



## 航空飞行器的状态参数 异常检测

### 5.1 引言

目前,航空飞行器地面监控系统只能对超出阈值范围的参数给以报警,报警之后,留给航空飞行器操控员应急反应处理的时间较短。航空飞行器专家在对航空飞行器事故分析时需要逐段查看飞行曲线查找异常数据,这种方式耗时效率低,亟需一种能在各子系统状态参数超出阈值之前,能自动识别异常变化数据的系统,提早给航空飞行器操控员以异常提示,方便航空飞行安全部门及时处理异常情况,保障飞行安全。

针对航空飞行器智能化异常检测需求,开展基于数据驱动的航空飞行器状态智能监控研究,利用航空飞行器历史飞行数据,设计智能监控算法,监控航空飞行器动力装置、导航子系统、飞行控制子系统和电气子系统 4 大组成子系统的状态参数的变化情况,识别出飞行数据中的各子系统状态参数的异常变化序列数据,并给以预警提示,为航空飞行器实时操控决策和预防性维修决策提供数据支持,方便机务保障人员及时查找异常原因,避免飞行事故的发生,保证飞行安全。

如何从大量的航空飞行器历史飞行数据中识别出异常序列的数据模式是亟待解决的问题。针对航空飞行器飞行数据缺少标签而且异常数据较少的情况,一种可行的方法是采用无监督机器学习算法。传统的用于异常检测的一类机器学习算法是基于距离的检测算法,如 K-Means 聚类算法,Bay 等采用聚类方法进行航空发动机的异常检测,Budalakoti 等采

用 K-Means 聚类算法进行离散飞行数据的异常检测,该类方法首先计算飞行数据中点之间的距离,然后找出那些离绝大多数点都比较远的点,这些点即确定为异常点。Matthews 等在哥伦比亚号航天飞机灾难的调查中,利用聚类算法监测飞机左翼的温度传感器数据,分析其中的异常飞行模式。这类方法的最大缺点是算法对数据敏感,异常检测效果不稳定。

第二类比较流行的无监督异常检测方法是单类支持向量机(One Class Support Vector Machine, OC-SVM)。OC-SVM 先寻找一个高维核空间中的最优超平面,然后利用超平面分离正常数据和异常数据。Das 等利用 OC-SVM 进行异常检测, Sarah 等利用线性单类支持向量机与深度学习网络结合,进行高维大数据的异常检测,该方法的最大困难是空间内核的构建复杂度比较高,对异常值敏感,异常检测效果同样存在不稳定性。

第三类方法是基于深度学习的算法。利用长短时记忆(Long Short-Term Memory, LSTM)网络捕获数据之间的按照时间顺序的循环依赖关系。在给定的序列长度内,正常数据通常遵循这种依赖关系,而异常数据通常会违反这些依赖关系,使用预测误差来定义数据的异常分数,从而识别出异常数据。丁建立等在论文《基于深度学习的 ADS-B 异常数据监测模型》中提出利用深度学习网络 LSTM 来检测广播式自动相关监视(Automatic Dependent Surveillance Broadcast, ADS-B)时间序列数据中的异常, Weixin Luo 等在论文 *Remembering History with Convolutional LSTM for Anomaly Detection* 中提出利用卷积 LSTM 网络构建异常数据检测模型,该方法的主要问题在于对数据样本敏感,容易产生拟合异常数据的风险。Reddy 等在论文 *Anomaly Detection and Fault Disambiguation in Large Flight Data: A Multi-modal Deep Auto-encoder Approach* 中提出了一种基于自编码器(Auto Encoder, AE)算法飞行数据异常检测方法, AE 是一种前馈多层神经网络,由编码网络和解码网络组成,编码网络将高维数据空间转换为低维空间,解码网络将低维空间恢复数据作为输出,通过最小化重建误差训练编码和解码网络,利用重建误差作为异常判别分数依据,通过异常分数的高低检测异常。Reddy 等通过滑动时间序列窗口划分飞行数据,作为输入向量,利用 AE 算法进行飞行时间序列数据的异常检测。Wang X 等的论文 *adVAE: A Self-adversarial Variational Auto-encoder with Gaussian Anomaly Prior Knowledge for Anomaly Detection*、D. Park 等的论文 *A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Auto-encoder* 在 AE 的基础上引入了变分自编码器(Variational Auto Encoder, VAE)的概念,将变分与深度学习结合起来进行异常检测,该方法应用在异常检测问题上的主要困难在于确定低维空间的维度,存在偏差,有变分下界,不能很好地捕获样本的数据模式。

生成性对抗网络(Generative Adversarial Nets, GAN)是另一种深度学习生成模型,该模型由用于捕获正常样本数据分布的生成网络和用于估计样本异常概率的判别网络两部分组成。相比 VAE 方法, GAN 是渐进一致的,没有变分下界。近年来, GAN 在异常检测中应用越来越广泛。

本章首先针对航空飞行器的状态参数异常检测问题进行建模;然后针对问题,设计一种基于 LSTM-GAN 组合网络的异常检测模型;最后,利用美国明尼苏达州大学航空飞行器实验室共享的飞行数据集状态监控参数进行验证实验,验证模型的有效性。

## 5.2 问题建模

航空飞行器状态参数的监控问题是指通过分析历史飞行数据,发现其中与期望不符的异常数据模式。按照飞行数据异常发生时间的范围,飞行数据的异常分为瞬时异常和序列异常。飞行数据的瞬时异常是指在某个时间点出现的单个异常点或者某一短暂时间出现的异常点。最基本的飞行数据的瞬时异常检测方法是基于数据标准值范围的异常检测,通过比较各个参数值与相应的标准数值范围查找异常。这种方法简单,但是不能发现飞行数据的序列异常。飞行数据的序列异常是指在某段飞行时间内,飞行数据存在异常模式的子序列。如图 5.1 所示为某航空飞行器 6 个架次平飞巡航时高度变化曲线,从图中可以看出,flight 1~flight 4 的航空飞行器都在高度 270~360m 巡航平飞,而 flight 5 在时刻点 730~790 高度发生突变,flight 6 在时刻点 1300~1700 变为 400m 左右的飞行高度,这两处的飞行数据都属于高度异常序列数据,其产生原因可能是 GPS 信号发生错误或者高度传感器发生故障。

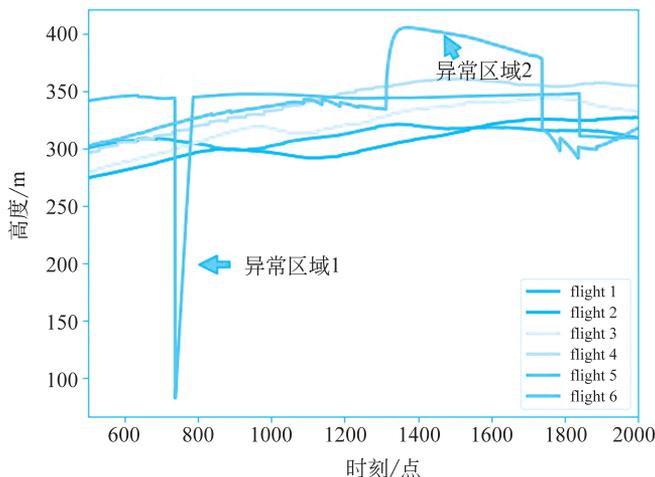


图 5.1 飞行数据序列异常示例

飞行数据异常检测问题就是从基于时间序列的飞行数据中找出异常序列数据集。因此,模型的输入为基于时间序列的飞行数据, $n$  个状态参数记为  $v_1, v_2, \dots, v_n$ ,  $m$  个时刻点记为  $t_1, t_2, \dots, t_m$ , 则  $n$  个状态参数在  $m$  个时刻点的监测值记作矩阵  $\mathbf{X}$ :

$$\mathbf{X} = \begin{matrix} & v_1 & v_2 & \cdots & v_k & \cdots & v_n \\ \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2k} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i1} & x_{i2} & \cdots & x_{ik} & \cdots & x_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mk} & \cdots & x_{mn} \end{bmatrix} & \begin{matrix} t_1 \\ t_2 \\ \vdots \\ t_i \\ \vdots \\ t_m \end{matrix} \end{matrix} \quad (5-1)$$

式中,  $(x_{i1}, x_{i2}, \dots, x_{ik}, \dots, x_{in})$  记作  $\mathbf{X}_i$  ( $i=1, 2, \dots, m$ ), 表示  $n$  个状态参数  $v_1, v_2, \dots, v_k, \dots, v_n$  在时刻点  $t_i$  的监测值,  $x_{ik}$  ( $k=1, 2, \dots, n$ ) 表示  $t_i$  时刻航空飞行器传感器采集得到的状态参数  $v_k$  的监测值。

飞行数据异常检测模型的输出为  $n$  个状态参数  $v_1, v_2, \dots, v_k, \dots, v_n$  的异常数据序列集合  $A = \{A_{v_1}, A_{v_2}, \dots, A_{v_k}, \dots, A_{v_n}\}$ , 其中, 状态参数  $v_k$  的  $p$  段异常数据序列可表示为集合  $A_{v_k}$ :

$$A_{v_k} = \{A_{v_k,1}^{\text{seq}}, A_{v_k,2}^{\text{seq}}, \dots, A_{v_k,p}^{\text{seq}}\} \quad (5-2)$$

式中,  $A_{v_k,j}^{\text{seq}}$  ( $j=1, 2, \dots, p$ ) 表示一段连续时刻点  $t_a, t_{a+1}, \dots, t_{a+b}$  的异常数据序列, 表示为向量  $\mathbf{A}_{v_k,j}^{\text{seq}}$ :

$$\mathbf{A}_{v_k,j}^{\text{seq}} = \begin{matrix} v_k \\ \begin{bmatrix} x_{a,k} \\ x_{a+1,k} \\ \vdots \\ x_{a+b,k} \end{bmatrix} \end{matrix} \begin{matrix} t_a \\ t_{a+1} \\ \vdots \\ t_{a+b} \end{matrix} \quad (5-3)$$

飞行数据的异常检测模型以基于时间序列的飞行数据矩阵  $\mathbf{X}$  为输入, 以异常序列集合  $A$  为输出, 构建异常检测算法, 正确识别异常序列, 为航空飞行器实时操控决策、各组成部件的预防性维修决策提供数据支持。

## 5.3 基于 LSTM-GAN 的异常检测模型

### 5.3.1 模型总体结构

基于 LSTM-GAN 进行飞行数据异常检测的模型结构如图 5.2 所示, 由模型训练和异常检测两部分组成。模型训练过程包括生成网络 (Generative Nets, GN) 和判别网络 (Discriminate Nets, DN) 的训练, GN 以随机生成的噪声时间序列  $Z$  为输入, 目标是尽可能伪装成真实的飞行数据时间序列以欺骗判别网络, 判别网络以真实飞行数据训练样本为输

入,尽可能辨别出真实飞行数据和随机噪声数据,GAN 的训练过程即为 GN 和 DN 不断博弈的过程,直至达到纳什均衡,得到的 GAN 网络中的 GN 将捕获训练序列的隐藏多元分布, DN 也被训练成高灵敏度地检测异常数据。异常检测部分以真实飞行数据测试样本  $X_{Test}$  为输入,利用训练好的 GN 计算测试样本的重构损失(Reconstruction Loss, RL),判别网络计算测试样本的判别损失(Discrimination Loss, DL),两部分相结合计算异常分数,根据异常分数判定是否为异常数据序列。

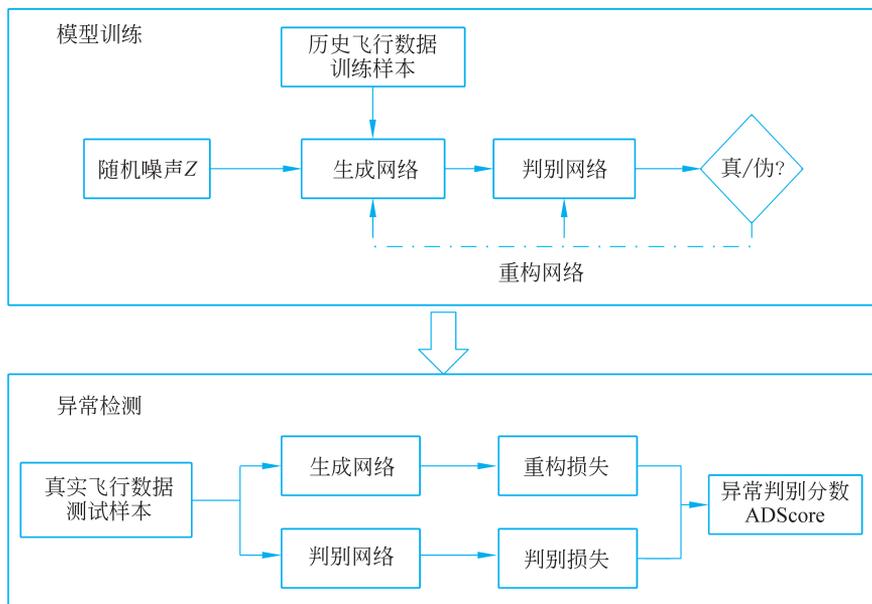


图 5.2 基于 LSTM-GAN 进行飞行数据异常检测的模型结构

### 5.3.2 模型训练结构

LSTM-GAN 模型训练结构包括 GN 和 DN 两部分,模型具体组成如图 5.3 所示。GN 建立两个映射函数  $G_1: X \rightarrow Z$  和  $G_2: Z \rightarrow X$ , 函数  $G_1$  是将真实飞行数据训练样本  $X$  进行编码降维到潜在空间  $Z$ , 函数  $G_2$  是将潜在空间  $Z$  生成随机时间序列数据  $X$ 。其中,  $Z$  服从多元正态分布, 即  $Z \sim N(0, 1)$ , 利用映射函数  $G_1$  和  $G_2$  实现时间序列数据  $X$  的重构, 即

$$\forall X_i \in X, \quad X_i \rightarrow G_1(X_i) \rightarrow G_2(G_1(X_i)) = \hat{X}_i \quad (5-4)$$

DN 建立两个判别函数  $D_x: X \rightarrow P$  和  $D_z: Z \rightarrow P$ ,  $D_x$  判定给定的样本  $X$  是真实数据样本的概率为  $P$ , 其目标是尽可能区分真实数据样本和  $G$  生成的随机数据样本;  $D_z$  判定给定的向量  $Z$  是由真实数据样本编码得到的概率为  $P$ , 其目标是尽可能区分真实样本编码得到

的潜在向量和随机数据样本。为了避免网络拟合异常数据风险,网络中引入掩蔽 Mask 层和 Dropout 层,随机对神经元进行遮蔽和丢弃,提高训练网络的稳定性。

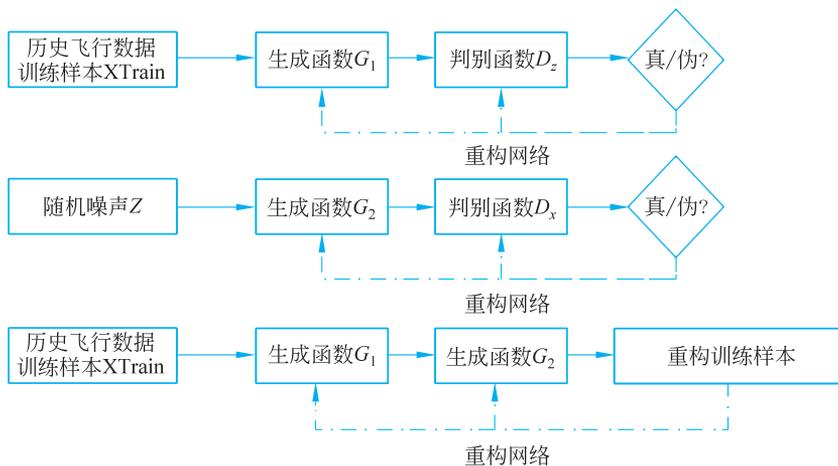


图 5.3 LSTM-GAN 模型训练结构

### 5.3.3 模型基础网络结构

异常检测算法的基础网络是 LSTM,该网络是在递归神经网络(Recurrent Neural Network,RNN)基础上发展而来的。RNN 网络组成结构如图 5.4 所示,通常由输入层、多层隐藏层和输出层组成,网络从输入层输入数据,并将数据信息传递给隐藏层,同一隐藏层中的节点进行信息传递从而构成循环学习,利用大量输入数据和输出数据训练网络,调整各层网络节点权重和偏置量,从而构建适应训练数据的网络模型。RNN 是一类专门用于处理和预测时间序列数据的神经网络,但经常出现梯度消失情况,偶尔发生梯度爆炸。为避免 RNN 的梯度消失和梯度爆炸问题,LSTM 网络被提出。

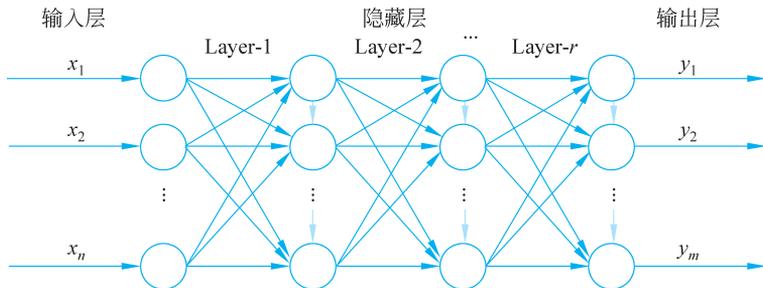


图 5.4 RNN 网络组成结构

LSTM 网络将 RNN 隐藏层的每节点神经元之间构建 3 个门——遗忘门  $f$ 、输入门  $i$  和

输出门  $o$ , 通过门结构控制信息在网络中的状态, 3 个门组成的记忆单元如图 5.5 所示。在时间步  $t$  时, LSTM 隐藏层的输入向量为  $\mathbf{x}_t$ , 输出向量为  $\mathbf{h}_t$ , 记忆单元为  $c_t$ 。

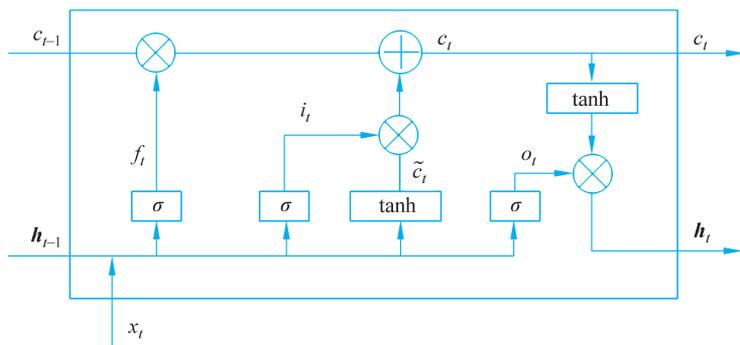


图 5.5 LSTM 网络门结构图

遗忘门:

$$f_t = \sigma(W_{xf}\mathbf{x}_t + W_{hf}\mathbf{h}_{t-1} + b_f) \quad (5-5)$$

输入门:

$$i_t = \sigma(W_{xi}\mathbf{x}_t + W_{hi}\mathbf{h}_{t-1} + b_i) \quad (5-6)$$

$$\tilde{c}_t = \tanh(W_{xc}\mathbf{x}_t + W_{hc}\mathbf{h}_{t-1} + b_c) \quad (5-7)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \quad (5-8)$$

输出门:

$$o_t = \sigma(W_{xo}\mathbf{x}_t + W_{ho}\mathbf{h}_{t-1} + b_o) \quad (5-9)$$

$$\mathbf{h}_t = o_t \otimes \tanh(c_t) \quad (5-10)$$

其中,  $\otimes$  表示向量元素对应乘。tanh 是双曲正切函数, 通过记忆单元和门控单元实现历史信息 and 长期状态的保存, 通过遗忘门使得 LSTM 能够学习连续任务, 并对内部状态进行重置。

### 5.3.4 模型目标损失函数

经典 GAN 的损失函数采用 Wasserstein 对抗损失(以下 log 函数的底数是 2, 后同):

$$WL = E_{x \sim P_X} [\log D_x(x)] + E_{z \sim P_Z} [\log(1 - D_x(G_2(z)))] \quad (5-11)$$

利用 Wasserstein 损失函数作为训练模型的目标损失函数存在梯度爆炸的风险, 为了避免该问题, 在训练判别网络时, 利用 Wasserstein 距离设计目标损失函数。

对于生成函数  $G_1: X \rightarrow Z$  和判别函数  $D_z: Z \rightarrow P$ , 目标损失函数设定为

$$\min_{G_1} \max_{D_z \in \mathcal{D}_z} V_Z(D_z, G_1) \quad (5-12)$$

其中:

$$V_Z(D_z, G_1) = E_{x \sim P_X} [D_z(G_1(x))] - E_{z \sim P_Z} [D_z(z)] \quad (5-13)$$

$D_z$  表示 1-Lipschitz 连续函数集, 即

$$\forall x_1, x_2 \in \text{dom} f, \|f(x_1) - f(x_2)\| \leq \|x_1 - x_2\| \quad (5-14)$$

1-Lipschitz 连续函数对函数值的上界进行了约束, 同时函数具有平滑性, 由此可以使网络权重的更新不会发生很大的变化, 这就降低了梯度爆炸的风险, 生成时间序列数据样本的分布与真实飞行数据的数据分布相匹配, 使得网络训练更加稳定可靠。

对于生成函数  $G_2: Z \rightarrow X$  和判别函数  $D_x: X \rightarrow P$ , 目标损失函数设定为

$$\min_{G_2} \max_{D_x \in \mathcal{D}_x} V_X(D_x, G_2) \quad (5-15)$$

其中:

$$V_X(D_x, G_2) = E_{x \sim P_X} [D_x(x)] - E_{z \sim P_Z} [D_x(G_2(z))] \quad (5-16)$$

训练 GAN 的目的是能较好地重构输入时间序列, 采用以上两个目标损失函数, 映射函数的搜索空间很大, 为了缩小搜索空间, 引入循环一致性损失函数, 如下式所示:

$$V_{L2}(G_1, G_2) = E_{x \sim P_X} [\|x - G_2(G_1(x))\|_2] \quad (5-17)$$

其中,  $\|\cdot\|_2$  表示原始样本  $x$  与重构样本  $\hat{x}$  差的 L2 范数。

综合式(5-13)、式(5-16)和式(5-17), 得到 GAN 的目标损失函数为

$$\min_{\{G_1, G_2\}} \max_{\{D_z \in \mathcal{D}_z, D_x \in \mathcal{D}_x\}} V_Z(D_z, G_1) + V_X(D_x, G_2) + V_{L2}(G_1, G_2) \quad (5-18)$$

### 5.3.5 模型算法描述

基于 LSTM-GAN 网络的异常检测算法描述如下。

#### 1. 算法输入

真实航空飞行器历史飞行数据样本  $X = \{X_1, X_2, \dots, X_t\}$ , 其中,  $X_i = (x_{i1}, x_{i2}, \dots, x_{ik}, \dots, x_{in})$  ( $i=1, 2, \dots, m$ ) 表示  $n$  个状态参数  $v_1, v_2, \dots, v_k, \dots, v_n$  在时刻点  $t_i$  的监测值,  $x_{ik}$  ( $k=1, 2, \dots, n$ ) 表示  $t_i$  时刻传感器采集得到的变量  $v_k$  的监测值。

#### 2. 算法输出

状态参数  $v_1, v_2, \dots, v_k, \dots, v_n$  的异常数据序列集合  $A = \{A_{v_1}, A_{v_2}, \dots, A_{v_k}, \dots, A_{v_n}\}$ 。

#### 3. 算法步骤

**步骤 1** 数据预处理, 主要包括噪声平滑、空缺值填补、数据归一化、划分数据集, 使用滑动窗口将训练样本 XTrain 划分为子序列, 滑动窗口大小设为  $m$ , 则训练样本序列分解为

$$X_{1 \sim m}, X_{m+1 \sim 2m}, \dots, X_{(n-1)m+1 \sim nm}$$

其中:

$$X_{1 \sim m} = \{X_1, X_2, \dots, X_m\}$$

$$X_{m+1 \sim 2m} = \{X_{m+1}, X_{m+2}, \dots, X_{2m}\}$$

$$\vdots$$

$$X_{(n-1)m+1 \sim nm} = \{X_{(n-1)m+1}, X_{(n-1)m+2}, \dots, X_{nm}\}$$

**步骤 2** 随机生成服从正态分布的噪声序列  $Z$ , 每组序列长度也为  $m$ , 即

$$Z_{1 \sim m}, Z_{m+1 \sim 2m}, \dots, Z_{(n-1)m+1 \sim nm}$$

其中:

$$Z_{1 \sim m} = \{Z_1, Z_2, \dots, Z_m\}$$

$$Z_{m+1 \sim 2m} = \{Z_{m+1}, Z_{m+2}, \dots, Z_{2m}\}$$

$$\vdots$$

$$Z_{(n-1)m+1 \sim nm} = \{Z_{(n-1)m+1}, Z_{(n-1)m+2}, \dots, Z_{nm}\}$$

**步骤 3** 训练 GAN。

置迭代次数  $i$  的初始值为 1。

当迭代次数  $i <$  最大迭代次数, 重复以下操作。

真实训练样本  $X$  生成潜在空间:  $X_{\text{Train}}L = G_1(X_{1 \sim m}, X_{m+1 \sim 2m}, \dots, X_{(n-1)m+1 \sim nm})$ 。

重构真实训练样本  $X$ :  $G_2(X_{\text{Train}}L)$ 。

随机噪声  $Z$  生成伪时间序列数据  $ZS = G_2(Z_{1 \sim m}, Z_{m+1 \sim 2m}, \dots, Z_{(n-1)m+1 \sim nm})$ 。

判别  $D_Z(X_{\text{Train}}L)$ 。

判别  $D_X(ZS)$ 。

依据目标损失函数

$$\min_{\{G_1, G_2\}} \max_{\{D_z, D_x \in \mathcal{D}_x\}} V_Z(D_z, G_1) + V_X(D_x, G_2) + V_{L_2}(G_1, G_2)$$

调整网络。

记录生成网络和判别网络的权重和偏置。

迭代次数  $i$  增 1。

**步骤 4** 异常检测。

使用滑动窗口将测试样本  $X_{\text{Test}}$  划分为子序列, 滑动窗口大小为  $m$ , 则测试样本序列分解为  $X_{1 \sim m}, X_{m+1 \sim 2m}, \dots, X_{(k-1)m+1 \sim km}$ 。

真实测试样本  $X$  生成潜在空间:  $X_{\text{Test}}L = G_1(X_{1 \sim m}, X_{m+1 \sim 2m}, \dots, X_{(k-1)m+1 \sim km})$ 。

重构测试样本  $X$ :  $\hat{X}_{\text{Test}} = G_2(X_{\text{Test}}L)$ 。

计算重构损失  $RL_{\text{Test}} = E_{x \sim P_X} [\|X_{\text{Test}} - \hat{X}_{\text{Test}}\|_2]$ 。

计算判别损失  $DL_{\text{Test}} = D_Z(X_{\text{Test}}L)$ 。

计算异常分数  $ADScore = \alpha RL_{\text{Test}} + (1 - \alpha) DL_{\text{Test}}$ 。

依据异常分数输出异常飞行数据序列:  $A = \{A_1^{\text{seq}}, A_2^{\text{seq}}, \dots, A_m^{\text{seq}}\}$ 。

## 5.4 模型验证与结果分析

### 5.4.1 验证实验设计

#### 1. 实验内容

为了评估 LSTM-GAN 异常检测模型的性能,使用明尼苏达大学的公开数据集进行实验。首先通过网络损失值的变化验证目标损失函数设定的合理性;然后,进行异常检测算法性能对比实验,与常见的异常检测算法 K-Means、OC-SVM、LSTM 和 Auto-Encoder 进行性能比较。通过公开数据集,验证算法的合理性和有效性。

#### 2. 评估指标

为全面评估所提算法的异常检测性能,实验采用准确率 (Accuracy, Acc)、精确率 (Precision, Pre)、召回率 (Recall, Rec) 和  $F_1$  分数 4 个异常检测评估指标,计算公式分别如式(5-19)~式(5-22)所示。

$$\text{Acc} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}} \quad (5-19)$$

$$\text{Pre} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (5-20)$$

$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5-21)$$

$$F_1 = \frac{2 \times \text{Pre} \times \text{Rec}}{\text{Pre} + \text{Rec}} \quad (5-22)$$

对应混淆矩阵如表 5.1 所示。

表 5.1 混淆矩阵

| 检测结果 | 实际情况    |         |
|------|---------|---------|
|      | 正常样本    | 异常样本    |
| 正常样本 | 真正例数 TP | 假正例数 FP |
| 异常样本 | 假反例数 FN | 真反例数 TN |

其中,TP 表示真正例数,即实际情况和检测结果都是正常样本;FP 表示假正例数,即实际情况为异常样本而检测结果为正常样本;TN 表示真反例数,表示实际情况和检测结果都为异常样本;FN 表示假反例数,表示实际情况为正常样本而检测结果为异常样本。

准确率是指模型检测正确的样本个数占总样本的比例,是衡量模型对整体样本判断正