

TranAD (2022)

一、解决的问题

- 长时序依赖捕捉**：相比LSTM等递归模型，Transformer的并行化注意力机制能高效捕捉长期时序趋势，避免局部窗口限制（如USAD、MTAD-GAT），适合处理高维、长序列数据。
- 训练效率与稳定性**：Transformer的并行化训练结合进化损失函数，使训练时间较基线（如LSTM-NDT、GDN）减少75%-99%，同时避免对抗训练中的不稳定性。
- 异常检测与诊断一体化**：通过维度级异常分数计算（ s_i ）和POT动态阈值，既能判断是否异常（ $y=y_i$ ），又能定位异常来源维度，解决传统模型仅检测不诊断的局限。

二、创新点

1. 两阶段对抗训练：

第一阶段初步重构输入窗口，生成聚焦分数（重构误差）；第二阶段基于聚焦分数强化对异常区域的关注，通过对抗性学习放大误差，解决传统模型对微小异常不敏感的问题。

2. 进化损失函数：

随训练轮次（ n ）动态平衡重构损失与对抗损失（ $L_1 = \epsilon^{-n} \|O_1 - W\|_2 + (1 - \epsilon^{-n}) \|\hat{O}_2 - W\|_2$ ），初期侧重重构以保证稳定，后期增强对抗性以提升泛化能力。

3. 元学习（MAML）：

通过快速调整模型权重（ $\theta' = \theta - \alpha \nabla \theta L(f(\theta))$ ），使模型在有限数据下仍能高效学习时序趋势，解决联邦学习等场景中数据不足的问题。

三、模型架构

TranAD基于Transformer架构构建，核心是编码器-解码器结构，结合两阶段处理流程实现多元时序异常检测与诊断，模型结构图如下：

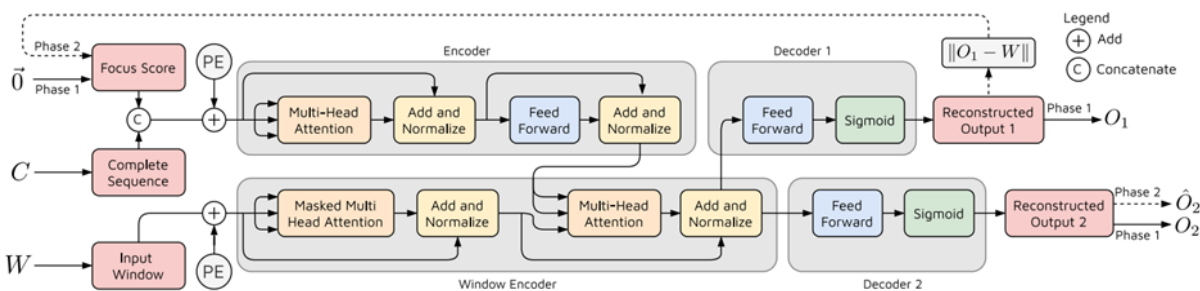


Figure 1: The TranAD Model.

具体架构介绍：

1. 编码器模块：包含两个编码器——完整序列编码器和窗口编码器。

- 完整序列编码器接收截至当前时间戳的完整序列（ C ）和初始聚焦分数（ F ），通过多头自注意力机制（含位置编码）捕捉全局时序趋势，输出编码后的全局特征（ I_1^2 ）。
- 窗口编码器接收局部滑动窗口（ W ）代码设置窗口大小为10，即10个时间点的测量，通过掩码多头自注意力（屏蔽未来时间步信息）结合全局特征（ I_1^2 ），生成局部窗口的编码表示（ I_2^3 ），兼顾局部上

下文与全局趋势。

$$W_t = \{x_{t-K+1}, \dots, x_t\}$$

2. **解码器模块**：包含两个解码器，基于窗口编码器的输出 (I_2^3) 生成重构结果。

- 第一阶段解码器输出初步重构 (O_1 、 O_2)，计算 O_1 和 W 重构误差作为“注意力分数”。
- 第二阶段解码器利用注意力分数调整注意力权重，生成强化重构 (\hat{O}_2)，放大异常区域的误差。

核心代码逻辑如下

```
1 def forward(self, src, tgt): # tgt:(1,128,38)
2     # Phase 1 - Without anomaly scores
3     c = torch.zeros_like(src) # c: (10,128,38) 初始条件c为零（无先验）
4     # 解码器1输出 $O_1$  # x1:(1,128,38)
5     x1 = self.fcn(self.transformer_decoder1(*self.encode(src, c, tgt)))
6     # Phase 2 - With anomaly scores
7     c = (x1 - src) ** 2 # c: (10,128,38) c更新为第一阶段重构误差
8     # 解码器2输出 $\hat{O}_2$  # x2:(1,128,38)
9     x2 = self.fcn(self.transformer_decoder2(*self.encode(src, c, tgt)))
10    return x1, x2
```

对抗性体现：两个解码器分别输出 O_1 和 \hat{O}_2 （对应代码中的 $x1$ 和 $x2$ ），后续通过损失函数（如文档中的 $L1$ 和 $L2$ ）形成对抗——解码器 1 最小化误差，解码器 2 放大误差。

$$\min_{\text{Decoder1}} \max_{\text{Decoder2}} \|\hat{O}_2 - W\|_2$$

$$L_1 = \epsilon^{-n} \|O_1 - W\|_2 + (1 - \epsilon^{-n}) \|\hat{O}_2 - W\|_2$$

$$L_2 = \epsilon^{-n} \|O_2 - W\|_2 - (1 - \epsilon^{-n}) \|\hat{O}_2 - W\|_2$$

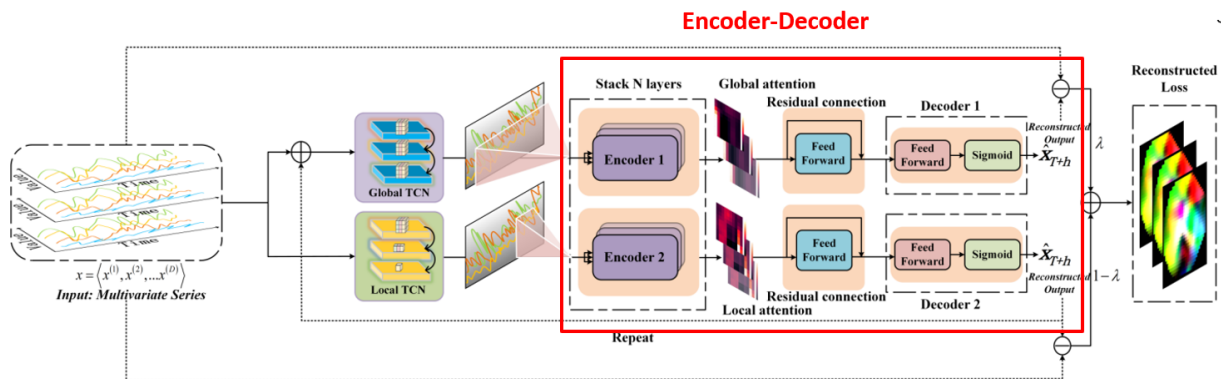
代码通过一个 $L1$ 融合了文档中提及的 $L1$ 和 $L2$ 。

```
1 # 损失计算：动态权重组合两阶段损失 ( $L_1 = \epsilon^{-n} \|O_1 - W\|_2 + (1 - \epsilon^{-n}) \|\hat{O}_2 - W\|_2$ ) z[0]->  $O_1$  z[1]->  $\hat{O}_2$ 
2 l1 = l(z, elem) if not isinstance(z, tuple) else (1 / n) * l(z[0], elem) + (1 - 1 / n) * l(z[1], elem) # l1:{1,128,38}
```

3. **核心机制**：通过多头自注意力 (`MultiHeadAtt`) 捕捉特征间依赖，位置编码保留时序信息，掩码机制确保训练时不泄露未来数据。

DTAAD(2024)

模型架构



双TCN结构，分别对输入数据采用因果卷积和扩张卷积进行特征提取后

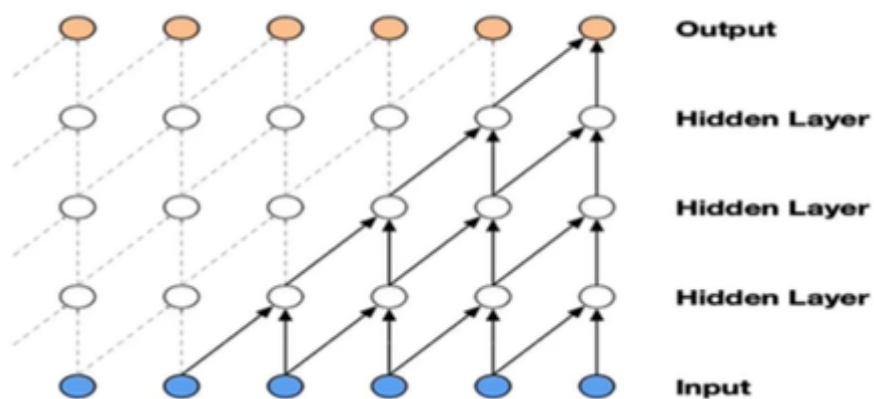
分别进入Transformer的编码层和解码层（AE） 重构出误差，再综合两个AE的重构误差得到最终的异常分数。

单层卷积必须满足的条件：1、输入长度=输出长度 2、当前时刻仅依赖于过去时刻，而不能看到未来信息(因果)。

因果卷积 (local TCN)

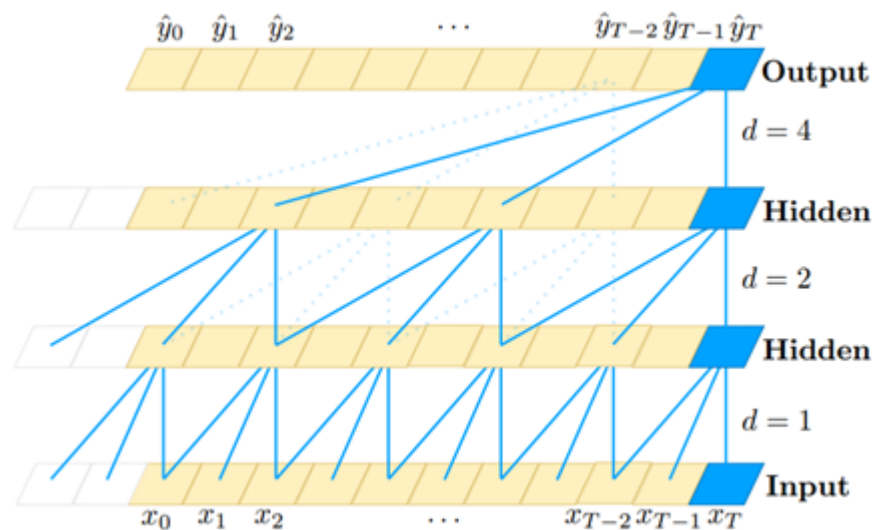
因果卷积通过堆叠多个卷积层扩大感受野，对长序列存在梯度消失，所以仅用于局部TCN

流程： conv1d -> Weight -> LeakyRelu -> Dropout



扩张卷积 (Global TCN)

跳跃式的扩大感受野，堆叠少量的卷积层能使感受野呈现指数级增长。



OmniAnomaly(2019)

一、解决的问题

OmniAnomaly主要针对多变量时间序列异常检测中的核心挑战，具体包括：

1. 复杂的时序依赖性与随机性

工业设备（如服务器、航天器）的多变量时间序列同时存在强时序依赖（如历史状态影响当前状态）和随机性（如外部环境干扰），传统方法难以同时建模这两种特性。

2. 实体级异常检测需求

实际场景中更关注设备整体异常（实体级）而非单个指标异常（metric级），但现有方法多针对单变量或忽略指标间关联。

3. 异常解释性不足

检测到异常后，难以定位关键影响指标，阻碍故障排查。

4. 无监督场景适应性

工业数据中异常标签稀缺，需无需标注的无监督方法。

二、创新点

1. stochastic recurrent neural network架构

融合GRU（捕捉时序依赖）与VAE（引入随机性），首次在latent空间中显式建模stochastic变量的时序依赖，通过**stochastic变量连接技术**（Linear Gaussian State Space Model）让当前stochastic变量依赖历史变量，提升表示学习能力。

2. 非高斯分布建模

采用**planar normalizing flows**，通过可逆映射将高斯分布转化为复杂非高斯分布，更贴合真实数据的分布特性。

3. 异常解释机制（这个还未理解）

基于重构概率分解，将实体级异常拆解为各单变量指标的贡献度，通过排序关键指标实现可解释性（HitRate@150%达0.89）。

4. 自适应阈值选择

结合极值理论（EVT）的POT（Peaks-Over-Threshold）方法，自动确定异常阈值，无需人工调参。

三、模型架构

OmniAnomaly整体分为离线训练和在线检测两阶段，核心架构包括**qnet（推理网络）**和**pnet（生成网络）**：

1. qnet（推理网络）

- 输入：多变量时间序列片段 ($x_{t-T:t}$)
- 流程：

1. 用GRU处理输入，生成确定性隐藏变量 (e_t)，捕捉时序依赖；

2. 拼接 (e_t) 与历史 stochastic变量 (z_{t-1}), 通过 dense层生成当前 stochastic变量 (z_t) 的均值和标准差;
3. 经planar normalizing flows转换, 得到非高斯的 (z_t), 提升表示能力。

2. pnet (生成网络)

- 输入: stochastic变量 (z_t)
- 流程:
 1. 用GRU处理 (z_t), 生成确定性隐藏变量 (d_t);
 2. 通过 dense层生成重构数据 (x'_t) 的均值和标准差, 假设 (x'_t) 服从高斯分布;

利用 *LinearGaussianStateSpaceModel* 建模 z_t 与 z_{t-1} 的依赖关系, 强化时序一致性。

3. 训练目标

最大化证据下界 (ELBO), 包含:

- 重构损失: 衡量 (x_t) 与 (x'_t) 的差异;
- KL散度: 正则化 (z_t) 的分布与先验分布的差异。

4. 异常检测

- 在线阶段输入新序列, 计算重构概率 作为异常分数;

$$(\log(p_\theta(x_t | z_{t-T:t})))$$

- 低于POT方法确定的阈值则标记为异常, 并通过各维度重构概率排序解释异常原因。