

# StabilizeNet:用于缓解时间序列非平稳性的新型框架

安俊秀, 万里浪

成都信息工程大学软件工程学院, 四川 成都 610225

## 摘要

时间序列预测在现代生活中广泛应用于众多领域, 其重要性日益凸显。非平稳性是影响时间序列预测准确性的主要问题之一。时间序列数据的统计特性随时间变化, 导致从历史数据中学习的规律难以被稳定地应用于未来预测, 从而增加了预测的难度和不确定性。针对该问题, 提出了一种新型框架 StabilizeNet, 旨在减少时间序列数据的非平稳性。该框架包括中心化和尺度变换、线性变换及逆序变换 3 个部分, 通过引入可学习的归一化线性变换矩阵, 优化数据的信息保留, 并增强模型对时间序列动态的捕捉能力。与先进的时间序列预测模型 Informer、SCINet、Pyraformer、FEDformer 和 Crossformer 相比, StabilizeNet 在多个数据集上展现出有效性和优越性。此框架为时间序列预测提供了一个新的预处理手段, 有助于提升时间序列预测模型的预测性能。

## 关键词

时间序列预测; 非平稳性; 归一化

中图分类号: TP183

文献标志码: A

doi:10.11959/j.issn.2096-0271.2025019

## *StabilizeNet: a novel framework for alleviating non-stationarity in time series*

AN Junxiu, WAN Lilang

School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China

## Abstract

Time series prediction is widely used in many fields in modern life, and its importance has become increasingly prominent. Non-stationarity is one of the main issues affecting the accuracy of time series predictions. Due to the statistical characteristics of time series data changing over time, it is difficult to stably apply the patterns learned from historical data to future predictions, thus increasing the difficulty and uncertainty of predictions. In order to solve this problem, a novel framework called StabilizeNet was proposed, which was designed to reduce the non-stationarity of time series data. The framework consisted of three parts: centralization and scaling transformation, linear transformation, and reverse transformation. By introducing a learnable normalized linear transformation matrix, it optimized data information retention and enhanced the model's ability to capture time series dynamics. Compared with advanced time series prediction models such as Informer, SCINet, Pyraformer, FEDformer, and Crossformer,

StabilizeNet demonstrated effectiveness and superiority on multiple datasets. This framework provides a new preprocessing method for time series prediction, which helps to improve the prediction performance of time series predictive models.

### Key words

time series predication, non-stationarity, normalization

## 0 引言

时间序列预测在现代生活中的应用越来越广泛,涵盖了异常检测<sup>[1]</sup>、股市行情分析<sup>[2]</sup>、电力负荷预测<sup>[3]</sup>、交通路况预测<sup>[4]</sup>、流感疫情预警<sup>[5]</sup>等。近几年,Transformer<sup>[6]</sup>架构的出现帮助多个领域取得了突破性的进展。在时间序列领域,研究人员发现自注意力机制能够有效地对时间序列中的复杂非线性关系进行建模,进而衍生出一系列高性能模型,极大地推动了时间序列预测的广泛应用。同时,一些研究人员也开始关注时间序列数据本身的特性。时间序列数据的主要问题在于其非平稳性,即数据的均值、方差或其他统计性质会随时间而变化,这种不稳定性导致从过往数据中学习的规律难以被稳定地应用于未来预测,从而使时间序列数据的相关预测任务变得具有挑战性。

目前,对于非平稳的时间序列数据,通常会进行归一化处理,即去除数据中平均值和标准差对应的数据。然而,一方面这种归一化操作会导致数据分布仍然相对分散,并且也会使数据失去原有的特征信息,从而降低模型的可解释性。逆归一化操作能部分恢复数据特征,但如果数据本身存在噪声,该操作会加剧还原后数据的不稳定性。另一方面,目前归一化操作还会造成信息丢失,使数据在分布上更加分散,进而减弱模型的预测效果。

针对以上问题,本文提出了一个用于缓解时间序列非平稳性的新型框架StabilizeNet。该框架主要由中心化和尺度变换、线性变换以及逆序变换3个部分组成。首先,它对输入数据进行中心化和尺度变换操作,去除数据中的不平稳信息。随后,通过一个归一化线性变换矩阵对数据进行线性变换,调整数据的分布,使数据变得更加集中,增强模型对时间序列动态的捕捉能力,以此解决常规归一化操作导致数据变得分散的问题。最后,该框架对数据逆序执行中心化和尺度变换操作,以避免中心化和尺度变换消除了数据中的有效信息,从而影响模型的预测效果。为了验证StabilizeNet的有效性,本文对数据中发生的具体改变进行深入分析,并以先进的时间序列预测模型Informer<sup>[7]</sup>、SCINet<sup>[8]</sup>、Pyraformer<sup>[9]</sup>、FEDformer<sup>[10]</sup>以及Crossformer<sup>[11]</sup>5个模型为基线模型,进行广泛的定量对比评估。

本文的贡献如下。

(1) 提出了StabilizeNet

本文设计了一个包含中心化和尺度变换、线性变换以及逆序变换的框架。该框架对时间序列数据进行了有效的预处理,能够降低数据的非平稳性。

(2) 优化了数据的信息保留

与传统的归一化方法相比,StabilizeNet在提高数据稳定性的同时,更加注重数据分布的集中性和信息的完整性。

(3) 引入了可学习归一化线性变换

矩阵

在 StabilizeNet 的线性变换部分，引入了一个可学习的归一化线性变换矩阵，它能够适应不同时间序列数据的分布，从而增强模型对时间序列动态的捕捉能力。

#### （4）广泛的定量对比评估

与当前先进的时间序列预测模型 Informer<sup>[7]</sup>、SCINet<sup>[8]</sup>、Pyraformer<sup>[9]</sup>、FEDformer<sup>[10]</sup>和 Crossformer<sup>[11]</sup>进行的比较，证明了 StabilizeNet 在多个数据集上的有效性和优越性。

## 1 相关工作

### 1.1 时间序列预测模型

时间序列预测模型在现代生活中的应用非常广泛，其发展历程可大致分为3个阶段。首先是传统统计方法（如ARIMA模型（autoregressive integrated moving average model）<sup>[12]</sup>、指数平滑<sup>[13]</sup>）和初期机器学习技术（如支持向量机<sup>[14]</sup>、随机森林<sup>[15]</sup>）的应用。随后，研究者开始探索方法的融合，比如将循环神经网络（recurrent neural network, RNN）<sup>[16]</sup>及其变种（如LSTM（long short-term memory）<sup>[17]</sup>）与卷积神经网络（convolutional neural network, CNN）<sup>[18]</sup>相结合，以解决梯度消失和爆炸问题，并提升模型的泛化能力，如Inception Time<sup>[19]</sup>和Hybrid Model<sup>[20]</sup>。最近，基于Transformer<sup>[6]</sup>的方法因其在捕捉长距离依赖性方面的出色表现而成为研究热点，涌现出了许多基于Transformer的模型，例如Informer<sup>[7]</sup>、SCINet<sup>[8]</sup>、FEDformer<sup>[10]</sup>和Crossformer<sup>[11]</sup>。Informer<sup>[7]</sup>通过引进ProbSparse Self-Attention机制大幅减小了计算和空间复杂度。SCINet<sup>[8]</sup>利用多卷

积滤波器提取下采样子序列中的丰富特征。

FEDformer<sup>[10]</sup>在频域应用注意力机制，以提高其对噪声的鲁棒性。Crossformer<sup>[11]</sup>的跨尺度嵌入层和长短距离注意模块在不同尺度间建立有效的注意力连接。

### 1.2 时间序列数据的非平稳性

在时间序列预测中，非平稳性是影响模型性能的关键因素，会导致数据的统计性质随时间变化。为解决此问题，一些研究如Wang等<sup>[21]</sup>和Li等<sup>[22]</sup>提出了结构域适应和域泛化方法，但这些方法要么依赖目标域的标签数据，要么面临数据不平衡和过拟合的风险。Kim等<sup>[23]</sup>提出了一种可逆的归一化方法（reversible instance normalization, RevIN），通过可逆归一化和仿射变换步骤，旨在消除并恢复时间序列数据中的非平稳信息。Duan等<sup>[24]</sup>通过超网络学习时间序列及其分布。Liu等<sup>[25]</sup>发现平稳处理可能引发“过平稳性”，从而影响Transformers<sup>[6]</sup>捕捉时间的相关性。基于此，他提出了Non-stationary Transformer，使用序列平稳化模块和去平稳注意力模块提升模型性能。Fan等<sup>[26]</sup>进一步探讨分布偏移，提出双Conet框架处理输入和输出空间的分布漂移，通过归一化与反归一化来适应数据变化。虽然上述方法在技术上有效，但它们都未充分考虑到归一化操作可能带来的潜在影响。归一化处理虽然能够使数据在统一尺度上进行处理，但同时也可能导致部分数据信息的损失。这种信息损失会使数据在分布上呈现出更加分散的状态，进而可能影响模型的准确性和稳定性。此外，特别是当涉及RevIN等多参数方法时，仿射变换不仅增加了计算的复杂性，还在一定程度上降低了方法的直观性和可解释性。

本文提出的 StabilizeNet 在提高数据稳定性的同时, 重视数据的集中性和信息的完整性, 以解决标准化后的数据分散问题。与 RevIN 相比, StabilizeNet 显示出其独特的简洁性和效率。RevIN 需要同时学习权重矩阵和偏置向量, 涉及多个参数的训练和调整。而 StabilizeNet 的设计更精巧, 构成更简单, 参数量也显著减少。在 StabilizeNet 中, 主要的学习目标是矩阵的分布设置, 这不仅简化了学习过程, 还使框架结构更加高效。

## 2 StabilizeNet

给定一组数据  $\mathcal{X} = \{x^{(i)}\}_{i=1}^N$  以及对应的预测结果  $\mathcal{Y} = \{y^{(i)}\}_{i=1}^N$ , 其中,  $N$  代表序列的数量。假设  $K$ 、 $T_x$  以及  $T_y$  分别代表变量的数量、输入序列长度和模型预测长度。给定一个输入序列  $x^{(i)} \in \mathbb{R}^{K \times T_x}$ ,  $x^{(i)}$  在经过 StabilizeNet 的处理后得到的输出序列为  $y^{(i)} \in \mathbb{R}^{K \times T_y}$ 。StabilizeNet 的整体架构如图 1 所示。

首先对于输入数据  $x^{(i)}$ , 计算它的平均值和标准差, 将输入数据减去平均值, 并除以标准差, 在 StabilizeNet 中, 可以称之为中心化和尺度变换。平均值和标准差的计算如式 (1)、式 (2) 所示。

$$E_t[x_{kt}^{(i)}] = \frac{1}{T_x} \sum_{j=1}^{T_x} x_{kj}^{(i)} \quad (1)$$

$$\text{Var}[x_{kt}^{(i)}] = \frac{1}{T_x} \sum_{j=1}^{T_x} (x_{kj}^{(i)} - E_t[x_{kt}^{(i)}])^2 \quad (2)$$

其中,  $k$  表示数据的第  $k$  个维度,  $t$  表示第  $t$  个时间步长。

接下来对输入数据进行中心化和尺度变换操作, 过程如式 (3) 所示。 $\epsilon$  是计算中引入的一个较小的正数值, 其作用在于防止在计算标准差的过程中分母变为零或极其接近零, 从而有效避免数值的不稳定性。

$$x_{kt}^{\prime(i)} = \left( \frac{x_{kt}^{(i)} - E_t[x_{kt}^{(i)}]}{\sqrt{\text{Var}[x_{kt}^{(i)}] + \epsilon}} \right) \quad (3)$$

经过这两步操作后, 数据中存在的部分非平稳性信息已经被去除。随后, 将生

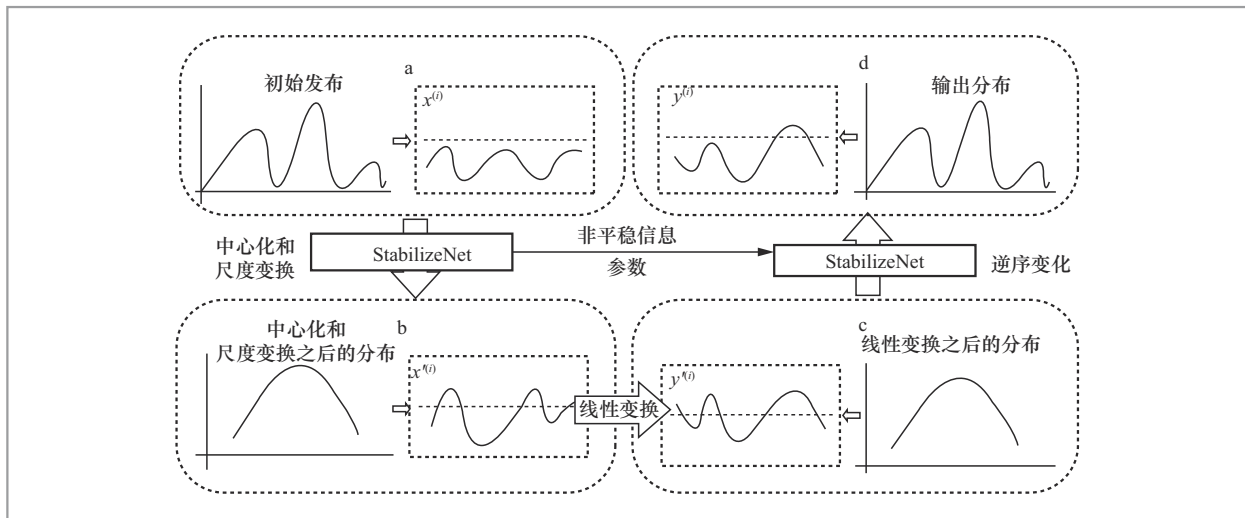


图1 StabilizeNet的整体框架

成一个矩阵  $\mathbf{M}$ ，其中每个元素从  $U(0, \delta)$ （0 到  $\delta$  的均匀分布，其中  $\delta$  被设置为一个可学习的参数）中随机采样值。随后对矩阵  $\mathbf{M}$  中的每一行进行归一化操作，使每行的元素之和为 1，从而得到归一化线性变换矩阵  $\hat{\mathbf{M}}$ 。得到归一化线性变换矩阵的过程如式（4）所示。

$$\hat{\mathbf{M}}_{ij} = \frac{\mathbf{M}_{ij}}{\sum_{k=1}^n \mathbf{M}_{ik}} \quad (4)$$

其中， $\mathbf{M}_{ij}$  表示矩阵  $\mathbf{M}$  中第  $i$  行、第  $j$  列的元素（通常  $i=j$ ）， $n$  表示矩阵的列数， $\sum_{k=1}^n \mathbf{M}_{ik}$  表示每一行的元素之和。

接下来通过归一化线性变换矩阵  $\hat{\mathbf{M}}_{ij}$  对输入数据  $\mathbf{x}_{kt}^{(0)}$  进行线性变换，并添加一个极小的正数  $\epsilon$  以确保数值的稳定性，处理后的数据为  $\bar{\mathbf{x}}_{kt}^{(0)}$ 。对输入数据进行线性变化的过程如式（5）所示。

$$\bar{\mathbf{x}}_{kt}^{(0)} = \hat{\mathbf{M}}_{ij} \cdot \mathbf{x}_{kt}^{(0)} + \epsilon \quad (5)$$

归一化线性变换矩阵每行的元素之和为 1，通过线性组合的方式能够对输入数据的特征进行重新加权，从而改变数据的分布特性。这种变换有助于减少数据中的噪声和冗余信息，使数据更加紧密地聚集在特定的区域内。此外，归一化线性变换矩阵中的元素是从不同分布中随机采样，而且这些元素可通过训练过程进行优化学习。这意味着模型可以根据数据的特性自适应地调整变换矩阵，以更好地适应不同的时间序列数据。以上操作即 StabilizeNet 中的线性变换部分，在该部分中，数据去除了部分不平稳信息，在分布上更加集中。

随后，模型将接收经过这两部分操作之后的数据  $\bar{\mathbf{x}}_{kt}^{(0)}$ ，并利用这些数据进行后续

的预测操作。经过以上操作后的数据能够提高模型的性能和收敛速度。然而此时的数据  $\bar{\mathbf{x}}_{kt}^{(0)}$  与其原始数据在分布上仍存在一定的差异，因此在模型的输出部分将对  $\mathbf{y}_{kt}^{(0)}$  执行框架的逆序变换操作，将部分被去除的不平稳信息返回给模型的输出部分。该部分操作的过程如式（6）所示。

$$\mathbf{y}_{kt}^{(0)} = \sqrt{\text{Var}[\mathbf{x}_{kt}^{(0)}] + \epsilon} \cdot \mathbf{y}_{kt}^{(0)} + E_t[\mathbf{x}_{kt}^{(0)}] \quad (6)$$

以上 3 个部分需要在模型中成对使用，且它们的使用位置是灵活的（通常将第一部分与第二部分相结合，称为平稳化，被运用在模型的输入部分，而第三部分被运用在模型的输出部分）。通过实验发现，StabilizeNet 被使用在具有 Encoder 和 Decoder 结构的模型中效果会更好。以 Transformer<sup>[6]</sup> 为例，StabilizeNet 方法的使用位置如图 2 所示。

## 3 实验

本文进行了广泛的实验，实验涉及 6 个来自于真实世界的数据集和 5 个基线模型，通过对比实验、消融实验以及相关的可视化视图进一步验证了 StabilizeNet 的通用性。

### 3.1 实验设置

#### （1）数据集

主要在 3 个大规模的真实时间序列数据集上评估本文提出的框架。

- 电力变压器温度（electricity transformer temperature, ETT）数据集：此数据集包括电力负荷和变压器油温，从我国的两个不同地区采集数据，时间跨度为两年。数据集分为 4 个子集

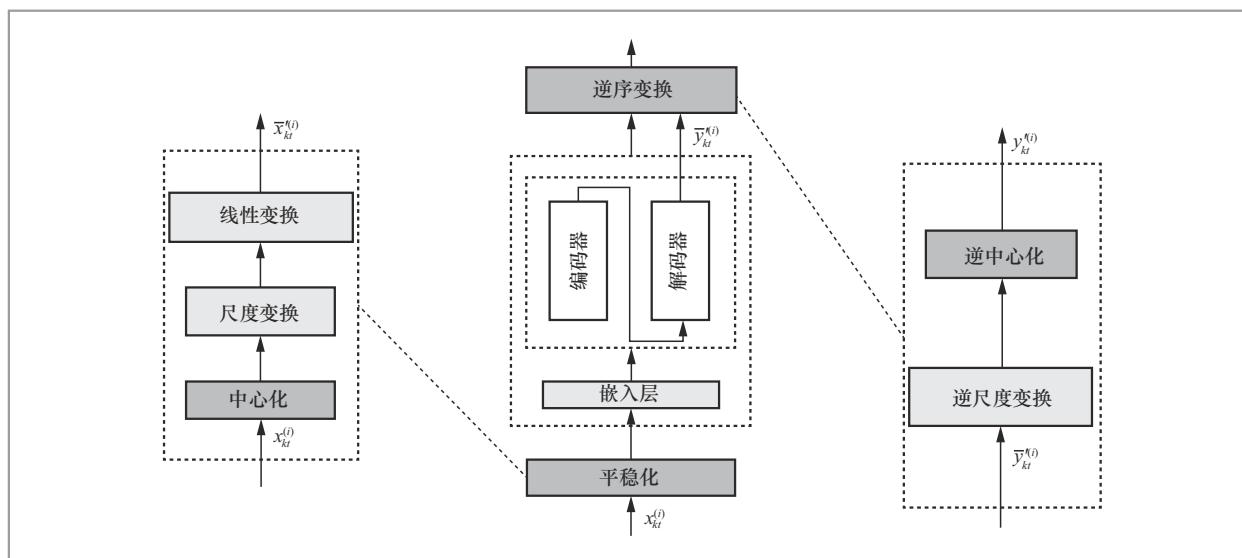


图2 将StabilizeNet应用在Transformer中

(ETTh1、ETTh2、ETTm1和ETTM2)。

- 电力消耗负载 (electricity consuming load, ECL) 数据集：此数据集包含321个客户的电力消耗数据（单位：千瓦时），以小时为单位。

- 天气 (weather, WTH) 数据集：此数据集包含美国2010—2013年约1 600个地方的气象站数据，每小时记录一次，每个数据点包括1个数据值和11个气候特征。

## (2) 超参数设置

在实验中主要采用了模型的默认超参数，并未对模型参数进行调整。框架中涉及的参数  $\epsilon$  在大多数实验中被设置为0.00001。

## (3) 评价指标

在实验中，主要使用均方根误差 (mean squared error, MSE) 和平均绝对误差 (mean absolute error, MAE) 作为评价指标。

## (4) 基线模型

StabilizeNet可以应用于多种模型。本

文选取了Informer<sup>[7]</sup>、SCINet<sup>[8]</sup>、Pyraformer<sup>[9]</sup>、FEDformer<sup>[10]</sup>和Crossformer<sup>[11]</sup>作为基线模型，通过对比实验来评估运用StabilizeNet后模型性能的变化。

## (5) 实验细节

本文提出的框架使用PyTorch平台实现，所有的实验均在一台配有12th Gen Intel(R) Core(TM) i5-12500H、NVIDIA GeForce RTX 3050 Laptop GPU和16 GB RAM的计算机上进行。遵循先前研究中有相关多变量时间序列预测的设置，学习率、Dropout等参数的设置均与基线模型在先前研究中的设置保持一致（如在Informer模型中，Dropout被设置为0.05，初始学习率被设置为0.0001）。本文对不同数据集下的预测步长进行了不同的设置。对于ETTh1数据集，预测步长分别为24、48、96、168、336、720；对于ETTh2、WTH数据集，预测步长分别为24、48、96、168、336；对于ETTm1、ECL数据集，预测步长分别为24、48、96、168、288；对于ETTM2数据集，预测步长分别

为 24、48、96、168。对于 StabilizeNet 中的  $\epsilon$  参数，在大部分情况下被设置为 0.00001。

3.2 实验结果和分析

本节展示使用 StabilizeNet 的具体效果，与当前先进的时间序列预测基线进行比较，并与当前常用的归一化方法进行比较。

(1) StabilizeNet 在时间序列预测模型上的有效性

表 1 是 StabilizeNet 在 SCINet<sup>[8]</sup>、Pyraformer<sup>[9]</sup>和 Crossformer<sup>[11]</sup>模型上的多变

量时间序列预测结果，表 1 中的数据通过对实验重复 5 次取平均值得出。该部分主要关注以上模型在不同预测步长和数据集上的表现。结果显示，StabilizeNet 在多个预测步长上均有效提升了模型性能。

例如，在数据集 ETTh1 上的预测步长从 24 增加到 720 时，SCINet 模型的预测均方根误差从 0.338 增加到 0.583，相比之下，StabilizeNet 表现出了更小的误差增加，预测均方根误差均从 0.286 增加到 0.576。其他预测长度、数据集和基线模型也出现出了类似的趋势。这些结果表明，StabilizeNet 在不同预测长度、数据集以及极限模型下的有效性。

表1 比较基线模型SCINet、Pyraformer、Crossformer在StabilizeNet下的预测误差

| 操作方法  |      | SCINet |       | +StabilizeNet |       | Pyraformer |       | +StabilizeNet |       | Crossformer |       | +StabilizeNet |       |
|-------|------|--------|-------|---------------|-------|------------|-------|---------------|-------|-------------|-------|---------------|-------|
| 数据集   | 预测步长 | MSE    | MAE   | MSE           | MAE   | MSE        | MAE   | MSE           | MAE   | MSE         | MAE   | MSE           | MAE   |
| ETTh1 | 24   | 0.338  | 0.373 | 0.286         | 0.328 | 0.521      | 0.522 | 0.338         | 0.393 | 0.341       | 0.397 | 0.322         | 0.368 |
|       | 48   | 0.436  | 0.459 | 0.342         | 0.388 | 0.575      | 0.554 | 0.381         | 0.419 | 0.383       | 0.405 | 0.343         | 0.388 |
|       | 96   | 0.438  | 0.451 | 0.411         | 0.407 | 0.714      | 0.642 | 0.428         | 0.425 | 0.401       | 0.431 | 0.391         | 0.412 |
|       | 168  | 0.442  | 0.446 | 0.429         | 0.425 | 0.764      | 0.666 | 0.530         | 0.509 | 0.429       | 0.467 | 0.413         | 0.433 |
|       | 336  | 0.536  | 0.524 | 0.484         | 0.476 | 0.906      | 0.743 | 0.558         | 0.521 | 0.481       | 0.501 | 0.442         | 0.479 |
|       | 720  | 0.583  | 0.560 | 0.576         | 0.548 | 0.953      | 0.773 | 0.577         | 0.556 | 0.556       | 0.538 | 0.497         | 0.516 |
| ETTh2 | 24   | 0.216  | 0.317 | 0.171         | 0.253 | 0.500      | 0.514 | 0.397         | 0.428 | 0.352       | 0.416 | 0.321         | 0.378 |
|       | 48   | 0.352  | 0.424 | 0.261         | 0.333 | 0.552      | 0.542 | 0.414         | 0.438 | 0.399       | 0.431 | 0.358         | 0.391 |
|       | 96   | 0.468  | 0.478 | 0.331         | 0.380 | 0.669      | 0.614 | 0.462         | 0.463 | 0.418       | 0.448 | 0.403         | 0.432 |
|       | 168  | 0.572  | 0.536 | 0.381         | 0.413 | 0.768      | 0.668 | 0.492         | 0.481 | 0.451       | 0.474 | 0.412         | 0.451 |
|       | 336  | 0.674  | 0.589 | 0.460         | 0.471 | 0.916      | 0.749 | 0.559         | 0.516 | 0.499       | 0.517 | 0.466         | 0.492 |
| ETTh1 | 24   | 0.336  | 0.382 | 0.306         | 0.348 | 0.337      | 0.389 | 0.326         | 0.385 | 0.232       | 0.319 | 0.204         | 0.282 |
|       | 48   | 0.368  | 0.392 | 0.351         | 0.373 | 0.414      | 0.441 | 0.382         | 0.416 | 0.321       | 0.376 | 0.273         | 0.328 |
|       | 96   | 0.379  | 0.423 | 0.372         | 0.416 | 0.493      | 0.490 | 0.392         | 0.424 | 0.347       | 0.393 | 0.301         | 0.341 |
|       | 168  | 0.396  | 0.447 | 0.392         | 0.432 | 0.652      | 0.619 | 0.448         | 0.453 | 0.384       | 0.427 | 0.333         | 0.372 |
|       | 288  | 0.441  | 0.481 | 0.401         | 0.443 | 0.743      | 0.657 | 0.467         | 0.465 | 0.427       | 0.453 | 0.393         | 0.427 |
| ETTh2 | 24   | 0.140  | 0.253 | 0.105         | 0.195 | 0.224      | 0.350 | 0.223         | 0.347 | 0.283       | 0.407 | 0.241         | 0.305 |
|       | 48   | 0.158  | 0.265 | 0.133         | 0.225 | 0.378      | 0.460 | 0.287         | 0.350 | 0.312       | 0.441 | 0.287         | 0.332 |
|       | 96   | 0.209  | 0.311 | 0.183         | 0.272 | 0.451      | 0.525 | 0.304         | 0.369 | 0.374       | 0.481 | 0.336         | 0.371 |
|       | 168  | 0.295  | 0.368 | 0.249         | 0.319 | 0.750      | 0.618 | 0.327         | 0.376 | 0.511       | 0.547 | 0.381         | 0.427 |

表 2 是 StabilizeNet 运用在 FEDformer<sup>[10]</sup>和 Informer<sup>[7]</sup>两个模型上的多变量时间序列预测结果，表 2 中的数据通过对实验重复 5 次取平均值得出。StabilizeNet 在提高 FEDformer 和 Informer 模型的预测精度方面表现出显著效果。使用 StabilizeNet 后，两种模型在所有数据集和预测步长下的 MSE 和 MAE 均显著降低，表明模型预测误差减小，性能提升。这种提升在 Informer 模型上更明显。例如，在数据集 ETTh1 上的预测步长从 24 增加到 720 时，Informer 的预测均方根误差从 0.572 增加到 1.436，相比之下，StabilizeNet 表现出了更小的误差增加，预测均方根误差均从 0.406 增加到 0.693。

对于其他预测长度、数据集也表出现了相同的趋势。

笔者还认为模型使用 StabilizeNet 后的性能与模型本身的具体构造相关，StabilizeNet 应用于 Informer 及其他基线模型带来的平均性能提升见表 3。综合表 1、表 2、表 3，在 Informer<sup>[7]</sup>、SCINet<sup>[8]</sup>、Pyraformer<sup>[9]</sup>、FEDformer<sup>[10]</sup>和 Crossformer<sup>[11]</sup>模型中，StabilizeNet 都提升了模型的性能。例如，在应用 StabilizeNet 后，Informer 模型在所有数据集上的 MSE 和 MAE 均显著降低。以 ETTh1 数据集为例，预测步长为 720 时，MSE 从 1.436 降至 0.693，MAE 从 1.217 降至 0.557，性能提升分别为 51.74% 和 54.23%。StabilizeNet

表 2 比较 Informer 和 FEDformer 模型在使用 StabilizeNet 后预测误差

| 操作方法  |      | FEDformer |       | +StabilizeNet |       | Informer |       | +StabilizeNet |       |
|-------|------|-----------|-------|---------------|-------|----------|-------|---------------|-------|
| 数据集   | 预测步长 | MSE       | MAE   | MSE           | MAE   | MSE      | MAE   | MSE           | MAE   |
| ETTm2 | 24   | 0.137     | 0.239 | 0.116         | 0.226 | 0.283    | 0.407 | 0.241         | 0.305 |
|       | 48   | 0.187     | 0.270 | 0.144         | 0.242 | 0.312    | 0.441 | 0.287         | 0.332 |
|       | 96   | 0.203     | 0.291 | 0.162         | 0.256 | 0.374    | 0.481 | 0.336         | 0.371 |
|       | 168  | 0.243     | 0.341 | 0.209         | 0.287 | 0.511    | 0.547 | 0.381         | 0.427 |
| WTH   | 24   | 0.184     | 0.249 | 0.177         | 0.239 | 0.353    | 0.401 | 0.332         | 0.365 |
|       | 48   | 0.208     | 0.261 | 0.191         | 0.240 | 0.397    | 0.433 | 0.367         | 0.411 |
|       | 96   | 0.231     | 0.301 | 0.209         | 0.281 | 0.504    | 0.512 | 0.482         | 0.477 |
|       | 168  | 0.291     | 0.353 | 0.271         | 0.319 | 0.607    | 0.569 | 0.492         | 0.483 |
|       | 336  | 0.383     | 0.419 | 0.365         | 0.411 | 0.617    | 0.576 | 0.527         | 0.520 |
| ECL   | 24   | 0.161     | 0.284 | 0.154         | 0.267 | 0.254    | 0.362 | 0.141         | 0.241 |
|       | 48   | 0.173     | 0.291 | 0.167         | 0.281 | 0.279    | 0.373 | 0.166         | 0.268 |
|       | 96   | 0.195     | 0.309 | 0.181         | 0.283 | 0.299    | 0.394 | 0.183         | 0.284 |
|       | 168  | 0.204     | 0.321 | 0.187         | 0.291 | 0.311    | 0.403 | 0.213         | 0.317 |
|       | 288  | 0.229     | 0.341 | 0.201         | 0.317 | 0.321    | 0.408 | 0.220         | 0.323 |
| ETTh1 | 24   | 0.336     | 0.407 | 0.319         | 0.377 | 0.572    | 0.550 | 0.406         | 0.420 |
|       | 48   | 0.351     | 0.415 | 0.337         | 0.391 | 0.789    | 0.658 | 0.462         | 0.463 |
|       | 96   | 0.385     | 0.427 | 0.376         | 0.415 | 1.102    | 0.891 | 0.541         | 0.533 |
|       | 168  | 0.421     | 0.448 | 0.401         | 0.432 | 1.132    | 0.932 | 0.625         | 0.549 |
|       | 336  | 0.467     | 0.491 | 0.448         | 0.461 | 1.312    | 1.031 | 0.681         | 0.551 |
|       | 720  | 0.491     | 0.531 | 0.481         | 0.471 | 1.436    | 1.217 | 0.693         | 0.557 |

表3 StabilizeNet应用于Informer及其他基线模型带来的平均性能提升

| 数据集           | ETTh1        |              | ETTTh2       |               | ETTM1        |              | ETTM2        |              |
|---------------|--------------|--------------|--------------|---------------|--------------|--------------|--------------|--------------|
|               | MSE          | MAE          | MSE          | MAE           | MSE          | MAE          | MSE          | MAE          |
| Informer      | 1.436        | 1.217        | 4.997        | 1.987         | 0.750        | 0.626        | 0.511        | 0.547        |
| +StabilizeNet | <b>0.693</b> | <b>0.557</b> | <b>0.498</b> | <b>0.496</b>  | <b>0.464</b> | <b>0.464</b> | <b>0.311</b> | <b>0.358</b> |
| 模型性能提升        | 51.74%       | 54.23%       | 90.03%       | 75.04%        | 38.13%       | 25.88%       | 39.14%       | 34.55%       |
| SCINet        | 0.462        | 0.468        | 0.456        | 0.468         | 0.384        | 0.396        | 0.200        | 0.299        |
| +StabilizeNet | <b>0.421</b> | <b>0.428</b> | <b>0.320</b> | <b>0.370</b>  | <b>0.364</b> | <b>0.384</b> | <b>0.167</b> | <b>0.252</b> |
| 模型性能提升        | 8.87%        | 8.55%        | 29.82%       | 20.94%        | 5.21%        | 3.03%        | 16.50%       | 15.54%       |
| Pyraformer    | 0.738        | 0.650        | 0.680        | 0.617         | 0.527        | 0.519        | 0.450        | 0.488        |
| +StabilizeNet | <b>0.468</b> | <b>0.469</b> | <b>0.464</b> | <b>0.465</b>  | <b>0.403</b> | <b>0.428</b> | <b>0.285</b> | <b>0.360</b> |
| 模型性能提升        | 36.59%       | 27.85%       | 31.75%       | 24.65%        | 23.65%       | 17.45%       | 36.72%       | 26.16%       |
| Crossformer   | 0.431        | 0.456        | 0.4238       | 0.4572        | 0.342        | 0.393        | 0.370        | 0.469        |
| +StabilizeNet | <b>0.401</b> | <b>0.432</b> | <b>0.392</b> | <b>0.4288</b> | <b>0.300</b> | <b>0.350</b> | <b>0.311</b> | <b>0.358</b> |
| 模型性能提升        | 6.96%        | 5.26%        | 7.50%        | 6.21%         | 12.10%       | 11.08%       | 15.88%       | 23.51%       |

对Informer<sup>[7]</sup>和Pyraformer<sup>[9]</sup>模型性能的提升最大，对其他3个基线模型也存在一定程度的性能提升。

(2) 与当前常用的归一化方法的比较  
比较StabilizeNet以及当前常用的归一化方法运用在Informer模型上的多变量时间序列预测结果，见表4。评价指标为MSE，表4中的数据通过对实验重复5次取平均值得出（如Layer Norm、Batch

Norm、Instance Norm、Deep Adaptive Input Normalization（DAIN）<sup>[26]</sup>以及RevIN<sup>[27]</sup>）。

由表4可知，相较于其他归一化方法，StabilizeNet在ETTh1和ETTTh2数据集上的表现更佳。在预测步数较长时，其他归一化方法的性能显著下降，而StabilizeNet能够保持较高的预测性能，具有更好的稳定性。表4中的数据也说明了逆归一化操作

表4 与当前常用的归一化方法的比较

| 数据集    | 预测步长 | Layer Norm | Batch Norm | Instance Norm | DAIN  | RevIN | StabilizeNet |
|--------|------|------------|------------|---------------|-------|-------|--------------|
| ETTh1  | 24   | 0.483      | 0.506      | 0.489         | 0.478 | 0.504 | <b>0.406</b> |
|        | 720  | 0.991      | 1.004      | 0.981         | 0.967 | 0.982 | <b>0.693</b> |
| ETTTh2 | 24   | 0.489      | 0.513      | 0.482         | 0.551 | 0.257 | <b>0.215</b> |
|        | 336  | 0.914      | 0.960      | 0.936         | 0.917 | 0.597 | <b>0.498</b> |
| ETTM1  | 24   | 0.363      | 0.296      | 0.296         | 0.342 | 0.322 | <b>0.275</b> |
|        | 288  | 1.158      | 0.712      | 1.038         | 0.789 | 0.742 | <b>0.712</b> |
| ETTM2  | 24   | 0.255      | 0.250      | 0.211         | 0.242 | 0.312 | <b>0.241</b> |
|        | 168  | 1.157      | 0.740      | 0.599         | 0.950 | 0.402 | <b>0.381</b> |

的必要性。值得注意的是,在该实验中,Layer Norm、Batch Norm、Instance Norm和DAIN<sup>[26]</sup>未进行相应的逆操作。

### (3) 消融实验

本节旨在从数据以及模型效果两个方面论证 StabilizeNet 的有效性。图3展示了选取的 ETTh1 和 ETTm1 两个数据集中数据在经过 StabilizeNet 处理前后的密度分布情况。对于每个数据集,随机选择了 10 000 条数据。

由图3可知,平稳化操作后,数据密度分布没有发生太大变化,但仍存在密度较低且分布不集中的现象,在 ETTm1 数据集中,数据甚至出现了密度降低的情况。随后,再通过归一化线性变换矩阵对数据进行线性变化。经过该操作后,可以观察到数据的密度得到了一定的提升,并且分布更加集中。在 ETTh1 数据集中,数据的密度由 0.372 左右提升到了 0.791 左右,数据也更加靠近  $[-2, 2]$ ; 在 ETTm1

数据集中,数据的密度由不到 0.368 提升到了 0.734 左右,数据也更加靠近  $[-2, 2]$ 。

本文还比较了多变量时间序列预测中,Informer 模型在使用 StabilizeNet 各部分操作后的 MAE 和 MSE,见表5。表5中的数据为实验重复 5 次的平均值。实验结果显示,仅通过平稳化操作,模型的性能就有了显著的提升。具体而言,在 ETTh2 数据集上,平稳化操作对模型的性能提升效果最显著,平均提升了 84.74% 和 65.65%,即便在提升效果最不明显的 ETTm2 数据集上,也分别提升了 6.76% 和 7.25%。此外,仅利用归一化线性变换矩阵对数据进行线性变换也在一定程度上提升了模型的性能。然而,当平稳化操作与归一化线性变换矩阵相结合使用,即完整应用本文提出的 StabilizeNet 时,模型预测性能的提升最大。以上结果证实了本文的假设,即在数据平稳化的基础上,再通过归一化线性变换矩阵对数据进行线性变换,可以更

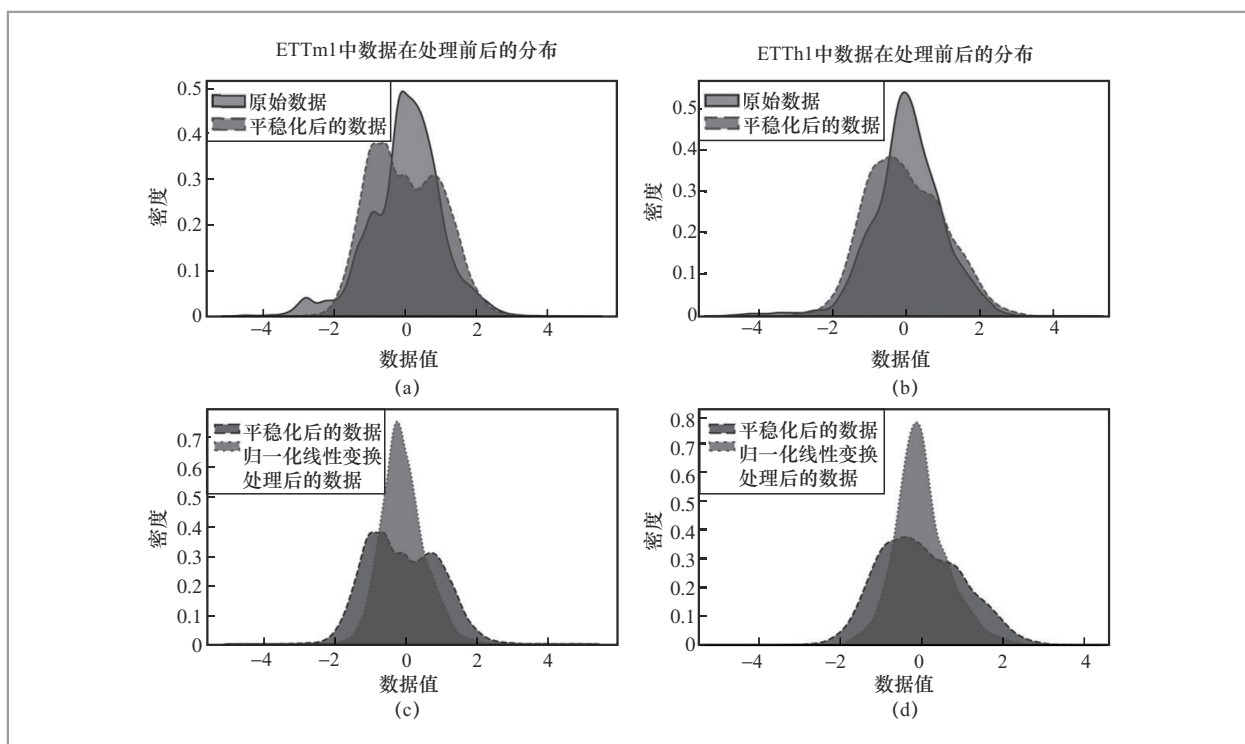


图3 StabilizeNet中每个步骤对数据影响的密度分布

表5 StabilizeNet中各步骤对模型效果的影响

| 操作方法   |      | Informer |       | +平稳化   |        | +线性变换  |        | +StabilizeNet |               |
|--------|------|----------|-------|--------|--------|--------|--------|---------------|---------------|
| 数据集    | 预测步长 | MSE      | MAE   | MSE    | MAE    | MSE    | MAE    | MSE           | MAE           |
| ETTth1 | 24   | 0.572    | 0.550 | 0.433  | 0.458  | 0.525  | 0.533  | <b>0.406</b>  | <b>0.420</b>  |
|        | 48   | 0.789    | 0.658 | 0.494  | 0.501  | 0.587  | 0.577  | <b>0.462</b>  | <b>0.463</b>  |
|        | 96   | 1.102    | 0.891 | 0.571  | 0.564  | 0.755  | 0.683  | <b>0.541</b>  | <b>0.533</b>  |
|        | 168  | 1.132    | 0.932 | 0.638  | 0.582  | 0.791  | 0.673  | <b>0.625</b>  | <b>0.549</b>  |
|        | 720  | 1.312    | 1.031 | 0.701  | 0.601  | 0.965  | 0.793  | <b>0.693</b>  | <b>0.557</b>  |
| 模型性能提升 |      |          |       | 42.18% | 33.38% | 26.17% | 19.77% | <b>42.67%</b> | <b>36.13%</b> |
| ETTh1  | 24   | 0.336    | 0.363 | 0.331  | 0.367  | 0.319  | 0.385  | <b>0.275</b>  | <b>0.332</b>  |
|        | 48   | 0.515    | 0.502 | 0.46   | 0.471  | 0.497  | 0.496  | <b>0.355</b>  | <b>0.379</b>  |
|        | 96   | 0.612    | 0.574 | 0.584  | 0.545  | 0.572  | 0.521  | <b>0.390</b>  | <b>0.420</b>  |
|        | 168  | 0.997    | 0.772 | 0.611  | 0.561  | 0.801  | 0.671  | <b>0.591</b>  | <b>0.508</b>  |
|        | 288  | 1.292    | 0.923 | 0.732  | 0.702  | 0.848  | 0.704  | <b>0.712</b>  | <b>0.683</b>  |
| 模型性能提升 |      |          |       | 27.56% | 15.57% | 19.06% | 11.39% | <b>38.09%</b> | <b>25.91%</b> |
| ETTth2 | 24   | 0.580    | 0.595 | 0.441  | 0.449  | 0.521  | 0.531  | <b>0.215</b>  | <b>0.305</b>  |
|        | 48   | 2.316    | 1.233 | 0.487  | 0.479  | 1.512  | 1.214  | <b>0.317</b>  | <b>0.354</b>  |
|        | 96   | 4.363    | 1.726 | 0.528  | 0.501  | 2.532  | 1.221  | <b>0.437</b>  | <b>0.412</b>  |
|        | 168  | 4.772    | 1.834 | 0.551  | 0.531  | 2.912  | 1.482  | <b>0.471</b>  | <b>0.480</b>  |
|        | 336  | 4.997    | 1.987 | 0.591  | 0.573  | 3.212  | 1.786  | <b>0.498</b>  | <b>0.496</b>  |
| 模型性能提升 |      |          |       | 84.74% | 65.65% | 37.23% | 15.47% | <b>88.62%</b> | <b>72.24%</b> |
| ETTh2  | 24   | 0.283    | 0.407 | 0.273  | 0.385  | 0.312  | 0.387  | <b>0.241</b>  | <b>0.305</b>  |
|        | 48   | 0.312    | 0.441 | 0.291  | 0.412  | 0.327  | 0.401  | <b>0.287</b>  | <b>0.332</b>  |
|        | 96   | 0.374    | 0.481 | 0.351  | 0.442  | 0.354  | 0.412  | <b>0.336</b>  | <b>0.371</b>  |
|        | 168  | 0.511    | 0.547 | 0.465  | 0.501  | 0.398  | 0.461  | <b>0.381</b>  | <b>0.427</b>  |
| 模型性能提升 |      |          |       | 6.76%  | 7.25%  | 6.01%  | 11.46% | <b>15.88%</b> | <b>23.51%</b> |

有效地集中数据分布，从而显著提高模型的预测性能。

4 结束语

本文创新性地提出了StabilizeNet，专门用于解决时间序列预测中的非平稳性问题。该框架旨在提高数据的稳定性，同时确保信息被完整保留。一系列对比实验和消融实验证实了StabilizeNet在多个数据集和基准模型上的卓越性能。该框架的核

心优势在于一个独特的、可学习的归一化线性变换矩阵，它可以适应各种时间序列的分布特性，这一特点在传统归一化技术中并不常见。总体而言，本文为处理非平稳数据的时间序列预测提供了一种新的技术途径，并且有望推动基于Transformer架构的其他时间序列预测模型的发展。未来的研究将进一步优化StabilizeNet组件，深入评估其在多样化场景中的性能，并将其扩展，以应对更广泛的时间序列预测挑战。

## 参考文献:

- [1] XU J H, WU H X, WANG J M, et al. Anomaly transformer: time series anomaly detection with association discrepancy[EB]. arXiv preprint, 2021, arXiv: 2110.02642.
- [2] IDREES S M, ALAM M A, AGARWAL P. A prediction approach for stock market volatility based on time series data[J]. IEEE Access, 2019, 7: 17287–17298.
- [3] HONG T, FAN S. Probabilistic electric load forecasting: a tutorial review[J]. International Journal of Forecasting, 2016, 32(3): 914–938.
- [4] SHAO H X, SOONG B H. Traffic flow prediction with long short-term memory networks (LSTMs)[C]//Proceedings of the 2016 IEEE Region 10 Conference (TEN-CON). Piscataway: IEEE Press, 2016: 2986–2989.
- [5] SALGOTRA R, GANDOMI M, GANDOMI A H. Time series analysis and forecast of the COVID-19 pandemic in India using genetic programming[J]. Chaos, Solitons & Fractals, 2020, 138: 109945.
- [6] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[EB]. arXiv preprint, 2017, arXiv: 1706.03762.
- [7] ZHOU H Y, ZHANG S H, PENG J Q, et al. Informer: beyond efficient transformer for long sequence time-series forecasting[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(12): 11106–11115.
- [8] LIU M, ZENG A, CHEN M, et al. Scinet: time series modeling and forecasting with sample convolution and interaction[J]. Advances in Neural Information Processing Systems, 2022, 35: 5816–5828.
- [9] LIU S, YU H, LIAO C, et al. Pyraformer: low-complexity pyramidal attention for long-range time series modeling and forecasting[C]// Proceedings of International conference on learning representations. [S.l.: s.n.], 2021.
- [10] ZHOU T, MA Z, WEN Q, et al. Fedformer: frequency enhanced decomposed transformer for long-term series forecasting[EB]. arXiv preprint, 2022, arXiv: 2201.12740.
- [11] ZHANG Y, YAN J. Crossformer: transformer utilizing cross-dimension dependency for multivariate time series forecasting[C]// Proceedings of the Eleventh International Conference on Learning Representations. [S.l.: s.n.], 2023.
- [12] BOX G E P, PIERCE D A. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models[J]. Journal of the American Statistical Association, 1970, 65(332): 1509–1526.
- [13] GARDNER JR E S. Exponential smoothing: the state of the art[J]. Journal of Forecasting, 1985, 4(1): 1–28.
- [14] NOBLE W S. What is a support vector machine?[J]. Nature Biotechnology, 2006, 24(12): 1565–1567.
- [15] BELGIU M, DRĂGUT L. Random forest in remote sensing: a review of applications and future directions[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2016, 114: 24–31.
- [16] MEDSKER L R, JAIN L C. Recurrent neural networks[J]. Design and Applications, 2001, 5(64–67): 2.
- [17] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735–1780.
- [18] GU J X, WANG Z H, KUEN J, et al. Recent advances in convolutional neural networks[J]. Pattern Recognition, 2018, 77: 354–377.
- [19] ISMAIL FAWAZ H, LUCAS B, FOESTER G, et al. InceptionTime: finding AlexNet for time series classification[J]. Data Mining and Knowledge Discovery,

- 2020, 34(6): 1936–1962.
- [20] ZHANG G P. Time series forecasting using a hybrid ARIMA and neural network model[J]. Neurocomputing, 2003, 50: 159–175.
- [21] WANG J D, FENG W J, CHEN Y Q, et al. Visual domain adaptation with manifold embedded distribution alignment[C]//Proceedings of the 26th ACM international conference on Multimedia. New York: ACM, 2018: 402–410.
- [22] LI H L, PAN S J, WANG S Q, et al. Domain generalization with adversarial feature learning[C]//Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2018: 5400–5409.
- [23] KIM T, KIM J, TAE Y, et al. Reversible instance normalization for accurate time-series forecasting against distribution shift[C]//Proceedings of International Conference on Learning Representations. [S.l.: s.n.], 2021.
- [24] DUAN W Y, HE X X, ZHOU L, et al. Combating distribution shift for accurate time series forecasting via hypernetworks[C]//Proceedings of the 2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS). Piscataway: IEEE Press, 2023: 900–907.
- [25] LIU Y, WU H X, WANG J M, et al. Non-stationary transformers: exploring the stationarity in time series forecasting[EB]. arXiv preprint, 2022, arXiv: 2205.14415.
- [26] FAN W, WANG P Y, WANG D K, et al. Dish-TS: a general paradigm for alleviating distribution shift in time series forecasting[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2023, 37(6): 7522–7529.
- [27] PASSALIS N, TEFASA A, KANNIAINEN J, et al. Deep adaptive input normalization for time series forecasting[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(9): 3760–3765.

## 作者简介



安俊秀（1970–），女，成都信息工程大学软件工程学院教授、中国计算机学会高级会员，主要研究方向为大数据、云计算。



万里浪（2000–），男，成都信息工程大学软件工程学院硕士生，主要研究方向为时间序列预测、云计算。

收稿日期: 2024-08-02

通信作者: 安俊秀, anjunxiu@cuit.edu.cn

基金项目: 国家社会科学基金项目 (No.22XWB01214)

Foundation Item: The National Social Science Foundation of China(No.22XWB01214)