPOT	0.285	0.468	0.354
	DONUT	0.219	0.792	0.343
	MAD-GAN	0.861	0.754	0.804
	LSTM-VAE	0.839	0.842	0.840
	SR-ACNN	0.887	0.849	0.868

表 3-4　SR-ACNN 与其他方法在 UIPD 数据集上的比较

数据集	模型	precision	recall	F1-score
UIPD	Luminol	0.713	0.336	0.457
	DSPOT	0.397	0.531	0.454
	DONUT	0.421	0.387	0.403
	MAD-GAN	0.893	0.841	0.866
	LSTM-VAE	0.884	0.797	0.838
	SR-ACNN	0.915	0.826	0.868

实验表明，本章提出的 SR-ACNN 方法在三个数据集上的性能表现均接近于或优于其他最佳基准方法，泛化性能最优，并且在准确率方面相对其他最优基准方法平均提高 4.1%，其平均准确率达到 85.0%。

2.7 本章小结

本章提出了一种融合注意力机制的 SR-ACNN 算法。该算法将视觉领域的谱残差方法应用于时间序列异常检测领域，但原始的谱残差方法利用显著图上的传统的单一阈值来识别异常点，其过于简单且无法适应时间序列数据中的复杂和多变的异常形式。为解决上述问题，本章将注意力机制和卷积神经网络相结合，可以对显著图中的重要特征进行加权，并动态判断阈值，提高异常检测的准确性和泛化性。最后，通过实验验证模型的有效性。

第三章　基于弱监督的深度强化学习异常检测算法

本章主要研究强化学习在时间序列异常检测领域的应用，相较于传统的监督学习方法，强化学习在该领域展现出了很多的优势。首先，它不需要依赖大规模的标注数据，而是通过与环境的交互来学习最优策略，这使得强化学习在难以获取标注数据的领域中更具适用性。其次，通过设置奖励机制，强化学习能够更好地识别时间序列数据中的异常特征。此外，强化学习还能够根据与环境的交互动态调整策略，以适应时间序列数据的不断变化。

时间序列异常检测可以被视为一个序列决策过程，而该过程可以通过强化学习中的马尔可夫决策过程（Markov Decision Process，MDP）来建模，从而将时间序列异常检测与强化学习相结合。强化学习具有泛化和增量自学习的特点[38]，因此，使用强化学习框架进行异常检测是可行的[39]。在强化学习中，智能体与环境相互作用，智能体根据当前的状态做出决策，并且能够从环境中获得反馈，从而学习和解决其决策策略。基于这些优势，本章提出了一种基于深度强化学习的时间序列异常检测方法。该方法采用第二章的 SR 方法作为奖励机制的一部分，通过与环境的交互来学习最优的异常检测策略。本章首先阐述了相关理论和数据预处理方法，然后提出了一种基于弱监督的深度强化学习算法，并详细介绍了算法的总体框架、Actor-Critic 算法以及强化学习的奖励机制。最后，通过实验验证了本章算法的有效性。

3.1 研究理论基础

3.1.1 强化学习

强化学习问题通常表示为马尔可夫决策过程，其可表示为五个元素的元组：<S,A,P,R,γ>。在时刻t，假设智能体处于状态$s_t \in S$，并根据可从s_t映射到a_t的策略π选择动作$a_t \in A$。智能体会得到一个即时的奖励r_t，其中$r_t = R(s_t, a_t)$，并根据状态转移的概率函数$P(s_{t+1} \mid s_t, a_t)$获取下一个状态。这些与环境之间的相互作用导致了策略轨迹τ的形成，直到智能体达到最终状态。智能体的目标是在每个状态s_t下获得的奖励$R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$时，最大化奖励期望$E[R_t]$。其中$\gamma$表示从 0 到 1 的折扣系数，用来当做奖励对未来奖励重要性的衡量指标。智能体不断从经验中进行学习，最终获得最优策略π^*。强化学习的一般框架如图 3-1 所示。

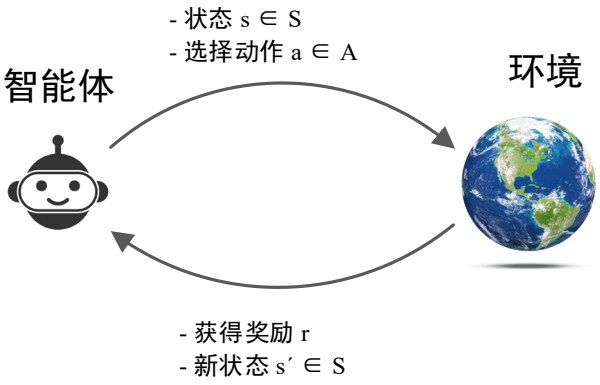

图 3-1　强化学习模型的一般框架

在强化学习的框架中，根据决策方式的不同，可以将其划分为基于策略的方法（policy-based）和基于价值的方法（value-based）。基于策略的方法直接优化自身策略，使得制定的策略获得最大的奖励。而基于价值的方法则不同于基于策略的方法，它只需要通过维护一个价值表格或价值函数来指导自身行为，而无需制定显式策略，通过这个价值表格或价值函数来选取价值最大的动作。为对强化学习中某个状态的进行评估，定义了一个状态值函数 $V(s)$，它能够量化处于特定状态下的价值，其定义为：

$$V(s)=E[\sum_{t}\gamma^{t}r_{t}\,|\,s] \tag{3-1}$$

状态动作值函数用函数 $Q(s,a)$ 来表示，也具有十分重要的意义，如下：

$$Q(s,a)=E[\sum_{t}\gamma^{t}r_{t}\,|\,s,a] \tag{3-2}$$

基于策略的强化学习，是直接对策略进行建模，使用参数 θ 优化目标策略 $\pi_{\theta}(s,a)=p(a\,|\,s,\theta)\approx\pi_{\theta}(a\,|\,s)$。选定状态 s，使用参数化的目标策略得到动作 a。其可应用在离散动作空间，智能体从当前状态 s 生成一个 $|A|$ 维的离散概率分布，当作衡量选取该动作的概率指标。基于此，智能体选择合适的动作。该方法的迭代公式如下：

$$\theta\leftarrow\theta+\alpha\nabla_{\theta}\log\pi_{\theta}(s_{t},a_{t})v_{t} \tag{3-3}$$

其中 α 是决定神经网络更新幅度的学习率，v_{t} 是在策略 π_{θ} 下的状态值函数，$\nabla\log\pi_{\theta}(s_{t},a_{t})$ 是在该策略下的得分函数。

在基于值的强化学习方法中，智能体通过最大化相应的价值函数间接优化目标策略，通常选中动作价值函数 $Q(s,a)$。off-policy 的基于值的学习，通常行为策略用于采样，目标策略用于优化，这种典型算法称为 Q 学习。目标策略 π 是贪婪的，其公式如下：

$$\pi(s_{t+1})=argmax_{a'}Q(s_{t+1},a') \tag{3-4}$$

当目标值和估计值的差值趋于 0 的时候，$Q(s,a)$ 就不再继续变化，Q 表趋于稳定，说明得到了一个收敛的结果，这就是算法想要达到的效果。

在智能体探索的过程中，执行的动作采用 ε-greedy 策略，其是权衡开发和探索（exploitation-exploration）的常用策略。在刚开始的时候，智能体不知道采取某个动作后会发生什么，所以只能通过试错去探索。开发是指直接采取已知的可以带来很好奖励的动作。这里面临一个权衡问题，即怎么通过牺牲一些短期的奖励来理解动作，从而学习到更好的策略。因此，提出 ε-greedy 策略，ε 就是权衡这两方面的超参数。ε 贪婪策略（epsilon greedy），其中智能体以 1-ε 选择贪婪行为，ε 的概率随机选择一个行为：

$$a=\begin{cases}\arg\max\limits_{a} Q(s,a), & \varepsilon\text{概率} \\ random, & \text{其他}\end{cases} \tag{3-5}$$

采取行动得到奖励后就可以用 Q 函数更新 $Q(s,a)$，直到训练停止，其迭代公式如下：

$$Q(s,a)\leftarrow Q(s,a)+\alpha[r+\gamma_{a'}^{\max}Q(s',a')-Q(s,a)] \tag{3-6}$$

在有限的状态和动作空间下，Q 学习的学习率 α 保持恒定，那么该算法将最终趋近于最佳 Q 值函数。

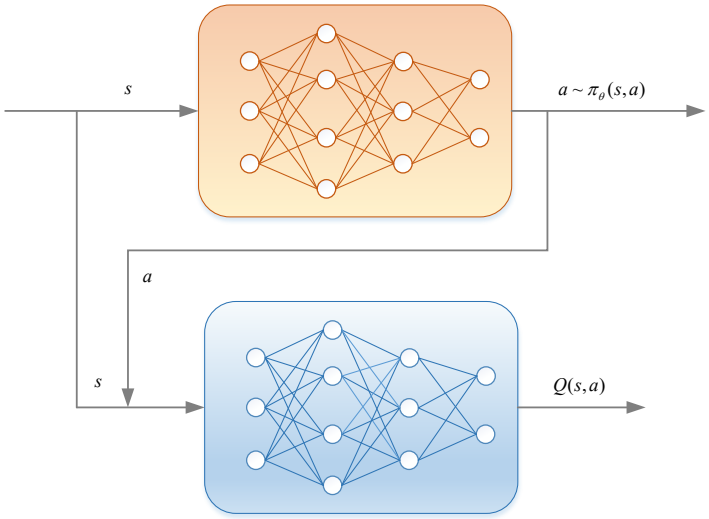

图 3-2　Actor-Critic 网络模型

如上图 3-2 所示，Actor-Critic 网络可以分为 Actor 网络（策略网络）和 Critic 网络（价值网络）两个部分。Actor-Critic 方法是强化学习领域的一种结合了基于策略模型（Actor）和基于值函数模型（Critic）的方法。Actor 网络负责学习在给定状态下选择动作的规则的策略。在连续动作空间中，Actor 网络通常使用 Policy Gradient 方法，因为它可以直接输出动作的概率分布或者动作的参数化表示，从

而在连续动作空间中选择合适的动作。Critic 网络通常使用 Q-learning 或 value-based 方法来学习一个值函数，这个值函数可以评估状态-动作对的期望回报。值函数基于当前策略进行更新，并反馈策略的好坏。Actor 网络根据当前的策略选择动作，而 Critic 网络则评估这个策略，在优化 Actor 网络的同时也进行自我优化。因此，Actor-Critic 网络能够在连续动作空间中高效地学习复杂的策略，并且通过值函数的单步更新提高学习效率。Actor-Critic 方法的优点在于，它结合了策略梯度方法的学习速度和值函数方法的学习效率，同时还可以通过值函数来避免 Policy Gradient 方法中可能出现的问题。在实际应用中，Actor-Critic 方法已经成功地应用于策略指定领域，如机器人控制、游戏智能和自动决策等。其算法流程如表 3-1 所示。

表 3-1　Actor-Critic 算法流程

算法：　Actor-Critic 算法
初始化策略网络参数 θ，价值网络参数 ω
for 序列 e=1→E do:
用当前策略 π^{θ} 采样轨迹 $\{s_1,a_1,r_1,s_2,a_2,r_2,...\}$
为每一步数据计算：$\delta_t = r_t + \gamma V_{\omega}(s_{t+1}) - V_{\omega}(s_t)$
更新价值参数：$\omega = \omega + \alpha_{\omega}\sum_t \delta_t \nabla_{\omega} V_{\omega}(s_t)$
更新策略参数：$\theta = \theta + \alpha_{\theta}\sum_t \delta_t \nabla_{\theta} \log \pi_{\theta}(a_t \mid s_t)$
end for

3.1.2 循环神经网络

循环神经网络是一种适用于序列数据建模的神经网络结构，其独特之处在于能够以递归方式处理输入序列，能够捕捉时间序列数据中的长距离依赖关系[40]。RNN 的主要特点是可以通过自身反馈来提取输入序列中的历史信息，并且基于先前信息生成相应的输出。

在 RNN 中，每个时间步的输入 x_t 和前一个时间步的输出 h_{t-1} 通过一个函数 f 转换为当前时间步的输出 h_t：

$$h_t = f(x_t, h_{t-1}) \tag{3-7}$$

RNN 的基本结构是将上一时刻的输出作为本时刻的输入，因此它可以自然地处理具有时间维度的数据。它通过循环单元来实现对序列数据的处理。循环单元是一种可以将当前输入和前一时刻的状态进行结合的神经网络结构，通常使用 tanh、ReLU 等激活函数来激活输出结果。图 3-3 是一个简单的 RNN 结构的示意图，如下所示：

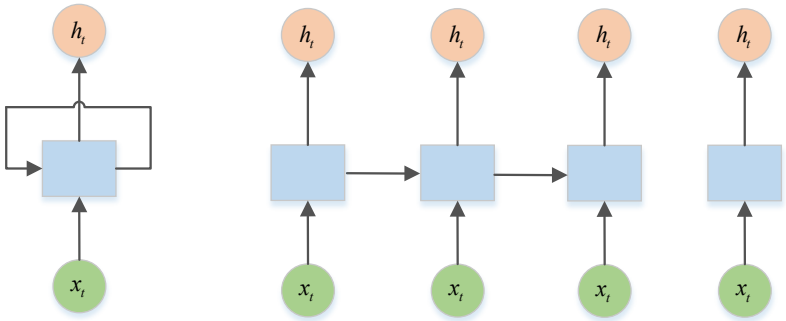

图 3-3　RNN 结构示意图

在循环神经网络中，$x(t)$ 代表输入序列中的第 t 个样本，$h(t)$ 表示在时间步 t 时的隐藏状态，而 $y(t)$ 表示相应的输出。$h(t-1)$ 指的是前一时间步的隐藏状态。循环神经网络的循环单元利用当前时间步的输入和前一时间步的隐藏状态来计算当前时间步的隐藏状态，这个隐藏状态随后被用来生成当前时间步的输出。RNN 有一个主要的问题，即长期依赖问题。为解决在处理长序列时，RNN 很难保留序列中前期的信息的问题，提出了 LSTM 网络。

1997 年 Hochreiter[41]提出了长短期记忆神经网络（Long-short Term Memory，LSTM），通过其改进了隐藏单元，有效解决了传统 RNN 在处理长序列数据时容易出现的梯度消失问题。它通过三个门控制器来选择性地保留和遗忘输入和状态，从而更好地捕捉时间序列中的长期依赖性。LSTM 的结构由一个存储单元、输入门、遗忘门和输出门组成。存储单元是 LSTM 的“记忆”，输入门决定新输入的占比程度，遗忘门决定旧信息的保留程度，输出门决定被传递给下一个时间步骤的信息。如图 3-4 所示：

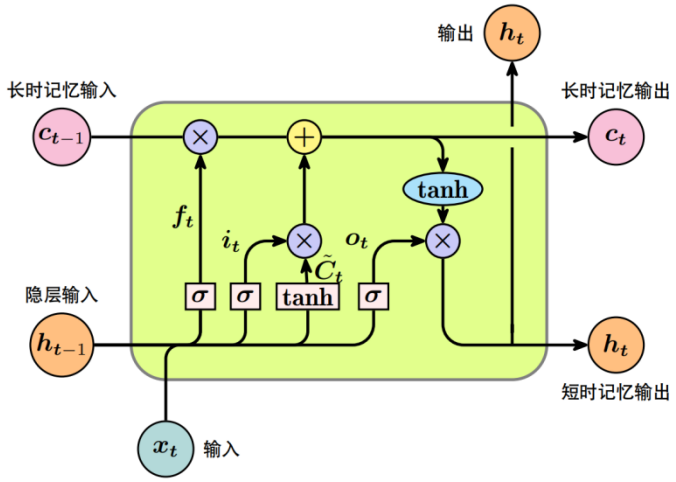

图 3-4　LSTM 具体结构

具体来说，LSTM 的计算流程如下：

（1）输入门：根据当前输入 x_t 和前一状态 h_{t-1} 计算输入门向量 i_t，控制当前输入 x_t 的加入程度。

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{3-8}$$

其中 W_i，U_i 和 b_i 是可学习参数，σ 是 sigmoid 函数。

（2）遗忘门：根据当前输入 x_t 和前一状态 h_{t-1} 计算遗忘门向量 f_t，控制前一状态 h_{t-1} 的遗忘程度。

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{3-9}$$

其中 W_f，U_f 和 b_f 是可学习参数。

（3）存储单元：根据当前输入 x_t，前一状态 h_{t-1} 和输入门向量 i_t 和遗忘门向量 f_t 计算存储单元向量 c_t。

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{3-10}$$

其中 $\odot$ 表示示逐元素相乘，tanh 表示双曲正切函数。

（4）输出门：根据当前输入 x_t 和前一状态 h_{t-1} 以及存储单元 c_t 计算输出门向量 o_t，控制输出 h_t 的程度。

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{3-11}$$

其中 W_o，U_o 和 b_o 是可学习参数。

（5）最终输出：根据存储单元 c_t 和输出门向量 o_t 计算当前状态 h_t。

$$h_t = o_t \odot \tanh(c_t) \tag{3-12}$$

3.1.3 变分自动编码器

早期研究已经广泛将自动编码器作为一种无监督的深度学习方法应用于异常检测领域[42-47]。自编码器是在传统主成分分析方法降维效率较低的背景下，由 Hiton 等人[48]提出一种通过神经网络实现的无监督学习模型。该模型由编码器（Encoder）和解码器（Decoder）两个核心部分组成，通过构建多层人工神经网络来实现对数据的降维和重构。并能够实现端到端的训练过程[49]。自编码器运用无监督学习策略，能够在未标记的大量数据中提取关键信息，实现对输入数据的高效降维和非线性特征提取[50]。

自编码器是前馈非循环神经网络，其通过训练能够输出与输入数据特征相符的重构数据[51]。其核心机制在于，自编码器的前半部分隐藏层负责将输入数据的维度进行降低，而后面的隐藏层则将这些维度增加至原始值。在这个过程中，输入数据被编码器处理的过程称为编码，而编码器输出的数据被解码器处理的过程则被称为解码。此外，自编码器的中间隐藏层的维度远小于输入层的维度，这使得自编码器在数据降维方面表现出远优于其他方法[52]。自编码器作为前馈神经网络的一种，因其出色的特征提取能力，可用作深度神经网络中的特征探测器进行

预训练[53]。然而，自编码器在性能上存在一定的局限性，例如对噪声的鲁棒性不足，以及在训练过程中可能对异常数据表现出敏感性，导致模型重构时的准确性下降[54]。自编码器的具体结构如图 3-5 所示：

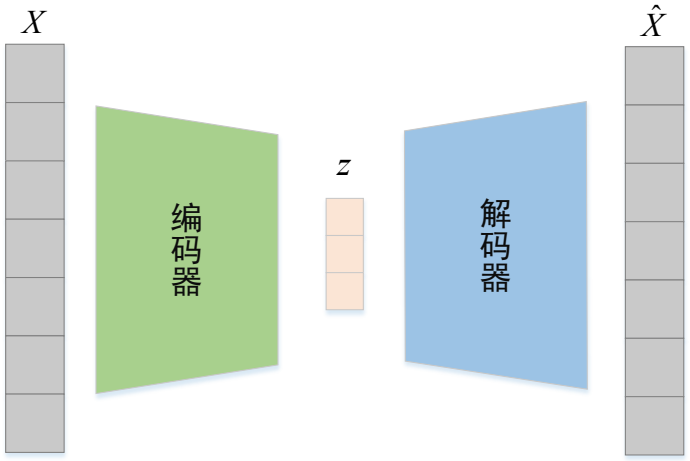

图 3-5　自编码器结构

变分自动编码器是生成模型的一种。这些方法的主要目标是从对象的学习分布中生成新的采样数据。2014 年，Kingm 等人[55]提出了一种生成模型，该模型可以从隐变量空间的概率分布中学习潜在属性并基于这些属性生成新的元素。

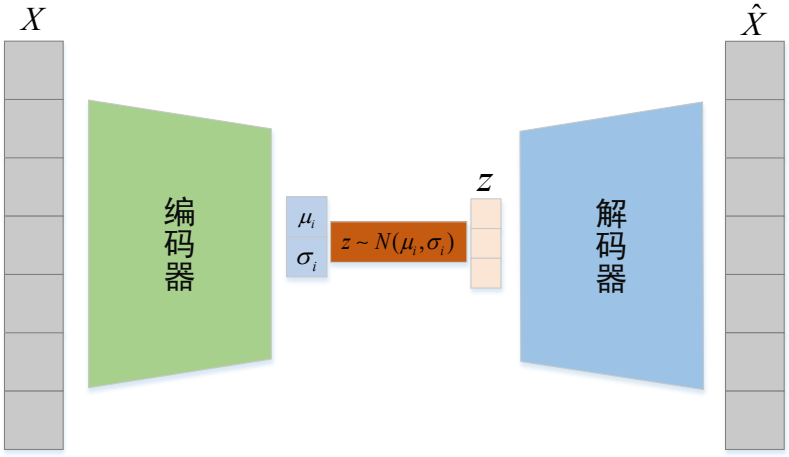

图 3-6　变分编码器结构

变分自动编码器核心架构同自动编码器，都是由编码器和解码器构成。其结构如图 3-6 所示，编码器计算输入数据 $X=\{X_1,X_2,...,X_i,...X_n\}$ 的低维均值 μ 和方差 σ^2，然后在隐变量空间中基于这些统计信息进行采样，生成相应的隐向量 $Z=\{Z_1,Z_2,...,Z_i,...,Z_n\}$，该隐向量随后被送入解码器，经解码器处理，最终生成新的数据 $\hat{X}=\{\hat{X}_1,\hat{X}_2,...,\hat{X}_i,...\hat{X}_n\}$。变分自编码器旨在创建一个能够通过隐变量 Z 来反

向生成目标数据模型[56]。这一过程中，假设输入 X 的数据分布是正态分布，通过模型的训练，使得生成的数据 $\hat{X}=G(Z)$ 也能符合正态分布，从而在模型训练得到的的概率分布与输入数据的真实分布之间建立起一种映射关系。

在模型中引入隐变量 Z_i 的目的是获取 X_i 的实际概率分布，将输入的数据 X_i 与正态分布匹配，在此过程中，从正态分布中采样得到 Z_i，然后对输入数据 X 进行编码操作，其分布形式为：

$$p(X_i)=\sum p(X_i \mid Z_i)p(Z_i) \tag{3-13}$$

其中，$p(X_i)$ 为输入的时间序列数据的原始分布，$p(Z_i)$ 为隐变量 Z_i 的概率分布，$p(X_i \mid Z_i)$ 为 X_i 的后验分布。为使生成器更好地进行解码，每个 X_i 都独立拥有一个正态分布。由于正态分布存在均值 μ 和方差 σ^2，所以构造一下两个神经网络进行拟合以获得每个 X_i 的专属均值和方差：

$$\mu_i=f_1(X_i) \tag{3-14}$$

$$\log\sigma_i^2=f_2(X_i) \tag{3-15}$$

由于拟合操作前需添加激活函数，而 σ^2 为非负数，故需对 $\log\sigma^2$ 进行拟合。经过神经网络的充分训练后，每变量 X_i 将获得其特定的均值 μ_i 以及方差 σ_i。在此基础上，隐变量 Z_i 的获取依赖于各 X_i 变量独立对应的正态分布进行采样。随后，通过解码生成网络重构出 $\hat{X}_i=g(Z_i)$，并采取最小化 $D(\hat{X}_i,X_i)^2$ 策略，旨在获得最好的原始数据 X 的重构效果。

在追求重构效果优化的目标下，通过对 $D(\hat{X}_i,X_i)^2$ 的最小化处理，目的是活的更为逼真的样本。由于隐变量 Z 并非直接由编码器推导而出，而是经过重采样流程，噪声数据对重构环节不可避免地产生影响。为提高重构精度，模型通过自适应学习策略有效滤除了噪声数据[57]，力求将隐变量的方差降至接近零，以此削弱随机性。这种调整使得采样结果呈现出唯一且确定的均值，而这个均值是由神经网络计算得出的数据特有属性。

为了防止模型简化为仅具有自编码特性的结构，并确保模型对噪声具有一定的鲁棒性，同时维持其强大的生成能力，在重构误差的计算中引入了 KL 散度（Kullback-Leibler divergence）项：

$$KL_{loss}=\frac{1}{2}\sum_{i=1}^{d}(-\log\sigma_i^2+\mu_i^2+\sigma_i^2-1) \tag{3-16}$$

通过引入 KL 散度，不仅确保了模型输出方差的非零性，而且避免了重构后的输出 $\hat{X}_i$ 仅仅是输入数据 X_i 的直接映射。变分自编码器的处理机制能够突出异常数据的特征，这对异常检测任务的后续开展十分有理。尽管 KL 散度的不对称性质可能导致模式坍缩[58]，这会加快模型收敛速度但可能会减少样本的多样性，然而，相较于噪声完全消失所带来的问题，KL 散度的这一副作用显得较不重要，

可以在一定程度上被接受。

在变分自编码框架中，由于编码过程输出的原始时间序列数据概率分布的均值和方差在直接采样时不可微分，这使得传统的梯度下降优化方法[59]无法有效更新网络参数。为解决此问题，变分自编码器采用了重参数化策略，在编码网络的权重训练中引入了高斯白噪声，以此对隐变量 z 的生成过程施加正态分布约束。以下是相应的数学描述：

$$z = \mu + \varepsilon \bullet \sigma \tag{3-17}$$

其中 ε 代表训练前所设定的初始参数，并遵循零均值、单位方差的正态分布规律，即 ε 遵循 N(0, 1)分布。

3.2 数据预处理

（1）数据清洗

因为本章提出的深度强化学习需要使用 VAE 作为部分奖励函数对时间序列数据进行重构，而在执行时间序列数据重构任务时，VAE 的训练通常依赖于对正常时间序列行为的学习。鉴于实际采集的数据往往缺乏明确的异常标注，并且可能包含大量的噪声信息，这使得直接区分正常数据与异常数据变得具有挑战性。为了确保模型能够有效地学习到时间序列的正常模式，必须在模型训练之前对原始数据进行彻底的预处理，包括异常值的识别与剔除以及噪声的消除，从而构建一个清洁的数据集，用于模拟正常模式。

鉴于目前还没有一种普适且完全准确的噪声和异常检测方法，本研究采纳了集成学习中多模型融合的思路，运用多种主流的简单高效的异常检测算法对训练数据集进行综合分析。通过整合不同模型的检测结果，可以最大化地识别并排除潜在的异常点和噪声，从而提高数据集的质量。

（2）数据集划分

本章的数据集由两个真实数据集和一个合成数据集组成，合成数据集与真实世界数据集均被划分为训练集与测试集，以便进行机器学习模型的训练和评估。由于本章是弱监督，故而只有小部分数据集有标签。将训练集的有标签的异常数据挑选出来作为 D_{la}，将剩余的数据集数据称为 D_u，其中 D_u 的数据集不保留标签。

3.3 基于弱监督的深度强化学习模型

本章提出了以强化学习为基础的异常检测模型，该模型会自动从时间序列数

据中学习并识别异常。时间序列中的观测值可以作为环境的状态，在每个时间步长智能体可根据当前观测值采取正常或异常的动作。该动作将触发环境的相应反馈，例如更新环境状态或提供奖励信号，从而影响智能体的下一个时间步长的动作。由于本章提出的弱监督深度强化学习模型其奖励函数包含 VAE 和 SR 方法，因此简称为 VS-DRL 模型。

3.3.1 基于弱监督的深度强化学习算法总体框架

本章提出的算法旨在通过 DRL 智能体与由训练数据构建的环境进行交互来学习。该算法通过设计采样函数和奖励机制，使得智能体可以基于一个较小的已标记异常数据集 D_{la} 进行学习，并在此基础上探索未标记数据中超出 D_{la} 范围的任何潜在异常。该算法的外部奖励函数 X 结合了已标记异常和可疑未标记异常的监督信号，以实现开发（利用已知的异常）和探索（发现新的异常）之间的平衡。在训练阶段，智能体通过开发和探索的过程来学习识别异常情况，并判断新的时间序列数据是否异常。

本章提出的方法框架总体如图 3-7 所示。智能体 A 从两个动作 a_0 和 a_1 中选择一个动作，其中 a_0 为正常状态选取的动作，a_1 为异常状态选择的动作。环境 E 由跟踪采样函数 S 和外部奖励函数 X 组成。

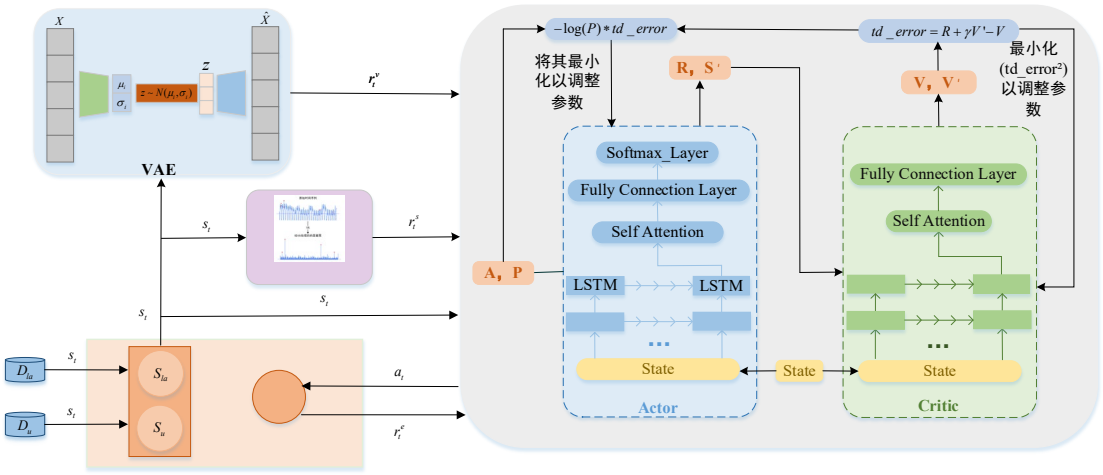

图 3-7　基于弱监督的深度强化学习框架

在每个时间步 t，智能体都与环境进行交互。状态采样函数 S 交替地从 D_{la} 和 D_u 中选取状态以训练智能体。当智能体接收到状态 s_t 后，它采取行动 a_t，然后接收奖励 r_t。奖励函数分为从外部奖励函数 X 接收外部奖励 r_t^e 和内部奖励函数 r_t^i。内部奖励基于状态的异常来奖励智能体，以鼓励对 D_u 的无监督探索，以识别新的未标记异常状态。在训练阶段，训练智能体以使这两个奖励的总和最大化。

在本研究框架中，强化学习的核心组件包括智能体、环境和奖励机制。强化

学习的过程从时间步 t 开始，智能体基于当前状态 s_t 和所采用的策略 π，选择一个动作，该动作可以是正常或异常的分类。之后，环境以奖励 r_t 的形式对所采取的行动作出反应。在每个时间步骤 t 中，智能体接收一个新的状态和奖励，并最终学会分析策略并执行最佳操作。智能体的训练目标是为了发现能够最大化折扣奖励的最优策略，对于时间序列异常检测，该公式的具体表述如下。

状态（State）:由于智能体的当前动作不仅受到由当前时刻点状态的影响，还受到先前时刻点状态的影响，因此状态定义为 $s_t =< x_{t-m+1}, x_{t-m+2}, ..., x_t >$。因此确定当前动作 a_t 需要过去 m 个时间戳。状态空间 S 被视为无限，因为实际的时间序列具有多种变化。

动作（Action）:操作空间定义为 $A = \{a_0, a_1\}$，其中 a_0 代表选择动作 0，a_1 代表选择动作 1。给定时间步长 t 的状态 s_t,时序异常检测器选择 a_0 代表当前值 x_t 被视为异常，选择 a_1 代表当前值 x_t 被视为正常。

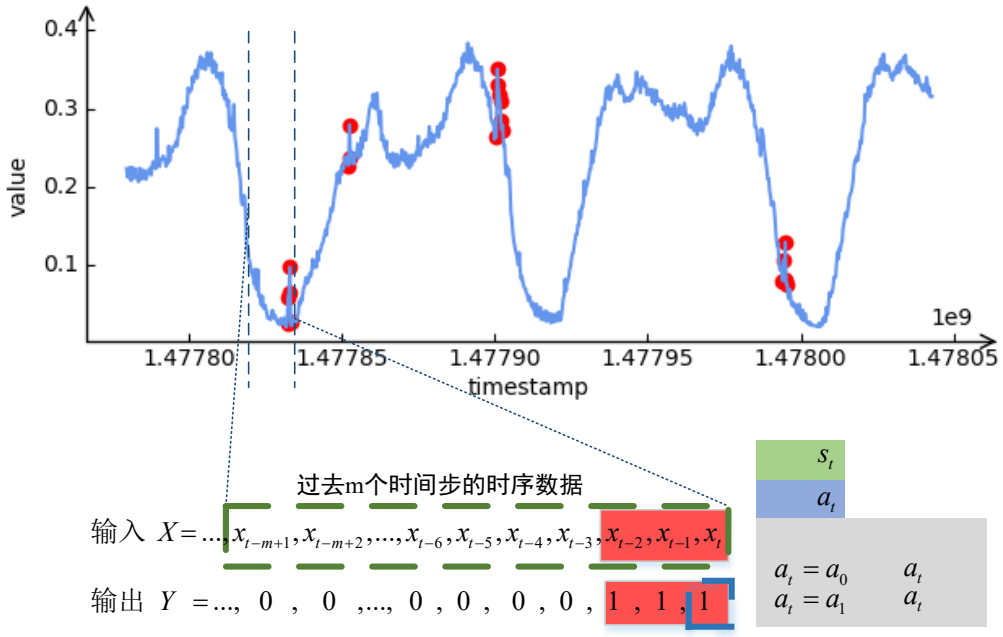

图 3-8　基于强化学习的异常检测器的状态和动作示例

在每个时间步长 t，本算法定义的状态和动作的示例如图 3-8 所示。异常检测器通过选择一个二进制动作 $a_t \in \{a_0, a_1\}$ 来确定用当前时间 x_t 是否是异常。与仅仅选取当前时间步长 x_t 作为状态不同，本章选取过去 m 个时间步长的时间序列 $\{x_{t-m+1}, ..., x_t\}$ 作为状态。因此，智能体可以综合考虑时间序列数据中的时间变化。与传统的单点异常检测方法不同，本章的方法更专注于对整体时间序列数据的异常行为进行建模和检测。这是因为在实际应用中，异常通常是由一系列相关的数据点共同引起的，而不仅仅是单个数据点的离群值。通过应用强化学习，能够处

理时间序列数据的时间依赖性，充分利用过去的决策和观察信息，从而更准确地捕捉到整体异常模式。这种方法能够更好地应对复杂的异常检测问题，并提高检测的准确性和实用性[60,61]。

3.3.2 改进 Actor-Critic 算法

由 Vaswani 等人[62]提出的作为序列建模的 Transformers 架构在文本翻译和图像分类等任务中表现出色。但是，进行序列建模的数据都是非时间性的。而完全依赖注意力机制 Transformers 缺乏递归建模，导致其无法有效捕获输入序列的时间信息和一些重要特征。因此，为了更好地建模时间序列数据并更有效地捕捉长期依赖关系，本章将 LSTM 和自注意力机制[63]相结合。使用自注意力组件对数据进行建模时，模型会关注序列的一部分，而自注意力机制则可以将序列的不同位置联系起来，以对序列不同部分之间的依赖关系进行建模。本章中应用的自注意力机制与 Transformers 中应用的自注意力机制相同，这也是 Transformers 架构的中心机制。由于 LSTM 具有更好的时间建模能力，结合自注意力机制可以更好地捕捉输入序列的长期依赖关系，从而实现更稳健的时间序列数据建模。

如下图 3-9 所示，其显示了 VS-DRL 模型的 Actor-Critic 部分的内部结构。其由 LSTM 实现的递归神经网络用于提取状态内的顺序信息，并将编码的特征输出到下一个单元。采用双层 LSTM[64]，上一层 LSTM 的隐藏状态作为输入提供给下一层 LSTM，有助于提供更丰富的特征表示，可以更好地学习和捕捉长期依赖关系，从而提高模型在长序列上的性能。LSTM 擅长捕捉长期依赖关系，而 Self-Attention[65]能够在无需先验信息的情况下建模长距离依赖。通过将这两者结合使用，可以更好地捕捉句子或文档中的长距离依赖，从而提高模型对上下文的理解和表达能力。VS-DRL 模型的 Actor-Critic 部分使用双层 LSTM 和 Self-Attention 结合，以提高模型对上下文的理解和表达能力。状态输入由估计策略的 Actor 网络和近似状态值函数的 Critic 网络类似地处理。将 RNN 和 Self-Attention 的输出作为输入，全连接层产生两个值，即：$P(a=0|s)$ 和 $P(a=1|s)$，指示 Actor 网络两个动作的可能性，并在 Critic 网络中输出当前状态的值。存在一个 softmax 层，将全连接层的归一化到[0,1]的范围内，并根据 Actor 网络中的概率给出当前状态的最终动作值。Criic 网络使用 TD 误差算法更新，其参数为 td_error，公式如下所示：

$$-log(P)\times td\_error \tag{3-18}$$

Actor 网络使用 Policy Gradient 算法更新，其计算如下：

$$td\_error = R + \gamma V' - V \tag{3-19}$$

其中 TD 差分项 td_error 来自 Critic 网络。

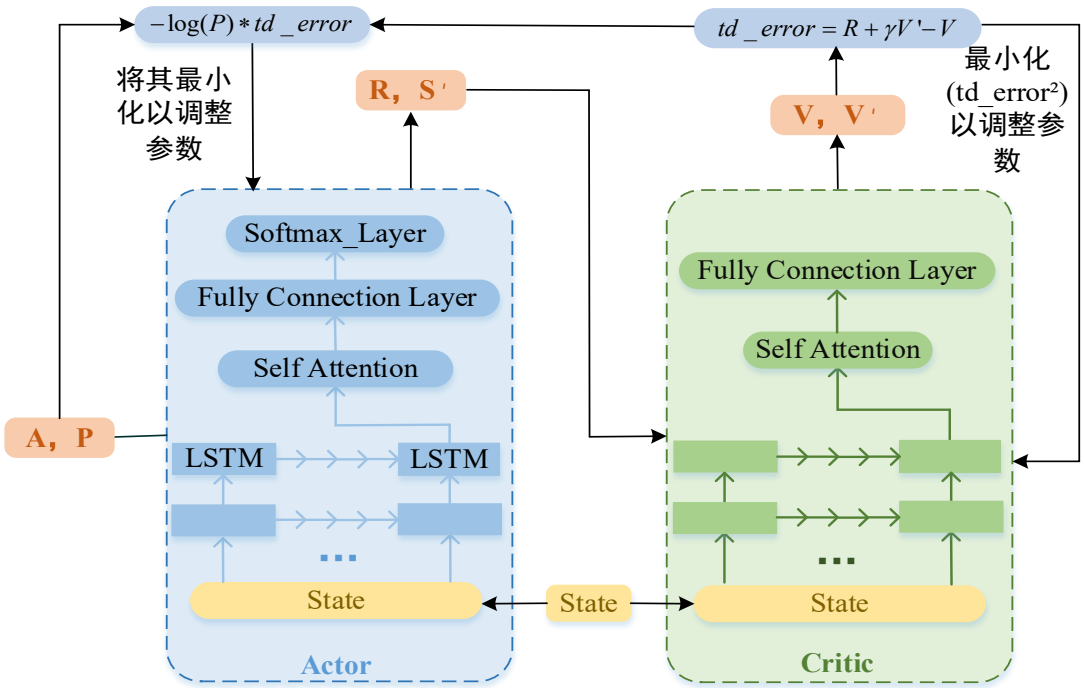

图 3-9　优化 Actor-Critic 算法

3.3.3 深度强化学习模型的奖励机制

在 DRL 框架中，VAE 和 SR 扮演着探索未标记数据集 D_u 的角色。VAE 和 SR 通过为该数据集提供一种隐式的监督信号，促进了对未标记数据的无监督探索。在本章的框架中，VAE 是在正常数集上训练而成的。当输入序列与正常模式存在差异时，VAE 在重建这些序列时无法达到与正常数据相同的精度。因此，通过比较重构序列与原始输入序列之间的差异，可以推导出一个异常分数。VAE 和 SR 构成了本方法中弱监督学习模块的关键，其独特之处在于能够在无需特定异常类型训练的情况下，识别并发出潜在异常的信号。创建环境 E 是为了使代智能体能够利用 D_{la} 和探索 D_u。环境由状态采样函数 S 和外部奖励函数 X 组成。其奖励机制包括采样函数 S、外部奖励函数 X 和内部奖励函数。

（1）采样函数 S

采样函数 S 由两个采样函数组成，随机采样函数 S_{la} 和基于距离的采样函数 S_u，这两个函数是为处理不平衡数据集问题而制定的。S_{la} 从 D_{la} 中随机采样后续状态 s_{t+1}，以允许利用每个标记的异常状态的等效机会。S_u 根据变分自动编码器的潜在空间中当前状态 s_t 和其他 D_u 状态之间的欧几里得距离，S_u 从 D_u 采样 s_{t+1}，以使智能体能够以高效和有效的方式探索 D_u。S_u 定义为：

$$S_u(s_{t+1}\mid s_t,a_t;\theta)=\begin{cases} argmind(s_t,s;\theta) \text{ 如果 } a_t=a_1 \\ argmaxd(s_t,s;\theta) \text{ 如果 } a_t=a_0 \end{cases} \tag{3-20}$$

其中，$s \in S$，S 是来自 D_u 的状态的随机子集，θ 是编码器的参数，d 返回变分自动编码器的潜在空间中 s_t 和 s 之间的欧几里得距离。

当智能体选择异常动作 a_1 时，S_u 返回与当前状态最接近的状态。这允许智能体调查与可疑异常状态相当的状态。当智能体选择正常动作 a_0 时，它还将最远的状态返回到当前状态。通过这样做，智能体可以探索更多可能的异常状态。接近度是基于 VAE 的潜在空间中编码迹线之间的欧几里得距离计算的，如图 3-10 所示。

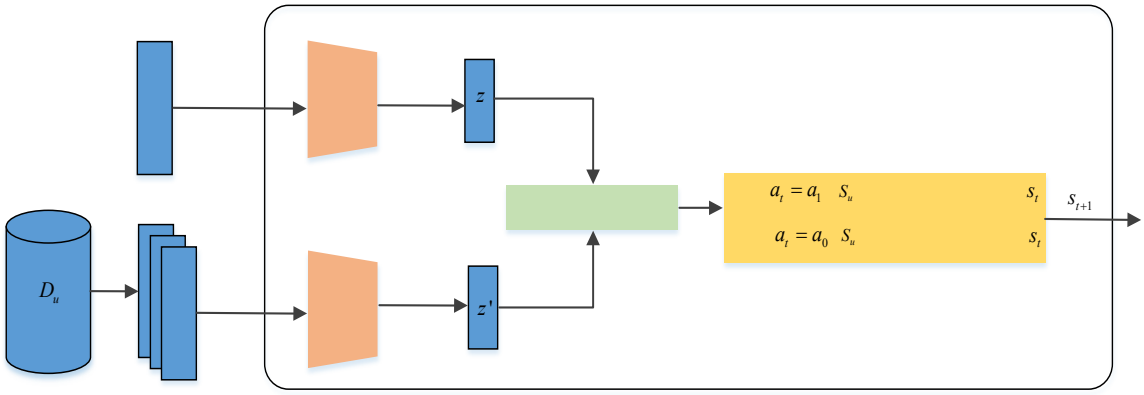

图 3-10　基于距离的采样函数 S_u 机制

出于效率的原因，对 D_u 的子集执行接近度计算。使用的子集是 D_u 的 15%。通过这种方式，S_u 在没有替换的情况下获得了比正常样本更多的异常样本，因此所提出的模型克服了数据不平衡的问题。采样函数 S_{la} 和 S_u 都以 0.5 的概率使用，以在搜索和开发之间取得平衡。

（2）外部奖励函数 X

奖励函数分为外部奖励和内部奖励。智能体收到的外部奖励 r_t^e，如下所示：

$$r_t^e = x(s,a) = \begin{cases} 1 & \text{如果} a_t = a_1, s \in D_{la} \\ 0 & \text{如果} a_t = a_0, s \in D_u \\ -1 & \text{其它} \end{cases} \tag{3-21}$$

只有当智能体正确地将已知异常状态标记为异常，并用−1 的值惩罚标记组的任何错误分类时，该函数才会给智能体正奖励。因此，此函数鼓励智能体充分利用标记的数据集 D_{la}。在 D_u 的情况下，智能体是中立的。VAE 将担任激励智能体探索未标记数据集 D_u 的任务。

（3）内部奖励函数

除了外部奖励 r_t^e 之外，智能体还从 VAE 和 SR 接收内部奖励，来鼓励智能体

对未标记数据集中的未发现异常状态进行无监督探索。VAE 原始输入 X 和重构输入 $\widehat{X}$ 之间的差被称为重构误差，其计算为：

$$\left\| X - \widehat{X} \right\| \tag{3-22}$$

通过使用 min-max 归一化方法，将重建误差重新缩放到 $[0,1]$ 范围内得到 re，此时，基于 VAE 的内部奖励 r_t^v 如下所示：

$$r_t^v = \begin{cases} re & \text{如果} a_t = a_1 \\ -re & \text{如果} a_t = a_0 \end{cases} \tag{3-23}$$

对于 t 时刻的输入状态 s_t，对 $\{x_{t-l},...x_t,...x_{t+l}\}$ 进行 SR 操作后使用 min-max 归一化至 $[0,1]$，得到 $\{p_{t-l},...p_t,...,p_{t+l}\}$，可得 t 时刻其基于 SR 的内部奖励 r_t^s 如下所示：

$$r_t^s = \begin{cases} p_t & \text{如果} a_t = a_1 \\ -p_t & \text{如果} a_t = a_0 \end{cases} \tag{3-24}$$

内部奖励函数定义如下：

$$r_t^i = r_t^v + r_t^s \tag{3-25}$$

基于以上奖励机制并考虑到 D_{la} 的开发和 D_u 的探索，智能体对每个状态获得的总奖励定义如下：

$$r_t = r_t^e + r_t^i \tag{3-26}$$

3.4 实验与分析

3.4.1 数据集

本章将合成数据集和真实数据集分为训练集和测试集，其中 80%在训练集中，其余 20%在测试集中。本章使用了三个数据集来评估 VS-DRL 模型，其中 KPI 和 NAB 是通常用于时间序列异常检测的公共数据集，而 IIDEP 是一个新的数据集，它收集了真实世界工业设备用电情况。KPI 数据集的具体情况在 2.6.1 节中已有介绍，本章使用的三个数据集的数据统计如表 3-1 所示。

表 3-1　数据集信息

	KPI	NAB	IIDEP
总点数	3004066	246814	45267
异常点数	79554(2.65%)	1344(0.62%)	764(5%)

Numenta Anomaly Benchmark（NAB）：是一个包含多个真实世界数据轨迹的数据集，包括纽约市温度传感器的读数、云机器的 CPU 利用率、服务请求延迟

和出租车需求等。然而，已知该数据集具有不正确异常标签的序列，例如 nyc-taxi trace，本章在实验中排除了这些序列。

Industrial Internet Device Electric Power（IIDEP）：随着能源转型的开展，电力数据的采集和分析显得极具价值。本章创建了一个电力数据集，记录了某物联网工业设备特定时间段内的三相电流、电压和有功功率指标，采样间隔为 1 分钟，并选择了 4 周的数据进行训练和测试。通过人工注入 5%异常数据，构建了一个带有标签的真实世界数据集，以便对本章所提出的方法进行更准确的评估和性能衡量。

3.4.2 实验设置与基准方法

滑动窗口的大小和状态的步数 m 都在 KPI 上设置为 720，在 NAB 上设置为 256，在 IIDEP 上设置为 128。对于改进 Actor-Critic 网络，LSTM 体系结构由输入层、输出层和隐藏层组成。折扣因子 γ =0.95，探索率 ε=0.05，Actor 网络的学习率 lr_a=0.0001，Critic 网络的学习率 lr_c=0.00001。

本文的基准方法有 Luminol、DSPOT、DQN、MAD-GAN 和 LSTM-VAE，其中除 DQN 外，均在 2.4.3 节中有介绍，DQN 方法介绍如下。

DQN：该方法是一种融合了深度神经网络和增强学习策略的算法，由 DeepMind 研究人员[66]首次提出，主要应用于具有复杂高维输入和连续动作空间的控制任务。

3.4.3 实验结果

为了验证 VS-DRL 在弱监督情况下建模复杂时序的能力，在 VS-DRL 和其他 5 个基线方法上进行实验，实验结果的 Precision、Recall 和 F1-score 如表 3-2、表 3-3 和表 3-4 所示。

表 3-2　VS-DRL 与其他方法在 KPI 数据集上的比较

数据集	模型	Precision	Recall	F1-score
KPI	Luminol	0.431	0.359	0.392
	DSPOT	0.624	0.439	0.515
	DONUT	0.368	0.394	0.381
	MAD-GAN	0.834	0.752	0.791
	LSTM-VAE	0.791	0.763	0.777
	VS-DRL	0.817	0.788	0.796

表 3-3　VS-DRL 与其他方法在 NAB 数据集上的比较

数据集	模型	Precision	Recall	F1-score
NAB	Luminol	0.585	0.503	0.541
	DSPOT	0.592	0.662	0.625
	DQN	0.602	0.589	0.595
	MAD-GAN	0.713	0.805	0.756
	LSTM-VAE	0.821	0.852	0.836
	VS-DRL	0.839	0.862	0.850

表 3-4　VS-DRL 与其他方法在 IIDEP 数据集上的比较

数据集	模型	Precision	Recall	F1-score
IIDEP	Luminol	0.601	0.632	0.616
	DSPOT	0.591	0.712	0.646
	DQN	0.623	0.596	0.609
	MAD-GAN	0.821	0.785	0.803
	LSTM-VAE	0.859	0.901	0.879
	VS-DRL	0.925	0.861	0.892

VS-DRL 模型在三个数据集的部分数据上的实验取得了不错的效果，相比于单纯的深度学习和强化学习，将两者相结合的深度强化学习算法效果更优，性能基本超过所有的对比方法，平均精确率为 85.6%，平均召回率为 83.7%，平均 F1 分数为 84.6%。

为了评估模型中每个主要组件的影响，本章进行了一系列实验，系统地禁用了每个组件，以观察结果对数据集性能的影响。首先，模型不使用 SR 方法作为奖励函数部分，该模型称为 VAE-DRL。其次，模型不使用 VAE 作为部分奖励函数。该模型称为 SR-DRL。随后，本章使用简单的 Actor-Critic 网络代替本章改进的 Actor-Critic 网络，该模型为未改进 Actor-Critic 的 VS-DRL，简称 UVS-DRL。最后，禁用平衡数据种类的采样函数，称之为 VS-DRL0。该实验的衡量指标为 F1-score，其结果如图 3-11 所示。其中禁用 SR 方法作为部分奖励函数对该模型的影响最大，其 F1 分数平均下降约 10.4%。其中未改进的 Actor-Critic 模型对其影响最小，平均 F1 分数下降 4.0%。由此可见，采用 SR 作为部分奖励函数更有利于训练处高效的模型。

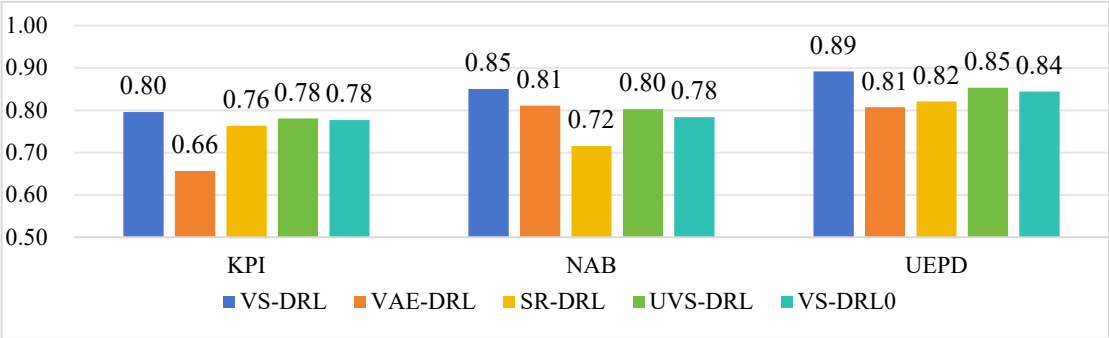

图 3-11　消融实验

3.5 本章小结

本章提出了一种使用小组标记异常数据的同时探索大量未标记数据的新的弱监督的时间序列异常检测模型。该模型引入谱残差方法作为部分内部奖励函数，其外部环境和内部奖励相结合的奖励机制可以使智能体根据时间序列数据的实际情况进行自适应学习，并且能够更好地适应不同时间段下的异常检测任务。VS-DRL 的 Actor-Critic 部分使用双层 LSTM 和 Self-Attention 结合，提高了模型对上下文的理解和表达能力。本文所提出的 VS-DRL 模型在三个不同行业的真实世界的数据集上的表现均优于基准方法，其中平均 F1-score 达到了 84.6%。

第四章　基于工业互联网的电力能源监测与异常诊断平台

4.1 开发背景与需求分析

4.1.1 开发背景

在前几章研究成果的基础上，本章将探讨如何将时间序列异常检测算法应用于实际的工业生产环境中。基于 Web 服务框架，搭建了一个基于工业互联网的电力能源检测与异常诊断平台。深度强化学习异常检测模块是该平台的核心模块，其训练所使用的数据集来自工业互联网设备的电力数据，记录了工业设备特定时间段内的三相电流、电压和有功功率等指标，采样间隔为 1 分钟。通过及时采集和处理工业互联网生产设备的用电信息，定期对模型进行训练评估后对相应时间序列数据进行异常检测，并将这些信息进行可视化展示，帮助相关人员更加准确地掌握设备的实时运行状态。这种实时监测和分析的方法能够有效地提升工业生产的效率和安全性，为工作人员提供及时的反馈信息，以便他们能够快速识别并解决可能出现的问题。

4.1.2 需求分析

基于工业互联网的时间序列异常检测系统是为了满足现代工业生产中对于实时监控和分析设备状态和生产过程的需求而设计的。系统需求可根据其关键功能划分为功能性需求和非功能需求两部分。

（1）功能性需求

数据采集：系统应具备实时监控硬件传感器状态的能力，并能够高效收集各类传感器的监测数据，为后续的分析处理提供数据信息。

数据预处理：系统需对采集到的监测数据执行预处理操作，包括数据清洗、缺失数据的删除或插补等，方便后续的数据处理和分析。

时间序列异常检测：系统采用本文提出的时间序列异常检测集成算法，对历史数据进行学习以优化检测模型，并对新数据进行异常检测，将检测结果保存并展示。为使管理人员或工作人员的详细了解设备情况，系统应具备有效的查询模块，以便查询和分析的异常检测结果。

系统管理：系统应实现日志管理功能，允许普通用户和高级管理员记录和查看系统操作日志，以便于对系统的日常运行情况进行监控和分析。系统应提供对

处理后监测数据及其他相关数据的存储功能，以便于后续的查询、分析和审计。

（2）非功能需求

非功能需求并不直接决定系统的具体功能实现，然而它们对系统的运行效率和性能有着重要的影响。本系统的非功能需求主要涵盖三个方面，即访问安全、兼容性强和维护简单。安全与访问控制是指系统须构建安全框架，以防止未授权的数据访问。此外，应实施细致的访问控制策略，以保证各用户根据其职责能够访问适当的信息和功能。系统整合与相容性是指系统应当具备与现行工业控制体系及企业信息系统协同工作的能力，且其运行不应对其他系统的稳定性造成影响。灵活性与维护便利性是指在系统设计时，需预见未来的增长和变化，确保系统能够容易地融入新增的功能和设备。同时，系统应便于维护，能够迅速地恢复服务并进行升级。遵循工业规范与法规是指系统开发过程中，必须遵循相关的工业标准和法律规范，以保障系统的合法合规性。

4.2 概要设计与功能设计

4.2.1 概要设计

（1）环境配置与技术选型

系统的环境配置主要涉及必要的硬件配置，系统硬件由计算机的各项组成部分构成，具体的配置参数如表 4-1 所示：

表 4-1　计算机配置参数

环境名称	环境配置
操作系统	Windows10 64 位
系统内存	16GB（2667MHz）
CPU	Intel Core i9-9700H
GPU	Nvidia GeForce GTX 2080Ti
GPU 显卡	12GB

在技术选型方面，本系统采用了前后端分离的架构，分别在以下四个层面进行详细介绍：前端、后端、数据科学计算服务以及数据管理服务。

前端开发工具选用的是 VSCode，而前端页面则是基于 Vue 框架搭建的。其核心理念是数据驱动，即当数据更新时，界面会自动更新以反映这些变化。这一特性使得系统在异常检测过程中能够实时可视化展示数据异常，提高了系统的实

时响应能力。后端开发工具则选用了 InteliJ IDEA，后端业务逻辑的实现基于 SpringBoot 框架。SpringBoot 是 Spring 框架的一个扩展，它是一个开放源代码的容器框架，集成了多种工具，可扩展性高，易维护。数据科学计算工具采用的是 PyCharm，数据的分析、处理及计算服务均基于 PyTorch 框架实现。数据管理服务包括数据的长期和临时存储。长期存储的数据分为两种：结构化数据，如管理员的信息登记，采用关系型数据库 MySQL 进行存储；另一类是系统需要实时监测的时间序列数据，这些数据使用专用的时序数据库 TimescaleDB 存储，TimescaleDB 的最大优势是支持全面的 SQL 功能。临时存储的数据通常是计算过程中产生的中间数据，这些数据使用缓存数据库 Redis 进行存储。

（2）系统架构

本异常检测系统的软件架构采用了 B/S 模型，通过网络浏览器为用户提供了一个直观的操作界面，而服务器端则承担了业务逻辑处理和数据处理与存储的任务。系统总体分为四层，包括客户端层、服务端层、数据持久层和数据采集层。具体的系统架构请参考图 4-1。

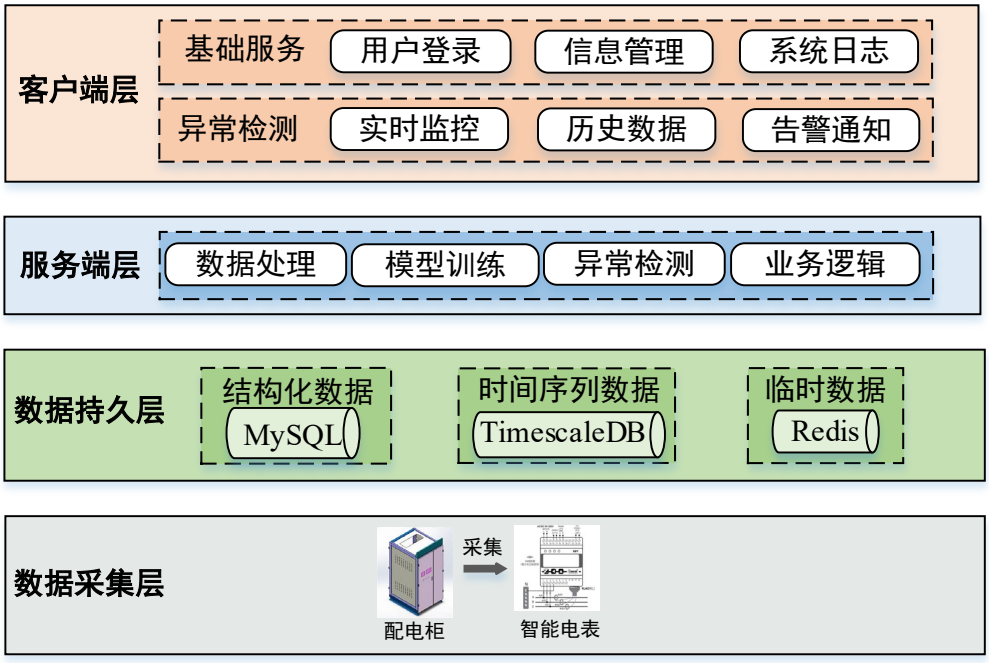

图 4-1　系统总体架构

客户端层负责呈现用户界面，包括了基础信息页面和异常检测页面。基础信息页面涵盖了用户登录、信息管理以及系统日志等界面；而异常检测页面则负责展示系统监控的实时数据、历史数据和告警通知。

服务端层由四个主要模块构成。首先是数据处理模块，接收来自数据采集层的原始传感器数据，并将其预处理成训练集数据或待检测数据，具体功能包括数

据清理、补插值、归一化和训练集样本构建。其次是模型训练模块，采用数据处理模块提供的训练集数据定期训练异常检测模型，并不断优化算法包，以提高监测的准确性和可靠性，动态监测设备状态。第三个是异常检测模块，这是系统的核心模块，负责实现数据的实时监控和异常情况分析，并在异常发生时提供报警服务。最后是业务逻辑模块，它提供了系统运行所必需的基础服务，例如用户通过账号密码和验证码登录系统，管理员进行信息的增删改查操作。

数据持久层负责与数据库直接交互，执行数据库的访问和操作，包括对数据表的 CRUD 操作。鉴于系统处理的数据类型包括结构化数据、时间序列数据和临时数据，数据持久层需要与 MySQL、TimescaleDB 和 Redis 三种数据库实现数据互操作性，并将数据传递至服务端层。

数据采集模块负责从工业互联网的电力设备中收集所需的电能数据，这些数据通过电表从配电柜中获取，这些电表能够记录相关的电能数据并将其存储在本地数据库中，随后通过 HTTP 协议传输到服务层模块。

4.2.2 功能设计

（1）系统基础功能设计

系统的基本功能是为了确保系统顺畅运行和提供必要的基础服务而实现的功能，这些功能并非本系统的核心特色。本部分将阐述这些功能的设计思路，分为四个主要方面进行介绍。

接口设计方面，系统的后端接口采用 MVC（Model-View-Controller）设计模式进行构建。在此模式中，M（Model）代表数据模型层，负责处理数据库的数据存取操作；V（View）代表视图层，负责界面的展示和用户交互；C（Controller）代表控制层，充当中央控制器，接收并处理用户请求，实现前后端分离。这种三层架构设计模式有助于分离关注点，提高系统的可维护性和可扩展性。

远程调用设计方面，本系统在与其他系统交互时，可能需要实现服务器之间的远程调用。因此，在后台设计中，采用了基于 HTTP 协议的声明式 WebService 客户端，并将其注入到 Spring 容器中，以简化远程服务调用过程。

后端校验设计方面，当用户在前端页面进行登录验证或执行管理操作时，后端采用了 JSR303 标准框架来进行参数校验。这样做可以防止恶意用户通过接口进行攻击，从而保护数据库结构的安全。

异常处理设计方面，这里的异常指的是系统运行过程中可能出现的程序异常，而非异常检测中的异常数据。本系统的异常处理遵循 SpringMVC 的异常处理机制，通过类注解和方法注解的方式，将所有异常统一交给全局异常处理器进行处理，确保了异常处理的统一性和一致性。

（2）异常检测功能设计

本系统的异常检测功能需要通过两大步骤来实现，第一步需要通过经过数据处理的历史数据对异常检测模型进行训练与评估，当模型通过了评估指标才能进行第二步的实时监控和异常检测，具体流程如图 4-2 所示：

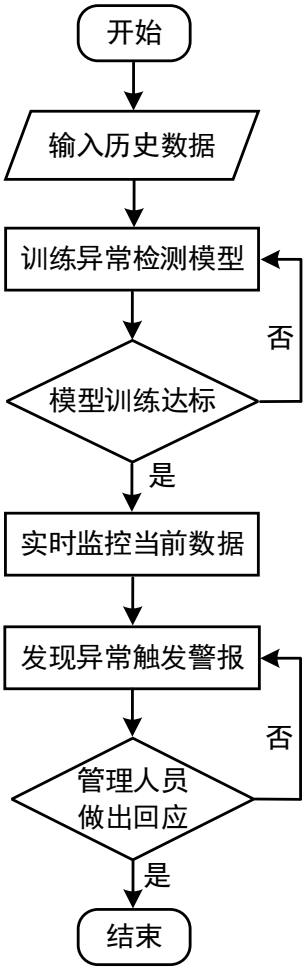

图 4-2　异常检测功能流程

异常检测系统的用户群体主要包括普通用户和高权限的管理人员。普通用户拥有数据上传、异常检测执行以及检测结果查询的基本操作权限。而管理人员则在普通用户权限基础上，享有管理用户账户、数据资源的高级管理权限，包括数据的增删改查等功能。

基于这些需求，将异常检测系统细分为四大功能模块：数据采集模块、数据处理模块、异常检测模块、系统管理模块。

数据采集模块负责从传感器采集工业设备对应的电能信息，并经以时间序列格式存储在 TimescaleDB 数据库中。

数据处理模块则首先对时间序列数据的连续性进行验证，并对存在缺失部分

的数据补插值。经过数据清洗和预处理步骤后，数据会被储存至数据库的特定表格中。其中训练集的数据和待检测数据分别存放在不同的数据表中。

异常检测模块包括三部分。第一模块涉及数据的实时监控功能，第二模块则专注于记录特征值的变化，第三模块负责异常检测并实施报警机制。实时数据监控涉及对监测系统生成的数据进行即时监控和记录，确保数据的时效性和完整性。特征变化记录功能则伴随着实时监控过程，通过绘制随时间变化的折线图，实现对数据特征波动情况的追踪，从而便于进行详细的数据分析和特征趋势评估。一旦系统侦测到异常数据，将激活异常检测报警机制，通过通知的形式，向注册的管理人员发出即时警报，以便迅速响应并采取必要的措施。

系统信息管理模块主要用来存储与维护两类重要信息。一类是对人员信息的管理，即对异常检测模块中提到的管理员信息进行登记与维护，这些信息包括用户 ID、用户名、邮箱号和手机号等，其中对于邮箱号和手机号的信息维护十分重要，需要及时更新与同步，因为涉及到异常检测模块中的报警信息发送。第二类信息为系统日志信息，系统日志会记录系统中所有的操作行为，其中包括管理员的所有信息变更手动操作，以及异常检测报警功能触发时的系统自动操作。

（3）数据存储功能设计

数据存储功能的设计主要包含数据库与服务器信息交互的设计。服务器通过一款半自动 ORM 持久层框架 MyBatis 实现与数据库层面的交互，ORM（Object Relation Mapping）指对象关系映射，对象即 Java 对象，关系即数据库中的关系模型。该框架封装了 JDBC（Java Data Base Connectivity），与 Spring 框架集成且兼容多种数据库类型，因此简化了很多数据交互操作。

4.3 系统实现

4.3.1 数据采集模块

本系统的数据采集模块采用 SDT640 导轨式安装的三相四线式多功能电表，该电表具备外接电流互感器，能够实时测量三相配电箱内的电流、电压、有功功率、无功功率等 30 个电气参数。通过 HTTP/MQTT/TCP 等多种数据传输协议，采集到的数据可以实时上报到服务端。在服务端，数据经过处理和解析后，会被存储到对应的数据库中，为模型训练和分析提供可靠的数据支持。表 4-2 中详细介绍了 SDT640 电表的具体参数。为了实现全天候的数据采集，将其安装在工业互联网设备的配电箱中，这样的安装方案能够确保 24 小时不间断地对工业设备用电数据进行采集。具体的接线方式和安装照片可以参考图 4-3。

表 4-2　SDT640 参数

参数名称	内容
通讯协议	802.11b/g/n
网络模式	Station (STA)
天线接口	2 米 3DBi 增益天线，SMA 接口
配网方式	网页配网，支持 PC、安卓和 IOS
工作模式	Modbus-RTU（虚拟串口）/Modbus-TCP/http 上报（json 格式）/Mqtt 上报（json 格式）/TCP 上报（json 格式）
接线	三相四线
电压	额定值：AC 57.7V/100V/220V/400V 等 过载能力：持续 480V；1 秒 1000V 功耗：<1VA
电流	额定值:5A 或 1A 过载能力：持续 1.2 倍；1 秒 10 倍 功耗：<1VA
频率	45~65Hz
精度	0.2%（电流、电压）；0.5%（频率：0.05Hz，无功能电能：1%）
电能脉冲	光耦电能脉冲，脉冲宽度：80ms±20ms
环境	工作温度：-20℃~+60℃ 存储温度：-40℃~+70℃ 相对湿度：5%~95%（无凝露） 海拔：<2500m
其他	尺寸：72×89×59mm（4P） 重量：175g

在完成接线和配置后，把电表设置成了 HTTP 上报模式，这样它就能够把 JSON 格式的数据上传到客户端了。为了准确地解释数据内容，表 4-3 中列出了每个数据项的参数代码和对应的参数名称。具体的数据上报示例可以参看图 4-4。在服务端，使用了阿里巴巴的 fastjson 库来解析这些数据。解析完成后，数据会按照既定的顺序存储到对应的数据库中，以确保数据的完整性和方便后续的数据处理与分析。

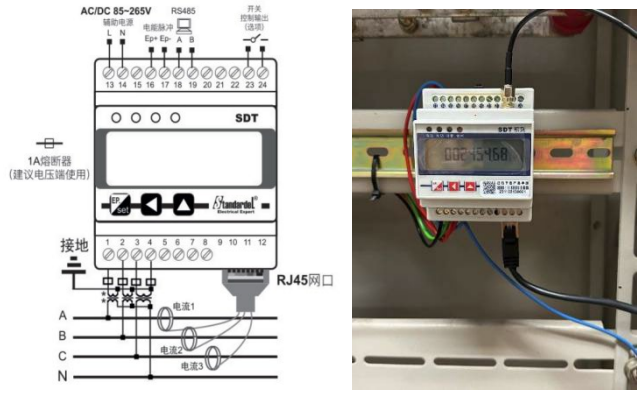

图 4-3　电表接线方式及实物图

表 4-3　数据项参数代码和参数名称

参数代码	参数名称
id	数据表号
e_id	设备编号
datetime	时间戳
Ia	A 相电流
Ib	B 相电流
Ic	C 相电流
Ua	A 相电压
Ub	B 相电压
Uc	C 相电压
Pa	A 相有功功率
Pb	B 相有功功率
Pc	C 相有功功率
P	三相总有功功率
Q	三相总无功功率
S	三相总视在功率
PF	三相总功率因数
Hz	频率
Econs	总能耗

[{"id":259,"e_id":231108130001,"datetime":1712474770,"Ia":12.3,"Ib":5.49,"Ic":0,"Ua":231.06,"Ub":231.17,"Uc":231.23,
"Pa":2.66,"Pb":0.97,"Pc":0,"P":3.68,"Q":1.14,"S":3.85,"PF":0.955,"Hz":50.06,"Econs :2474.66}]

图 4-4　数据上报格式

4.3.2 可视化界面展示

在本节中，将根据系统的功能需求分析以及功能设计内容，通过界面展示的方式，详细介绍系统中各个功能的实际实现过程。

图 4-5 呈现了系统用户的登录界面。在登录过程中，用户需输入其账号、密码进行身份验证。只有在数据库确认信息准确无误后，用户才能获准进入系统的主操作界面。

图 4-5　系统用户登录页面

用户登录成功后将进入工业设备监测云平台的核心入口，云平台的主页面呈现了项目信息设备信息、近期及当日设备的各项指标，如图 4-6 所示。系统可添加传感器或其他硬件设备，其操作界面如图 4-7 所示。

图 4-6　系统操作主页面

图 4-7　系统添加设备界面

参数报表界面允许用户查阅及导出硬件传感器的相关数据，提高数据的透明性。用户可自定义查阅数据的起始时间和终止时间，并可以根据需求灵活选择传感器参数，实现个性化管理并有效地提高数据分析效率。该功能提供了历史数据的分析基础，支持用户做出更准确的业务决策，该功能的界面示意图如图 4-8 所示。

图 4-8　系统数据采集界面

实时监控界面可以帮助用户实时地查看电表的工作状态及其测量到的最新数据并获取最新信息。当采集终端数量较多时，该功能可以让相关工作人员无需打开配电箱，直接通过系统数据显示来完成抄表等工作，大大提高了工作效率和便利性。具体示例可见图 4-9。

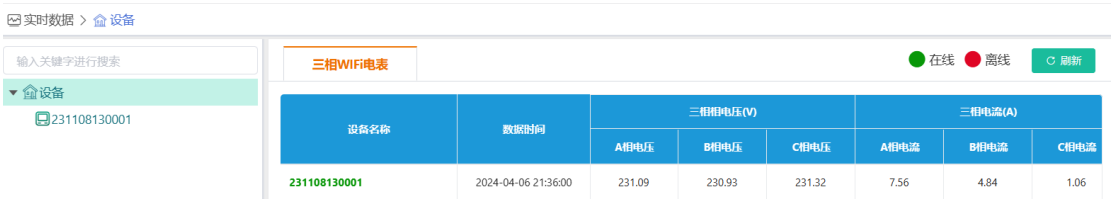

图 4-9　系统化的实时监控界面

一旦系统检测到异常数据，它会立即启动预警机制，并及时向用户发送告警通知。图 4-10 展示了异常检测界面的用户界面，所有告警信息都会通过异常检测页面显示，并且用户可以对历史信息进行查询，便于追踪和回顾。在此界

面中，用户可以查看系统检测出的异常状态通知情况以及告警类型占比统计和告警状态占比统计，其中异常状态通知情况包括告警 ID、告警名称、告警内容、设备名称、告警类型、告警等级、告警状态及发生时间等信息。如图 4-11 所示，用户可直接在界面中查看每条告警的详细信息，处理或回应告警信息，实现及时反应。

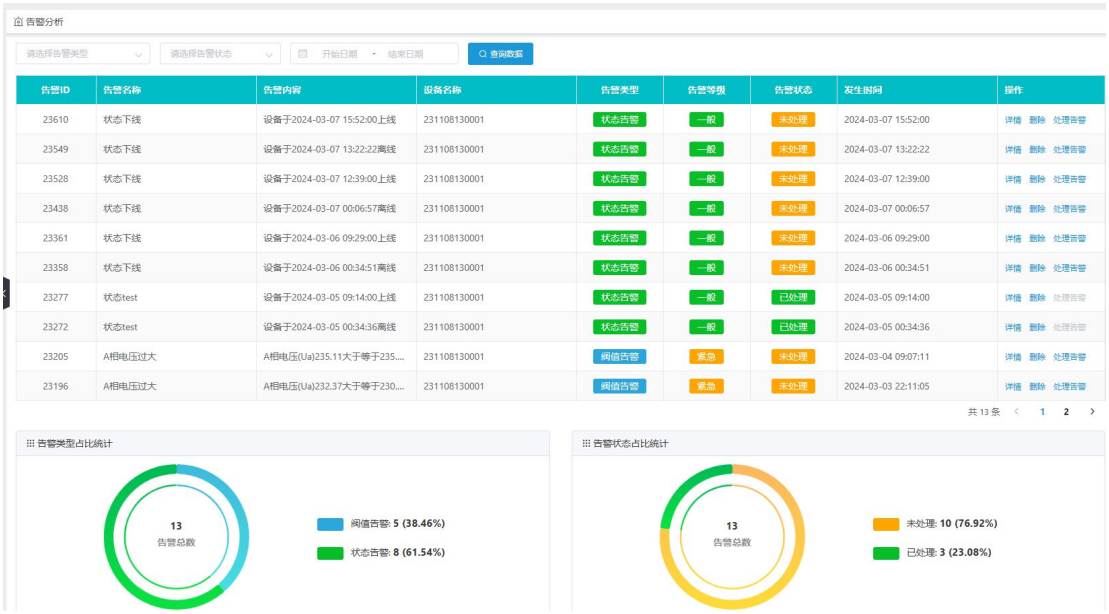

图 4-10　系统异常检测界面

图 4-11　系统告警及告警处理页面

通过对历史数据的分析，可以监控传感器的运行状况，并辅助识别与分析其在使用中的异常情况。模型的训练也依赖于这些历史数据。因此，开发了一个历史数据查询功能，允许用户指定查询的开始和结束时间。在这个时间范围内收集到的所有数据都可以以列表和折线图的形式展示，用户还可以根据自己的需求筛选特定的参数进行比较。此外，为了提高系统的便捷性，增加了一个数据导出功能。具体的界面展示如图 4-12 所示。

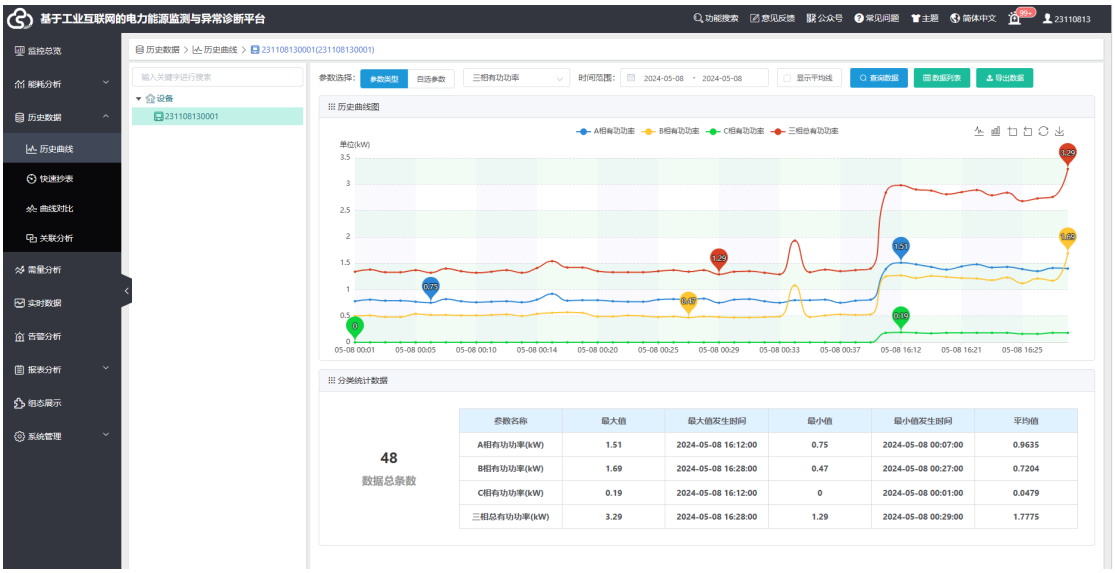

图 4-12　系统历史数据界面

4.4 本章小结

在本章中，基于第三章所提出的基于弱监督的深度强化学习异常检测算法，构建并实现了一个弱监督的深度强化学习异常检测系统。该系统能够对其他工业设备产生的时间序列数据进行分类存储和实时监控，并在发现异常数据时立即触发报警机制。本章首先阐述了开发背景和需求分析，揭示了构建该系统的必要性和实用性。随后，本文从总体架构设计和功能设计两个维度详细介绍了系统的环境配置、技术选型、架构设计以及系统应具备的各项功能。最后，本文展示了系统实现后的多样化功能界面。

第五章 结论与展望

5.1 主要结论

随着科技进步和数据量的快速增长，工业互联网产生的时间序列数据呈现出更高的维度、复杂的相关性和时变性。传统深度学习模型难以有效处理这些结构复杂的数据。此外，由于设备故障或操作失误，工业互联网数据可能会产生少量异常数据，这些数据虽然数量不多，但对工业设备正常运行的影响可能极为严重。因此，精确检测这类异常数据是一个长期存在的挑战。考虑到异常数据的样本稀少性，本文采用弱监督作为研究的基础，采用深度强化学习进行异常检测。以下是本文的主要研究结论：

（1）提出了一种融合注意力机制的 SR-ACNN 算法，该算法借鉴谱残差方法在视觉检测领域的优势，将其与融合注意力机制的卷积神经网络相结合，提高了算法的泛化性和高效性。该算法可以更有效地识别和监测异常情况，提高了异常检测的效率和准确性。

（2）提出了一种基于弱监督的深度强化学习算法，旨在提高时间序列异常检测的性能。该算法通过结合谱残差方法和变分编码器，设计了一个奖励机制，以帮助智能体在与环境的交互中学习更有效的异常检测策略。此外，为了解决样本不平衡问题，该算法引入了一种新的采样方法，使其倾向于从未标记数据集中选取异常样本。通过结合长短记忆网络和自注意力机制改善 Actor-Critic 算法，使得该算法能够更有效地捕捉时间序列数据的关键特征，从而提高异常检测的准确性。实验结果表明，该算法在性能上优于当前主流的五种基准异常检测算法。

（3）设计和实现了基于弱监督的深度强化学习异常检测系统，该系统采用前后端分离技术，能够实现对时序数据的实时监控、异常检测和报警等多种功能。通过该系统，可以有效提高工业互联网环境下复杂时间序列数据的异常监测效率和准确性，具有较高的实用价值和推广前景。

5.2 工作展望

本文深入研究了融合注意力机制的 SR-ACNN 算法和基于弱监督的深度强化学习算法，并对其在工业互联网时序数据异常检测领域的应用进行了改进。基于这些改进，设计并实现了一个异常检测系统，为用户提供不间断的异常检测服务。

尽管本文提出的弱监督强化学习模型 VS-DRL 在工业互联网时间序列异常检测方面取得了一定成效，但 VS-DRL 模型在训练过程中仍表现出不稳定的性能。此外，目前实现的异常检测系统在功能设计上还有待进一步完善。因此，未来的研究和工作中，将考虑以下几个方面的优化和拓展：

（1）针对本文提出的 VS-DRL 模型，未来研究专注于弱监督环境下强化学习算法的奖励机制的研究。期望设计一种更为高效的奖励函数，该函数可显著增强了模型的泛化能力，而不需要频繁地对模型进行训练。同时显著降低异常监测的成本，确保模型在训练过程中的性能稳定性。

（2）本研究针对的是工业互联网中的异常检测问题，但该模型也可以应用于其他领域。在未来的研究中，可以将该模型应用于其他领域，如金融、医疗等，以探索其在不同领域的应用效果。

参考文献

[1] Yepmo V, Smits G, Pivert O. Anomaly explanation: A review[J]. Data & Knowledge Engineering, 2022, 137:101946.

[2] Peng H K, Marculescu R. Multi-scale compositionality: identifying the compositional structures of social dynamics using deep learning[J]. PloS one, 2015, 10(4):e0118309.

[3] Javaid A, Niyaz Q, Sun W, et al. A deep learning approach for network intrusion detection system[C]//Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS). 2016:21-26.

[4] Chalapathy R, Chawla S. Deep learning for anomaly detection: A survey[J]. arxiv preprint arxiv:1901.03407, 2019.

[5] Borghesi A, Bartolini A, Lombardi M, et al. A semisupervised autoencoder-based approach for anomaly detection in high performance computing systems[J]. Engineering Applications of Artificial Intelligence, 2019, 85:634-644.

[6] Hilal W, Gadsden S A, Yawney J. Financial fraud: a review of anomaly detection techniques and recent advances[J]. Expert systems With applications, 2022, 193:116429.

[7] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553):436-444.

[8] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J]. arxiv preprint arxiv:1312.5602, 2013.

[9] Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey[J]. ACM computing surveys (CSUR), 2009, 41(3):1-58.

[10] 王天送, 张杰, 孙明明. 拉伊达准则在交通调查数据处理中的应用[J]. 西部交通科技, 2016 (4):96-99.

[11] Yu Y, Zhu Y, Li S, et al. Time series outlier detection based on sliding window prediction[J]. Mathematical problems in Engineering, 2014, 2014.

[12] Son S, Gil M S, Moon Y S. Anomaly detection for big log data using a Hadoop ecosystem[C]//2017 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, 2017:377-380.

[13] Wang Y, Huang K, Tan T. Group activity recognition based on ARMA shape sequence modeling[C]//2007 IEEE International Conference on Image Processing. IEEE, 2007, 3:III-209-III-212.

[14] Kadri F, Harrou F, Chaabane S, et al. Seasonal ARMA-based SPC charts for anomaly detection: Application to emergency department systems[J]. Neurocomputing, 2016, 173:2102-2114.

[15] Yu Q, Jibin L, Jiang L. An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks[J]. International Journal of Distributed Sensor Networks, 2016, 12(1):9653230.

[16] Zhang R, Zhang S, Muthuraman S, et al. One class support vector machine for anomaly detection in the communication network performance data[C]//Proceedings of the 5th conference on Applied electromagnetics, wireless and optical communications. World Scientific and Engineering Academy and Society (WSEAS), 2007:31-37.

[17] Das K, Schneider J. Detecting anomalous records in categorical datasets[C]//Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. 2007:220-229.

[18] Mukkamala S, Janoski G, Sung A. Intrusion detection using neural networks and support vector machines[C]//Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290). IEEE, 2002, 2:1702-1707.

[19] Malhotra P, Vig L, Shroff G, et al. Long Short Term Memory Networks for Anomaly Detection in Time Series[C]//Esann. 2015, 2015:89.

[20] Chen N, Tu H, Duan X, et al. Semisupervised anomaly detection of multivariate time series based on a variational autoencoder[J]. Applied Intelligence, 2023, 53(5):6074-6098.

[21] Li D, Chen D, Shi L, et al. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks[C]//International conference on artificial neural networks. Cham: Springer International Publishing, 2019:703-716.

[22] Xu H, Chen W, Zhao N, 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.

[23] Yu M, Sun S. Policy-based reinforcement learning for time series anomaly

detection[J]. Engineering Applications of Artificial Intelligence, 2020, 95:103919.

[24] Wu T, Ortiz J. Rlad: Time series anomaly detection through reinforcement learning and active learning[J]. arxiv preprint arxiv:2104.00543, 2021.

[25] He D, Kim J, Shi H, et al. Autonomous anomaly detection on traffic flow time series with reinforcement learning[J]. Transportation Research Part C: Emerging Technologies, 2023, 150:104089.

[26] Oh M, Iyengar G. Sequential anomaly detection using inverse reinforcement learning[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & data mining. 2019:1480-1490.

[27] Ren H, Xu B, Wang Y, 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.

[28] Hou X, Zhang L. Saliency detection: A spectral residual approach[C]. 2007 IEEE Conference on computer vision and pattern recognition. Ieee, 2007:1-8.

[29] Park Y, Jang J G, Kang U. Fast and accurate partial fourier transform for time series data[C]. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021:1309-1318.

[30] Mnih V, Heess N, Graves A. Recurrent models of visual attention[J]. Advances in neural information processing systems, 2014, 27.

[31] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[32] 胡韬.深度学习卷积神经网络研究概述[J].科技视界, 2020(9):2.DOI:CNKI:SUN:KJSJ.0.2020-09-058.

[33] Lin M, Chen Q, Yan S. Network in network[J]. arxiv preprint arxiv:1312.4400, 2013.

[34] Gupta M, Gao J, Aggarwal C, et al. Outlier detection for temporal data[J]. Synthesis Lectures on Data Mining and Knowledge Discovery, 2014, 5(1):1-129.

[35] Deldari S, Smith D V, Xue H, et al. Time series change point detection with self-supervised contrastive predictive coding[C]//Proceedings of the Web Conference 2021. 2021:3124-3135.

[36] 莹笑,项璇,杨彦红,等. 基于平均池化层时间卷积网络的轴承故障诊断方法[J]. 印刷与数字媒体技术研究, 2024, (02): 85-91+115.

[37] Siffer A, Fouque P A, Termier A, et al. Anomaly detection in streams with extreme

value theory[C]//Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017:1067-1075..

[38] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. MIT press, 2018.

[39] Huang C, Wu Y, Zuo Y, et al. Towards experienced anomaly detector through reinforcement learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2018, 32(1).

[40] 胡越, 罗东阳, 花奎,等. 关于深度学习的综述与讨论[J]. 智能系统学报, 2019, 14(1):1-19.

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

[42] Krajsic P, Franczyk B. Variational Autoencoder for Anomaly Detection in Event Data in Online Process Mining[C]//ICEIS (1). 2021:567-574.

[43] Amarbayasgalan T, Jargalsaikhan B, Ryu K H. Unsupervised novelty detection using deep autoencoders with density based clustering[J]. Applied Sciences, 2018, 8(9):1468.

[44] Schreyer M, Sattarov T, Borth D, et al. Detection of anomalies in large scale accounting data using deep autoencoder networks[J]. arxiv preprint arxiv:1709.05254, 2017.

[45] Luo T , Nagarajan S G. Distributed Anomaly Detection using Autoencoder Neural Networks in WSN for IoT[C]//2018 IEEE International Conference on Communications (ICC 2018).IEEE, 2018.DOI:10.1109/ICC.2018.8422402.

[46] Kieu T, Yang B, Jensen C S. Outlier detection for multidimensional time series using deep neural networks[C]//2018 19th IEEE international conference on mobile data management (MDM). IEEE, 2018:125-134.

[47] Zhou C, Paffenroth R C. Anomaly detection with robust deep autoencoders[C]//Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017:665-674.

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

[49] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[J]. Advances in neural information processing systems, 2014, 27.

[50] Yin C, Zhang S, Wang J, et al. Anomaly detection based on convolutional recurrent

autoencoder for IoT time series[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 52(1):112-122.

[51] Li L, Yan J, Wang H, et al. Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder[J]. IEEE transactions on neural networks and learning systems, 2020, 32(3):1177-1191.

[52] Pereira J, Silveira M. Unsupervised anomaly detection in energy time series data using variational recurrent autoencoders with attention[C]//2018 17th IEEE international conference on machine learning and applications (ICMLA). IEEE, 2018:1275-1282.

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

[54] Lu X, Tsao Y, Matsuda S, et al. Speech enhancement based on deep denoising autoencoder[C]//Interspeech. 2013, 2013:436-440.

[55] Kingma D P, Welling M. Auto-encoding variational bayes[J]. arxiv preprint arxiv:1312.6114, 2013.

[56] 李衍淼. 基于深度学习的网络异常流量检测技术研究[D]. 北京邮电大学, 2023.

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

[58] Madhukar A, Williamson C. A longitudinal study of P2P traffic classification[C]//14th IEEE international symposium on modeling, analysis, and simulation. IEEE, 2006:179-188.

[59] Risso F, Baldi M, Morandi O, et al. Lightweight, payload-based traffic classification: An experimental evaluation[C]//2008 IEEE International Conference on Communications. IEEE, 2008:5869-5875.

[60] Sayed G I, Tharwat A, Hassanien A E. Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection[J]. Applied Intelligence, 2019, 49:188-205.

[61] Guo C, Ma Q, Zhang L. Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform[C]. 2008 IEEE conference on computer vision and pattern recognition. IEEE, 2008:1-8.

[62] Lee D D, Pham P, Largman Y, et al. Advances in neural information processing

systems 22[J]. Tech Rep, 2009.

[63] Zhong H, Lv Y, Yuan R, Yang D. Bearing fault diagnosis using transfer learning and self-attention ensemble lightweight convolutional neural network. Neurocomputing. 2022, 501:765-77.

[64] YIN C, ZHOU L. Unsupervised time series anomaly detection model based on re-encoding[J]. Journal of Computer Applications, 2023, 43(3):804.

[65] Elaziz E A, Fathalla R, Shaheen M. Deep reinforcement learning for data-efficient weakly supervised business process anomaly detection[J]. Journal of Big Data, 2023, 10(1):33.

[66] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J]. arXiv preprint arXiv:1312. 5602, 2013.