Journal of Image and Signal Processing 图像与信号处理, 2024, 13(2), 151-162
Published Online April 2024 in Hans. https://www.hanspub.org/journal/jisp
https://doi.org/10.12677/jisp.2024.132014

Hans汉斯

# 基于Transformer的时间序列插补技术研究

谷小兵[1]，牛少彰[1,2]，王茂森[2]，安洪旭[2]，史成洁[3]

[1]北京邮电大学计算机学院国家示范性软件学院，北京
[2]东南数字经济发展研究院，浙江 衢州
[3]中国科学院信息工程研究所，北京

收稿日期：2024年3月11日；录用日期：2024年4月1日；发布日期：2024年4月9日

## 摘 要

本文旨在解决多元时间序列数据中的缺失值插补问题，提升时间序列数据插补的效果。时间序列数据是反映随时间变化的随机变量的结果，在物联网应用中得到广泛应用。然而，数据缺失问题是时间序列处理中的一个重要挑战，因为大多数下游算法需要完整的数据进行训练。本文通过总结以往时间序列建模过程中采用的插补方法，改进了一种基于Transformer模型的插补模型，并在多个数据集中验证了本文中插补模型的效果。通过本文的研究，可提高时间序列预测的准确性和实用性，对于物联网应用和其他领域中的时间序列分析具有一定的实用价值。

## 关键词

时间序列，多元时间序列，缺失值插补，Transformer 模型，时间序列建模，数据完整性，自注意力，神经网络

# Research on Transformer-Based Time Series Imputation Technique

**Xiaobing Gu[1], Shaozhang Niu[1,2], Maosen Wang[2], Hongxu An[2], Chengjie Shi[3]**

[1]School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Tele-Communications, Beijing
[2]Southeast Digital Economy Development Research Institute, Quzhou Zhejiang
[3]Institute of Information Engineering, Chinese Academy of Sciences, Beijing

Received: Mar. 11th, 2024; accepted: Apr. 1st, 2024; published: Apr. 9th, 2024

## Abstract

**This article aims to address the issue of missing value imputation in multivariate time series data**

文章引用: 谷小兵, 牛少彰, 王茂森, 安洪旭, 史成洁. 基于 Transformer 的时间序列插补技术研究[J]. 图像与信号处理, 2024, 13(2): 151-162. DOI: 10.12677/jisp.2024.132014

**to enhance the effectiveness of imputation. Time series data, widely utilized in Internet of Things (IoT) applications, reflects the outcomes of random variables changing over time. However, data missingness poses a significant challenge in time series processing, as most downstream algorithms require complete data for training. By summarizing past imputation methods used in time series modeling and improving a Transformer-based imputation model, this paper validates the effectiveness of the proposed imputation model across multiple datasets. The research presented in this paper can improve the accuracy and practicality of time series prediction, providing practical value for time series analysis in IoT applications and other domains.**

## Keywords

**Time Series, Multivariate Time Series, Missing Value Imputation, Transformer Model, Time Series Modeling, Data Completeness, Self-Attention, Neural Network**



## 1. 引言

在当代物联网技术的快速发展背景下，大规模多元时间序列数据的生成与应用已成为一种普遍现象。然而，数据缺失问题普遍存在，这不仅对数据质量构成了严重威胁，也对后续的数据分析准确性带来了显著影响。传统的数据插补技术在处理现代数据结构的复杂性方面存在局限性，这在一定程度上制约了相关研究和应用的深入发展。

针对这一挑战，本文提出了一种基于深度学习的先进时间序列数据插补方法。该方法利用深度神经网络的强大学习能力，旨在有效地解决时间序列数据中的缺失值问题，以期提升数据处理技术的效能，并为相关领域的科学研究和实际应用提供更为精确和可靠的数据支持。

## 2. 研究背景和意义

时间序列数据是一种按照时间顺序排列的随机变量序列，通常代表了在等间隔的时间点上对某一潜在过程进行的连续观测。这些数据序列不仅捕捉了随机变量随时间的演变趋势，而且其内在的统计依赖性揭示了客观世界及其动态变化的深层信息。

时间序列中的每个数据点，无论是其在序列中的位置还是其数值大小，都蕴含着对过去和未来趋势的预测价值，从而使得时间序列分析成为理解和预测复杂系统行为的重要工具。通过对时间序列数据的结构和模式的深入研究，可以揭示数据生成过程中的内在规律。

随着 5G、云计算、快速增长的传感器制造技术等新兴技术的出现，物联网应用越来越普及。种类繁多的物联网应用产生了大量的时间序列数据。工业状态监控应用中，设备通常会被安装不同的传感器，对设备的温度，用电量等状态数据进行监控，在此过程中会产生大量的状态数据，这些状态数据就是一种时间序列。

时间序列分析的核心目的在于探究长期变动中的统计规律，从而理解动态系统，预测将来事件，甚至通过干预控制未来发生。如，在工业场景下，对传感器数据的分析能够实现状态监控、异常检测、故障预测以及预测性维护。尽管时间序列数据蕴含丰富信息，但这些信息往往隐藏在复杂的数据结构中，需要细致的处理和分析才能提取出来。时间序列分析面临的挑战包括无序时间戳、缺失值、异常值和数

据噪声等问题。其中，缺失值的处理尤其困难，因为传统方法往往无法适用于时间序列数据的特性。

大量的时间序列数据在企业扩充产线、扩大规模的过程中被生产出来，这些数据包含产线生产过程中的海量信息，具有一定时效性。如何在短期内对实时产生的大量时间序列数据进行处理，获取有价值的数据并使其能够参与下游任务，比如时间序列预测、异常检测等工作，这些工作中的第一道关卡就是数据处理，而数据处理中的缺失数据处理是本文研究的重点。

在多元时间序列数据的采集中，数据缺失是常见现象。比如在金融、医疗还是交通等领域，因传感器故障、环境不稳定或隐私保护等因素，数据集往往包含大量缺失值，这对下游分析和决策的准确性造成影响。在工业环境下，数据的采集也可能因电力或传感器故障等问题而受阻，且由于数据量庞大、维度高，处理缺失值的难度更加显著。

对缺失数据进行高质量的插补对下游任务影响深远，低质量的插补可能导致分析偏差，降低预测效果。因此，本文总结了以往时间序列建模过程中的插补方法，并提出了一种基于 Transformer 的深度学习插补算法。通过在多个数据集上的对比分析，验证了该模型的有效性，对工业时序数据分析具有重要的实践价值。

## 3. 相关工作

时间序列中缺失值插补分为传统的基于统计和基于机器学习的方法，以及近年来兴起的基于深度学习的方法。

基于深度学习的时间序列插补方法可以根据使用的技术大致分为两类：

1) 基于 RNN 的方法，如 BRITS [1]，GRU-D [2]，M-RNN [3]等，使用单向或多向的 RNN 进行建模。GRU-D [2]的作者 Che 等人提出了一个 GRU 的变体模型，并引入了 masking 矩阵信息；M-RNN [3]与 BRITS [1]则根据双向 RNN 的隐藏状态来插补数据，但是这些算法受到不同的训练限制，它们中多数会在不同的缺失场景以及缺失率下表现出较优的性能；

2) 基于生成模型的方法，包含 GAN 以及 VAE 的方法。GAN 的算法比如 GRUI [4]的作者使用 GRUI 与 GAN 的组合方式；E2GAN [5]则在 GRUI 模型的基础上增加了自动编码器(Auto-Encoder)结构在其生成器和判别器阶段。Liu 等人提出了一个名为 NAOMI [6]的非自回归模型，用于时序数据的插补，GAN 生成器和判别器采用的是双向 RNN 的模型。SS-GAN [7]在传统 GAN 模型基础上增加了一个分类器的结构，考虑了下游任务对插补效果的影响。但是这类模型呈现出训练不稳定的特点，难以达到最高水平的表现。基于 VAE 的模型比如 GP-VAE [8]，将非线性降维与表达丰富的时间序列模型相结合，SGP-VAE [9]作者提出了一种基于上下文无关文法解析的变分自动编码器。它直接对这些解析树进行编码和解码，确保生成的输出始终有效。

时间序列拥有自相关的前后依赖关系，多元的时间序列处理就需要从时间的前后联系与各变量之间关系两个维度去捕捉信息。传统的统计方法利用自相关函数建模提取时序信息，但是对于变量之间的联系捉襟见肘，而且对于时序信息的提取能力有限，比如自回归模型，受限于自身的复杂程度所观察的数据窗口大小非常有限。基于 RNN 的特征提取方法拓展了窗口大小，但是 RNN 依然不能解决序列的长期依赖问题。在 RNN 的基础上，LSTM 精心设计了“门”结构，在长序列上能够捕捉序列的长期依赖信息，但是 LSTM 依然受限于 RNN 的循环网络结构，无法实现并行计算。

Transformer 是一个面向 sequence to sequence 任务的模型，在 17 年的论文《Attention is all you need [10]》中首次提出。在《Transformers in Time Series: A Survey [11]》一文中肯定了 Transformer 在统一建模方面表现的巨大威力，并展示了将 Transformer 应用于时序领域的部分成果。Transformer 是第一个完全依赖自注意力(self-attention)来计算输入和输出的表示，而不使用序列对齐的递归神经网络或卷积神经网络的

转换模型。正是 Transformer 这种不同于 RNN 的循环网络的结构，使得 Transformer 可以实现并行计算，大大加快了数据的处理过程，突破了 RNN 与 LSTM 模型不能并行计算的限制。并且自注意力机制可以产生更具可解释性的模型，可以从模型中检查注意力分布。各个注意头(attention head)可以学会执行不同的任务。

将 Transformer 作为特征提取层，我们可以从变量之间和时间两个尺度设计网络，作为下一步插补模型的输入，同时得益于 Transformer 网络的并行计算优势，可以加快模型的运行速度。

已有多篇文章将 Transformer 应用于时间序列插补领域，比如 CDSA [12]采用时间、位置、多维度三个领域联合推算缺失数据；DeepMVI [13]模型采取了将沿时间序列的细粒度和粗粒度模式以及跨类别维的相关序列的趋势进行组合的策略，包括具有新颖卷积窗口功能的时变子，以及具有学习性嵌入的核回归对时序数据进行了插补。SAITS [14]则在损失函数上作出改进提出了重建损失与插补损失结合的联合任务训练方法。

我们注意到这些利用 Transformer 来作为插补任务基础模型的文章都利用了 Transformer 多头注意力机制在提取时间维度特征方面的优势，在面对工业领域大量维度复杂数据 FFN 的机制便不足以支持挖掘多维度之间的深度关系，为此我们参考了 Itransformer [15]的思想，将组合维度关系的职责交予多头注意力层，而原有的 FFN 进行改造提取时序特征。

## 4. TransImputeNet

### 4.1. 含有缺失值的多变量时间序列定义

设有 $n$ 个变量 $\{X_1, X_2, \cdots, X_n\}$ 的多维时间序列数据，该时间序列数据拥有 Time 与 Variate 两个维度，每个变量均为长度为 $T$ 的时间序列 $\left\{x_t^1, x_t^2, \cdots, x_t^n\right\}$，$t = 1,2,\cdots,T$。存在若干个时间点的某些变量值缺失。可以表示如下：

$$X = \left\{x_j^i\right\}, i = 1,2,\cdots,n;\ j = 1,2,\cdots,T \tag{1}$$

其中，$x_j^i$ 表示第 $i$ 个变量在第 $j$ 个时间点的值，我们使用 Mask 矩阵 $M$ 来表示数据缺失的位置：

$$M_j^i = \begin{cases} 1 & X_j^i \text{ 为真实值} \\ 0 & X_j^i \text{ 为缺失值} \end{cases} \tag{2}$$

### 4.2. 数据预处理

在上一节含有缺失值的多变量时间序列中，缺失位置的信息往往缺少对照，缺少真实信息意味着模型无法判断插补效果的好坏，因此我们在原始的时序数据集上做了 mask 处理以获取含有对照真实值的训练数据。

我们将标准化后的数据按照(2)计算得到 Real Mask 矩阵，用于表示 mask 处理前数据的缺失位置。此处的 mask 操作为完全随机缺失情况的模拟，按照一定比例掩去部分真实值。

标准化后的数据 $X$ 作 mask 处理，掩去一部分真实数据用于对照，再按照(2)式计算得到 Missing Mask 矩阵，用于表示 mask 处理后数据的缺失位置。

### 4.3. 模型架构

如图 1 所示本文所改进模型在原始 Transformer 模型基础上作了比较大的改动，舍弃了原始 Transformer 模型结构中对称的解码器，使用 Linear 线性层作为解码器，以输出多变量。

模式输入阶段的结构如图 1 所示，为了尽可能地的提高插补质量，除了网络组件的改动，

TransImputeNet 在模型结构上也做了很大改动，采用两个编码器－解码器结构串联的方法，分两个阶段输出插补结果并根据两个阶段的结果计算损失函数，降低单一结构可能出现的插补误差。

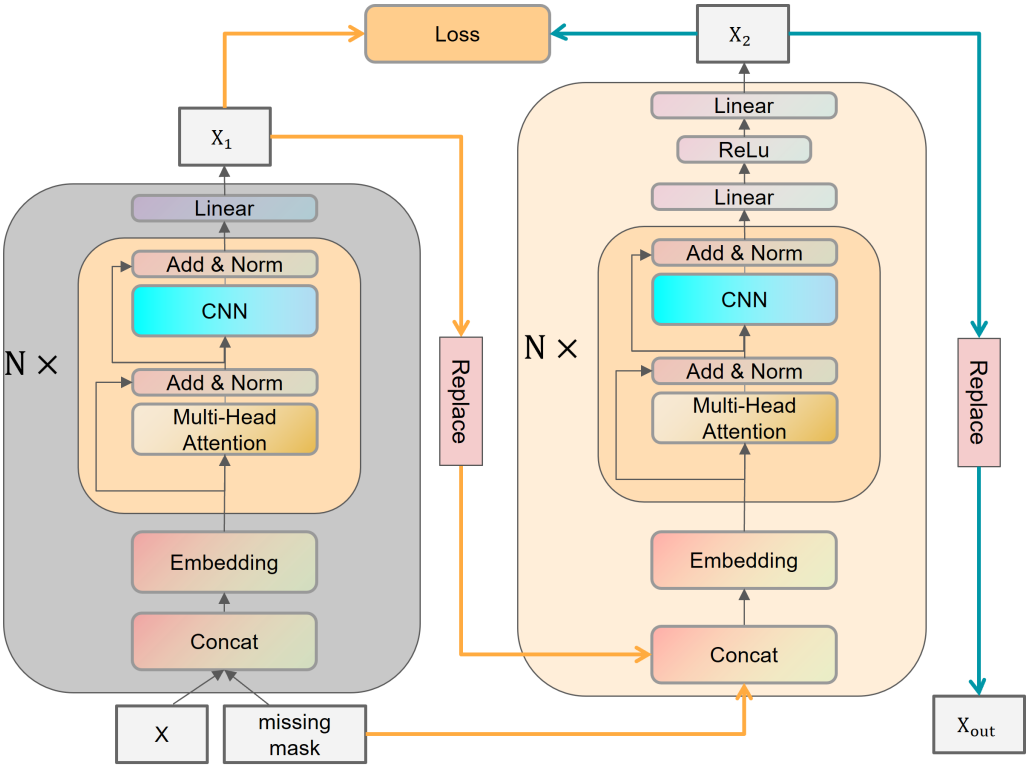


**Figure 1.** TransInputeNet model structure diagram
**图 1.** TransImputeNet 模型结构图

### 4.3.1. Embedding 层

Embedding层即数据嵌入层将输入的时间序列数据嵌入到注意力层需要的维度空间之中，与经典的Transformer模型不同，我们舍弃了传统的Positional Encoding (位置编码)层，直接在Embedding层引入位置信息，将4.2中得到的mask矩阵missing mask与原始数据$X$在Variate维度作连接处理，并将Variate维度作为注意力层的输入参与下一阶段的运算，最终得到的嵌入后的数据$E$公式如下：

$$\text{concat\_}X = \text{Concat}(X, M) \tag{3}$$

Concat为连接操作，将$X$的缺失位置信息$M$合并到$X$，用于提示模型忽略缺失位置的信息，重点关注现有的真实值。

$$E = \text{Dropout}\left(\text{Embedding}\left(\text{concat\_}X^{*}\right)\right) \tag{4}$$

$\text{concat_}X^{*}$为$X$将Time与Variate两个维度交换之后的结果。

Embedding层得到的结果$E$输入至多个Attention-CNN组合块中，简称为ACB，此为原始Transformer中变化而来的编码器结构。

### 4.3.2. Attention 层

在多维时序数据中，Attention层用于捕捉Variate维度之间的关系。该层引入了多头注意力机制，允许模型同时关注输入序列中的不同部分，从而提高了模型的学习能力。以下是Attention层的关键步骤：首

先，计算查询矩阵$Q$与相应键$K$的点积得到注意力分数：

$$Q = EW_Q, K = EW_K, V = EW_V \tag{5}$$

$$A_{Q,K}^{i} = \text{Softmax}\left(Q^i \left(K^i\right)^{\text{T}}\right) \tag{6}$$

通过对注意力分数进行Softmax操作，得到Variate维度上的注意力权重($A$)。这些权重用于加权求和值，以获得在Variate维度上考虑了关系的最终输出。最后，通过线性层，得到了表示多维关系的最终输出：

$$head_i = A_{Q,K}^{i} V \tag{7}$$

$$A_{Q,K,V} = C\left(head_1, head_2, \cdots, head_h W\right) \tag{8}$$

注意力机制的引入旨在使模型更好地理解多维时序数据中 Variate 维度之间的关联性，提高时间序列结构的建模能力。

#### 4.3.3. CNN 层

为了更全面地捕获时序数据的局部特征，我们引入了卷积神经网络(CNN)层。首先，我们对输入的时序特征进行了一维卷积操作，其中输入张量通过卷积核的运算得到了高维的中间表示。随后，通过激活函数的非线性变换，我们引入了非线性特征。

$$X_{\text{LN}} = \text{LN}\left(X + \text{Dropout}\left(A_{Q,K,V}\right)\right) \tag{9}$$

$$X_{\text{con1v\_1}} = \text{Dropout}\left(\text{ReLu}\left(\text{Conv1D}\left(X_{\text{LN}}\right)\right)\right) \tag{10}$$

由于卷积操作的特性，模型能够有效地捕获输入序列中的局部模式。接着，我们使用卷积层进行降维，通过线性投影将高维特征映射回原有维度。最终的输出表示通过残差连接和层标准化得到，以确保模型的稳定性和更好的训练效果。CNN层的引入旨在强化模型对时序数据局部特征的建模能力，从而提高整体性能。我们使用CNN网络替换原始Transformer原始的前馈全连接网络层。

$$X_{\text{con1v\_2}} = \text{Dropout}\left(\text{Conv1D}\left(X_{\text{con1v\_1}}\right)\right) \tag{11}$$

$$X_{\text{out}} = \text{LN}\left(X_{\text{LN}} + X_{\text{con1v\_2}}\right) \tag{12}$$

#### 4.3.4. Attention-CNN Block

ACB是模型架构中的关键组件，负责提取时序数据的关键特征。以下ACB由单层Attention层和CNN层串联而成，整体模型在经由Embedding层后由多个ACB组成：

$$z = \left[\text{Conv1D}\left(\text{MultiHeadAttention}\left(X_{m-1}\right)\right)\right]^N \tag{13}$$

$$X_m = zW_z + b_z \tag{14}$$

通过注意力和CNN机制的融合，ACB被赋予了识别和利用关键时间模式的能力，从而提高了其性能和泛化能力。

#### 4.3.5. 两阶段插补

为了与损失函数结合增强训练的鲁棒性，提高结果的泛化性，采用了分阶段的串联模式，将第一阶段的结果作为下一个阶段的输入。

第一阶段插补在Embedding层之后，接受多个级联的ACB作为输入，并通过线性层作为解码层来生成第一阶段的输出。具体地，首先进行线性变换：

$$X_1 = zW_z + b_z \tag{15}$$

其中，$z$表示多级ACB的输入特征向量，$W_z$和$b_z$分别表示线性层的权重和偏置。

然后，利用插值操作融合原始输入$X$和第一阶段的输出$X_1$：

$$X_m = M \odot X + (1-M) \odot X_1 \tag{16}$$

这里，$M$是一个二元掩码矩阵，代表Missing Mask，用于表示mask处理后数据的缺失位置，$X_m$表示经过第一阶段插补后的特征表示，(16)式即Replace操作，将结果中不需要插补的部分还原成真实值。

第二阶段插补的输入来自于第一阶段插补的输出，同时在第一阶段插补的基础上增加了ReLU函数的非线性变换。具体地：

$$X_2 = \mathrm{ReLu}(\alpha W_\alpha + b_\alpha) W_\beta + b_\beta \tag{17}$$

其中，$\alpha$是ReLU函数的参数，$W_\alpha$，$b_\alpha$，$W_\beta$和$b_\beta$分别是与非线性变换相关的权重和偏置。

最后，再次利用插值操作融合原始输入$X$和第二阶段的输出$X_2$：

$$X_{\text{out}} = M \odot X + (1-M) \odot X_2 \tag{18}$$

这里，$X_{\text{out}}$表示经过两阶段插补后的最终特征表示。

通过两阶段插补的设计，模型能够在保留原始特征的基础上，利用ACB提取的特征进行补充和调整，从而增强了模型对时间序列数据的建模能力。

### 4.4. 损失函数

本文中损失函数涉及两方面，参考SAITS作者采取的插补损失和重建损失两种损失函数结合的策略，我们在其基础上进行了改进，维持了插补损失，将重建损失改为两个阶段的结果的结合，简化了计算过程：

$$L_1\left(X, \hat{X}\right) = \sqrt{\frac{1}{|R-M|} \sum_{i=1}^{I} \sum_{j=1}^{J} \left(X_{ij} - \hat{X}_{ij}\right)^2 \cdot (R-M)_{ij}} \tag{19}$$

$L_1$为插补损失函数，只计算mask过程中掩去数据真实值与插补后值的均方根损失，其中$X$代表原始数据，$\hat{X}$代表插补后的数据，$R$代表数据预处理时得到的Real Mask，$M$代表Missing Mask，只计算mask过程中预处理造成的缺失位置，因为这些位置都有真实值对照。

$$L_2\left(X, \hat{X}\right) = \sqrt{\frac{1}{|M|} \sum_{i=1}^{I} \sum_{j=1}^{J} \left(X_{ij} - \hat{X}_{ij}\right)^2 \cdot M_{ij}} \tag{20}$$

$L_2$为重建任务损失，与$L_1$不同，它只计算mask操作前后均未缺失的数据变化，用于表示模型学习整体时序数据模式的能力。

$$L = L_1\left(X, X_{\text{out}}\right) + \lambda \frac{L_2\left(X, X_1\right) + L_2\left(X, X_2\right)}{2} \tag{21}$$

$L$即最终的损失函数，$\lambda$为重建损失的权重占比，这里我们取$\lambda$为1，既兼顾了模型学习整体时序数据的能力，又能提高泛化性，这一损失函数的设计旨在平衡模型对数据缺失的插补能力和对整体时序数据模式的学习能力。

## 5. 实验结果

为了本文中改进过的自注意力模型，实验在来自不同领域的两个公共数据集和一个由我们实时采集的工业数据集上进行：Beijing Multi-Site Air-Quality，PhysioNet 2019 Mortality Prediction Challenge，造纸

工业流水线数据集。

Beijing Multi-Site Air-Quality (Air-Quality)：数据集包括北京12个监测站的小时空气污染物数据。数据收集时间为2013年3月1日至2017年2月28日(共48个月)。对于每个监测站，有11个连续的时间序列变量被测量(例如PM2.5、PM10、$SO_2$)。我们将12个站点的变量聚合在一起，因此该数据集有132个特征。该数据集中总共有1.6%的缺失值。为生成时间序列数据样本，我们选择每24小时的数据，即每24个连续步骤作为一个样本。与数据集PhysioNet-2012类似，我们按照采样站点分层，采取随机划分法训练集，测试集，验证集，与PhysioNet-2019数据集比例相同。

PhysioNet 2019 Mortality Prediction Challenge (PhysioNet-2019)：数据集包含来自ICU (重症监护室)患者的40,000个多变量临床时间序列样本。根据患者的状态，最多有39个时间序列变量被测量，例如温度、心率、血压。测量可能以定期间隔(每小时或每天)收集，并且也可能以不规则间隔(仅按需要收集)记录。并非所有样本都有所有变量可用。该数据集非常稀疏，缺失最多的患者状态缺失率甚至超过了90%。我们首先提取了所有记录超过48个小时患者样本并将此作为训练集，根据80%和20%将数据集分为训练集和测试集。然后，从训练集中分割出25%的样本作为验证集。我们随机删除了验证集和测试集中一定比例的观察值，并使用这些值作为基准来评估模型的插补性能。

造纸工业流水线数据集取自某地造纸工业流水线2022年5月7日一天的产线传感器记录，包含85,032个实时监控数据，每秒钟做一次记录，有高达1500个变量记录，我们筛选了其中的数值型变量427个，并按照60个为一个时间单位作分割，同样以3:1:1的比例划分训练集、测试集、验证集。

用于评估方法的插补性能的指标：RMSE (均平方根误差)。以下是评估指标的数学定义。请注意，仅在方程式的输入中由掩码指示的值上计算误差。

$$\mathrm{RMSE}\left(\text{estimation, target, mask}\right)=\sqrt{\frac{\sum_{d}^{D}\sum_{t}^{T}\left(\left(\left(\text{estimation-target}\right)\odot\text{mask}\right)^{2}\right)_{t}^{d}}{\sum_{d}^{D}\sum_{t}^{T}m_{\text{task}}^{d}}} \tag{22}$$

estimation：表示估计的值或插补值，target：表示目标值，即真实值，mask：掩码，用于指示哪些值应该被考虑在内，$D$：表示数据的维度，$T$：表示时间步数。RMSE表示了估计值与目标值之间的平均平方根误差，其中只考虑了由掩码指示的值。

实验表格中黄色部分代表当前缺失率下最好的结果。

模型的性能随着缺失率的增加而逐渐变差。当缺失率从0.1增加到0.9时，模型的指标值都呈现出增加的趋势，随着数据缺失程度的增加，模型的插补性能普遍下降。

LOCF是Last Observation Carries Forward，就是用前一个记录值填补现在的缺失值，其他模型均已在前文中提及，TransImputeNet是本文中的模型。本次选用的模型均具有一定代表性，通过实验对比能反映TransImputeNet在数据插补上的具体效果如何。

根据实验结果表格，我们可以观察到不同缺失率下各个模型的性能。在表格中，每个模型在不同缺失率下的表现由相应的RMSE指标值表示，指标值越低代表模型性能越好。

如表1显示，加粗的数据表示插补效果最好，在Air-Quality数据集上多个模型进行了插补实验，结果显示，Air-Quality数据集上缺失率低于0.8时，TransImputeNet 模型的性能是最佳的，这表明TransImputeNet在处理缺失值时表现出了很强的能力，但是在Air-Quality数据集上超过约80%缺失率时性能较其他模型有所下降，这表明TransImputeNet模型在极高缺失率下对时序数据的插补效果相对劣势。以插补实验中效果较好的SAITS模型作为基准，TransImputeNet相比SAITS，在Air-Quality数据集上的插补效果平均提升了1.73%，在缺失率为0.2的情况下提升最大约为7.73%，在缺失率为0.9的情况下性能降低了约16.97%。

**Table 1.** Performance comparison table of various models in the Air Quality dataset
**表 1.** Air-Quality 数据集下各模型的性能比较表

| Air-Quality 数据集缺失率 | LOCF | GP-VAE | BRITS | SS-GAN | SAITS | TransImputeNet |
|---|---|---|---|---|---|---|
| 0.1 | 0.5130 | 0.4383 | 0.3716 | 0.3684 | 0.3414 | **0.3150** |
| 0.2 | 0.5387 | 0.4682 | 0.3751 | 0.3881 | 0.3479 | **0.3209** |
| 0.3 | 0.5664 | 0.4801 | 0.3930 | 0.3972 | 0.3605 | **0.3343** |
| 0.4 | 0.6001 | 0.5126 | 0.4301 | 0.4348 | 0.3921 | **0.3708** |
| 0.5 | 0.6763 | 0.5424 | 0.4574 | 0.4617 | 0.4146 | **0.3959** |
| 0.6 | 0.6804 | 0.5846 | 0.4958 | 0.5101 | 0.4539 | **0.4385** |
| 0.7 | 0.7537 | 0.6227 | 0.5173 | 0.5410 | 0.4689 | **0.4667** |
| 0.8 | 0.7916 | 0.6718 | 0.5817 | 0.6048 | **0.5238** | 0.5449 |
| 0.9 | 0.9010 | 0.7566 | 0.6736 | 0.7047 | **0.5941** | 0.6949 |

表2显示了多种模型在PhysioNet-2019数据集上的插补效果，不同于Air-Quality数据集，本次实验中TransImputeNet模型的插补效果略强于目前实验中效果最好的SAITS模型，在PhysioNet-2019数据集上的插补效果相对于SAITS，最大提升发生在缺失率为0.3时，提升了5.19%，最小提升发生在缺失率为0.9时，提升了0.18%，平均提高了2.53%。

**Table 2.** Performance comparison table of various models in the PhysioNet-2019 dataset
**表 2.** PhysioNet-2019 数据集下各模型的性能比较表

| PhysioNet-2019 数据集缺失率 | LOCF | GP-VAE | BRITS | SS-GAN | SAITS | TransImputeNet |
|---|---|---|---|---|---|---|
| 0.1 | 0.5983 | 0.5728 | 0.5639 | 0.5962 | 0.4617 | **0.4469** |
| 0.2 | 0.6092 | 0.6076 | 0.5746 | 0.6122 | 0.4850 | **0.4616** |
| 0.3 | 0.6359 | 0.6035 | 0.5874 | 0.6417 | 0.5050 | **0.4788** |
| 0.4 | 0.6507 | 0.6496 | 0.6121 | 0.6439 | 0.5150 | **0.4957** |
| 0.5 | 0.6551 | 0.6542 | 0.6291 | 0.6631 | 0.5320 | **0.5290** |
| 0.6 | 0.6890 | 0.6912 | 0.6503 | 0.6865 | 0.5573 | **0.5419** |
| 0.7 | 0.7143 | 0.7168 | 0.6904 | 0.7197 | 0.5778 | **0.5730** |
| 0.8 | 0.7565 | 0.7574 | 0.7087 | 0.8150 | 0.6133 | **0.6041** |
| 0.9 | 0.8270 | 0.9620 | 0.6806 | 0.8343 | 0.6663 | **0.6651** |

表3展示了多个模型在造纸行业流水线数据集上的插补效果，与前两个数据集不同，本研究中模型在此次实验的效果较SAITS有了更高的提升，而且在实验所用全部缺失率情况下均领先于其他模型。在造纸行业流水线数据集上的插补效果相比SAITS，平均提升了约11.86%，最大提升发生在缺失率0.9时，提升了约15.11%，最小提升发生在缺失率0.1时，提升了约6.60%.

Air-Quality数据集上缺失率低于0.8时，TransImputeNet模型的性能是最佳的，这表明本研究中的TransImputeNet模型在处理缺失值时表现出了很强的能力，但是在Air-Quality数据集上超过约80%缺失率时性能较其他模型有所下降。这表明TransImputeNet模型在极高缺失率下对数据的应用不够完美，但是效果依然可以接受。

**Table 3.** Comparison table of performance of various models in the paper industry dataset
**表 3.** 造纸行业数据集下各模型的性能比较表

| 造纸行业数据集缺失率 | LOCF | GP-VAE | BRITS | SS-GAN | SAITS | TransImputeNet |
|---|---|---|---|---|---|---|
| 0.1 | 0.5043 | 0.5347 | 0.4213 | 0.4047 | 0.4075 | **0.3806** |
| 0.2 | 0.5094 | 0.5070 | 0.4340 | 0.3886 | 0.4229 | **0.3777** |
| 0.3 | 0.5103 | 0.5588 | 0.4422 | 0.3972 | 0.4296 | **0.3843** |
| 0.4 | 0.5139 | 0.5044 | 0.4511 | 0.4348 | 0.4412 | **0.3860** |
| 0.5 | 0.5181 | 0.5045 | 0.4636 | 0.4632 | 0.4508 | **0.4000** |
| 0.6 | 0.5278 | 0.5062 | 0.4743 | 0.5098 | 0.4613 | **0.3939** |
| 0.7 | 0.5447 | 0.5424 | 0.4968 | 0.5409 | 0.4712 | **0.4080** |
| 0.8 | 0.5708 | 0.5237 | 0.5299 | 0.6048 | 0.4905 | **0.4318** |
| 0.9 | 0.6287 | 0.5558 | 0.6058 | 0.7024 | 0.5216 | **0.4428** |

因为工业场景中出现这种高缺失率的情况是比较少的，通过与历史数据的结合，可以人为调整缺失率，使之维持在一定范围之内，所以TransImputeNet在高缺失率下可能出现的效果下降对插补质量的影响不大；当缺失率过高时，如果数据依然拥有插补的必要，考虑其他模型可能更加稳妥。

同时，TransImputeNet在处理PhysioNet-2019数据集与造纸行业数据集时表现出较好的性能和稳定性，体现了它本身良好的效果，在面对高维度的时序数据集效果更显著，因为翻转的思想使得多头注意力机制对维度这一变量更加敏感，而真实工业场景下传感器数量数以百计的甚至还要多，所以TransImputeNet在处理时更有优势。综上所述，TransImputeNet在大多数场景下的时序数据插补效果都较为良好。

## 6. 消融实验

下面我们就我们作出的改进是否有效果在特定数据集上作出了实验验证：

消融实验的结果如表4所示，为了体现TransImputeNet插补的效果变化，我们选取了在上文实验结果中表现相对不好的Air-Quality数据集。

**Table 4.** Ablation experiment
**表 4.** 消融实验

| Air-Quality 数据集缺失率 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|
| TransImputeNet (without fliped) | 0.3327 | 0.3376 | 0.3632 | 0.3946 | 0.4159 | 0.4607 | 0.4666 | 0.5359 | 0.5843 |
| TransImputeNet | 0.3156 | 0.3206 | 0.3348 | 0.3705 | 0.3947 | 0.4385 | 0.4666 | 0.5444 | 0.6954 |
| TransImputeNet (without $L_2$) | 0.3264 | 0.3327 | 0.3458 | 0.3836 | 0.4138 | 0.4573 | 0.4862 | 0.5635 | 0.7009 |

在上表结果中TransImputeNet (without fliped)代表未使用翻转的TransImputeNet，代表多头注意力模块提取时间特征而FFN层提取维度特征的原始Transformer结构，TransImputeNet (without $L_2$)代表未使用$L_2$损失的TransImputeNet。

通过TransImputeNet (without fliped)与TransImputeNet的对比，可以看出在缺失率位于0.7时两个模型插补效果几乎完全相同，但是在0.7以下TransImputeNet的效果就完全领先于TransImputeNet (without fliped)，翻转的操作使得TransImputeNet在大部分情况下的效果都得到了改善，而极高的缺失率在我们的实际应用场景中并不常见，当达到一定缺失率时，数据的价值也会随之降低，因此，我们认为翻转操作

是必要的。

通过TransImputeNet (without $L_2$)与TransImputeNet的对比，可以看出在全部的缺失率情况下TransImputeNet的效果均领先，说明$L_2$损失的加入的确改善了插补效果。

## 7. 总结与展望

现实生活中确实可能存在大量缺失的情况，比如大规模的停电事故，这种情况下一般的插补模型无法捕捉其中的长期依赖关系，插补质量低，无法胜任。

针对大量缺失的高维度长序列的插补任务，我们吸取了时序建模其他领域的思想改进了原始的Transformer，训练出了插补模型，提高了多维时序数据插补的效果。

未来的研究将集中在进一步提高时间序列数据插补模型的准确性和效率上。从缺失数据插补实验的结果可以看出，TransImputeNet依然具有较大的进步空间，在低维度小规模数据集上，面对较高的缺失率，TransImputeNet的插补效果是差强人意的，此时需要借助其他深度学习插补模型填补其劣势领域，与它协同完成插补任务，所以，未来将进行以下两方面的探索：

在插补模型结构上，继续对TransImputeNet进行改造，通过做局部调整或者引入其他模块完善它的能力，使其在多种缺失情况下都有良好的表现；

在插补算法上，运用集成学习的思想将TransImputeNet与其他优秀的时序插补模型以某种方式组合起来时，提高整体的性能，通过组合多个模型来减少预测误差，提高模型的泛化能力。

## 致　谢

首先要衷心感谢我的研究生导师牛少彰教授。牛教授储备深厚、治学严谨，他以其博学之师魅力，精心引导着每一位学子的求学之路，为我们营造了一个知识的庄严殿堂，让每一位求学者都能够收获满满。在人生征程中，他对我们如春风般的关怀备至，对每一位同窗的起居行止了然于心，尽职尽责。在论文的创作过程中，牛教授从选题、构思到研究方向等各个环节都给予了我悉心指导，使我的论文得以顺利完成。牛教授那求索学术的精神、治学严谨的态度以及对工作的尽心尽责，都深深地打动了我，成为我终身学习的楷模。

其次，我要感谢实验室中与我并肩学习、并肩研究、并肩工作的同窗们。在过去的三载时光里，我们共同生活、同舟共济，共同经历了困境与欢乐，彼此扶持、共享喜悦，结下了深厚的友谊。我由衷地祝愿同窗们未来事业顺遂、生活幸福。

最后，我要感谢师兄师姐慷慨的支持。没有你们的支持与帮助，我将无法完成自己的研究使命。感谢你们一路相伴、信任与支持。

## 参考文献

[1] Cao, W., Wang, D., Li, J., *et al.* (2018) Brits: Bidirectional Recurrent Imputation for Time Series. *Advances in Neural Information Processing Systems*, **31**, 6775-6785.

[2] Che, Z., Purushotham, S., Cho, K., *et al.* (2018) Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports*, **8**, Article No. 6085. https://doi.org/10.1038/s41598-018-24271-9

[3] Yoon, J., Zame, W.R. and van der Schaar, M. (2018) Estimating Missing Data in Temporal Data Streams Using Multi-Directional Recurrent Neural Networks. *IEEE Transactions on Biomedical Engineering*, **66**, 1477-1490. https://doi.org/10.1109/TBME.2018.2874712

[4] Luo, Y., Cai, X., Zhang, Y., *et al.* (2018) Multivariate Time Series Imputation with Generative Adversarial Networks. *Advances in Neural Information Processing Systems*, **31**, 1596-1607.

[5] Luo, Y., Zhang, Y., Cai, X., *et al.* (2019) E2gan: End-to-End Generative Adversarial Network for Multivariate Time Series Imputation. *Proceedings of the* 28*th International Joint Conference on Artificial Intelligence*, Macao, 10-16

August 2019, 3094-3100. https://doi.org/10.24963/ijcai.2019/429

[6] Liu, Y., Yu, R., Zheng, S., *et al.* (2019) Naomi: Non-Autoregressive Multiresolution Sequence Imputation. *Advances in Neural Information Processing Systems*, **32**, 11236-11246.

[7] Miao, X., Wu, Y., Wang, J., *et al.* (2021) Generative Semi-Supervised Learning for Multivariate Time Series Imputation. *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**, 8983-8991. https://doi.org/10.1609/aaai.v35i10.17086

[8] Fortuin, V., Baranchuk, D., Rätsch, G., *et al.* (2020) Gp-Vae: Deep Probabilistic Time Series Imputation. *The* 23*rd International Conference on Artificial Intelligence and Statistics*, Online Conference, 26-28 August 2020, 1651-1661.

[9] Ashman, M., So, J., Tebbutt, W., *et al.* (2020) Sparse Gaussian Process Variational Autoencoders. arXiv preprint arXiv:2010.10177.

[10] Vaswani, A., Shazeer, N., Parmar, N., *et al.* (2017) Attention Is All You Need. *Advances in Neural Information Processing Systems*, **30**, 6000-6010.

[11] Wen, Q., Zhou, T., Zhang, C., *et al.* (2022) Transformers in Time Series: A Survey. arXiv preprint arXiv:2202.07125.

[12] Ma, J., Shou, Z., Zareian, A., *et al.* (2019) CDSA: Cross-Dimensional Self-Attention for Multivariate, Geo-Tagged Time Series Imputation. arXiv preprint arXiv:1905.09904.

[13] Bansal, P., Deshpande, P. and Sarawagi, S. (2021) Missing Value Imputation on Multidimensional Time Series. *Proceedings of the VLDB Endowment*, **14**, 2533-2545. https://doi.org/10.14778/3476249.3476300

[14] Du, W., Côté, D. and Liu, Y. (2023) Saits: Self-Attention-Based Imputation for Time Series. *Expert Systems with Applications*, **219**, Article 119619. https://doi.org/10.1016/j.eswa.2023.119619

[15] Liu, Y., Hu, T., Zhang, H., *et al.* (2023) iTransformer: Inverted Transformers Are Effective for Time Series Forecasting.