融合全局和序列特征的多变量时间序列预测方法

李兆玺[1)] 刘红岩[2)]

[1)](中国人民大学信息学院数据工程与知识工程教育部重点实验室 北京 100872)

[2)](清华大学经济管理学院管理科学与工程系 北京 100084)

摘 要 时间序列在现实生活中具有广泛的用途，使用时间序列预测模型能够预估序列的未来变化趋势，为决策提供支撑. 对于多变量时间序列的预测研究，已经提出了很多模型，但已有方法存在如下问题：不能同时考虑时间序列本身和协变量的信息；忽略了多变量时间序列中的全局信息；不能对预测结果进行解释. 针对这些问题，本文提出了一个基于深度学习的多变量时间序列预测模型 TEDGER，可以提取隐藏在单个时间序列中的序列模式和隐藏在多变量时间序列中的全局特征，并将序列模式和全局特征进行融合，通过残差预测的方式实现时间序列的预测. 本文所提模型在真实的时间序列数据集上进行了实验评估. 结果表明，本文提出的模型在预测准确度上超越了其他基准模型，同时模型拥有一定的可解释性.

关键词 时间序列预测；全局特征；矩阵分解；深度学习；注意力机制

中图法分类号 TP391 **DOI 号** 10.11897/SP.J.1016.2023.00070

Combining Global and Sequential Patterns for Multivariate Time Series Forecasting

LI Zhao-Xi[1)] LIU Hong-Yan[2)]

[1)](*School of Information, Renmin University of China, Key Lab of Data Eng. and Know. Engineering, MOE, Renmin University of China, Beijing* 100872)

[2)](*School of Economics and Management, Department of Management Science and Engineering, Tsinghua University, Beijing* 100084)

Abstract Time series is widely used in real world with many applications nowadays. By developing time series forecasting models, we can predict how the time series evolve in the future, which provides support for decision making in different scenarios. Many models have been proposed in literature. Existing studies on multivariate time series forecasting either cannot take both times series and covariates into consideration, lack interpretability, or ignore global trends across multivariate time series. To solve these issues, we propose a new deep learning based multivariate time series forecasting model TEDGER (Tensorized recurrent encoder-decoder framework with global patterns and residual forecasting). TEDGER can capture sequential patterns hidden in individual time series and can extract global trends hidden across multivariate times series. Specifically, our proposed model follows encoder and decoder framework and we adopt Tensorized Long Short-Term Memory network as the basic processing unit to provide the model with possibility to distinguish the importance of different features. The encoder is designed to capture sequential pattern hidden in each time series, in which two different kinds of attentions are designed to weigh the importance of historical information and covariate information, offering interpretability to the model in the same time. Global change pattern hidden in multiple time series is extracted based on temporal regularized

收稿日期：2021-07-17；在线发布日期：2022-05-31. 本课题得到国家社会科学基金（20&ZD161）资助. **李兆玺**，硕士，主要研究领域为数据挖掘、时间序列分析和计算机视觉. E-mail: lizhaoxi@ruc.edu.cn. **刘红岩**（通信作者），博士，教授，中国计算机学会高级会员，主要研究领域为人工智能、商务智能、推荐系统、金融科技、计算机视觉等. E-mail: hyliu@tsinghua.edu.cn.

matrix factorization. We further propose two different ways to combine sequential pattern and global trend to make residual prediction for final time series forecasting, which correspond to two variants of our proposed model: TEDGER I 和 TEDGER II. We provide time complexity analysis of the proposed model. Meanwhile, we conduct comprehensive experiments on real-world time series datasets to evaluate the performance of the proposed model. We compare the performance of our proposed model with eight benchmark models, which belong to three different categories. The comparison results demonstrate superior performance of our proposed model over benchmark models. Ablation study is conducted to evaluate the necessity of the global pattern exaction module and results confirm its benefit. We also check the impact of different input window size and different future steps in prediction on the model's performance. Case study shows that our proposed model can explain the forecasting results to some extent.

Keywords　time series forecasting; global pattern; matrix factorization; deep learning; attention mechanism

1　引　言

时间序列（Time Series）是指一组按照时间先后顺序排列的数据点序列，且一般情况下每个数据点所在的时间步之间的间隔相同. 在多变量时间序列中，存在多个与时间相关的变量，每个时间步对应多个变量值.

时间序列在现实生活中具有广泛的用途，例如经济金融、工业、气象、交通等领域每天都在产生大量的时间序列数据. 通过对这些时间序列的分析和挖掘，可以发现蕴含在数据中的潜在规律，更清晰地认识事物. 时间序列预测任务是时间序列分析领域的一个重要研究方向，通过对未来的预测，还可以预估未来事物发展的趋势，从而为决策提供支撑. 同时，由于时间序列预测经常服务于决策的制定，如果模型缺乏对预测结果的解释能力，决策者可能就无法相信预测结果并以此为依据制定决策.

近年来，随着互联网的普及以及科技的不断发展，各行各业、各种各样的时间序列数据被大量采集. 传统的时间序列预测模型难以捕捉特征变量之间的复杂关系，而深度学习模型则在学习复杂非线性关系方面更胜一筹. 因此，如何基于深度学习方法构建准确且可解释的时间序列预测模型既具有理论研究价值，也具有重要的现实意义.

本文主要研究使用深度学习模型对多变量时间序列进行建模并预测的方法. 在现实世界中，同一场景下的不同时间序列往往遵从一些全局的特征. 例如，在纳斯达克股票价格这一时间序列中，不同公司的股票价格往往共同跟随大盘指数变动. 这对预测时间序列有重要的意义. 单变量时间序列预测丰富则无法充分利用多变量时间序列之间的相互关系进行预测，因此，本文研究多变量时间序列的预测问题.

时间序列的协变量（Covariate）是指除了要预测的目标变量之外，与预测目标变量相关的其他变量. 在时间序列预测任务中，不能忽略协变量的作用，因为其中可能蕴含了有用的外部环境信息.

此外，可解释性也是时间序列预测任务中不可忽视的要素. 因为预测任务经常要服务于决策制定者，而一个无法解释的“黑盒”模型产生的预测结果无法令人信赖，可解释的预测结果对决策者更有意义.

虽然此前已有很多关于时间序列预测的研究工作，但现有的研究工作中或是不能同时考虑协变量和序列本身的信息，或是忽视了全局信息的影响，或是不能提供可解释的预测结果.

本文致力于解决时间序列预测任务中的上述问题. 目标是通过深度学习技术构建多变量多步的时间序列预测模型，能够充分考虑每个序列及其协变量中的信息，以及跨序列的全局信息，提升预测准确度. 同时，期望模型能提供可解释性.

本文的主要贡献总结如下：

（1）提出了一种基于深度学习的多变量时间序列预测模型 TEDGER (Tensorized recurrent encoder-decoder framework with global patterns and residual forecasting). 该模型使用一个基于张量化长短期记忆网络的深度循环编码器-解码器框架，解码器使用两种注意力机制，为预测结果提供解释性依据. 利用基于时序正则化的矩阵分解方法来提取跨序列的全局特征，用来编码隐藏在序列和协变量中

的局部规律. 同时，通过两种权重计算方法将全局特征重新组合成新的趋势序列，并将其作为未来预测的基准，在此基础上进行残差预测，以进一步提高预测精度.

（2）在多个真实的时间序列数据集上对所提的模型进行了实验，结果表明，本文提出的模型在预测性能上超越了其他基准模型，同时模型能够对预测结果进行解释.

2 相关工作

针对时间序列预测问题，已有很多的研究工作. 早期的研究工作主要关注统计学方法，如自回归模型[1]和指数平滑模型[2]. 这些模型只能建模特征之间的线性关系，限制了模型的预测精度. 随着深度学习在各领域的应用愈发广泛，深度学习模型可以建模特征之间的非线性关系，具有更强的学习能力，因此，基于深度学习的时间序列预测方法也受到了很多关注. 这些工作根据其输入输出不同，主要可以分为全局方法和局部方法两类.

基于深度学习的全局预测方法，通常是将多变量时间序列拼成一个矩阵作为输入，同样的，在预测时也是同时预测多个序列. Lai 等人[3]提出了一种结合卷积神经网络（Convolutional Neural Network，CNN）和循环神经网络（Recurrent Neural Network，RNN）的单步预测模型 LSTNet，使用 CNN 提取多变量时间序列的特征，然后使用 RNN 建模长短期时序模式进行单步预测，并设计了循环跳跃机制来捕捉周期性的依赖，实验结果表明该模型预测效果显著优于传统统计学方法和机器学习方法. Shih 等人[4]使用 RNN 建立时间序列的时序正则化，并使用 CNN 来提取频域信息，随后通过注意力机制结合频域信息进行单步预测. Huang 等人[5]使用 CNN 来捕捉单序列的两种时序特征，并使用自注意力机制来建模跨序列之间的依赖. 这类全局方法的优点是能够捕捉跨序列的全局特征，但是忽略了协变量信息对于每个序列预测的作用，也没有考虑模型的可解释性.

基于深度学习的局部预测方法，是将单个时间序列及其协变量输入神经网络，在预测时也分别预测各序列的未来值，而模型的参数在序列间共享. 相比全局预测方法，更多的方法属于局部预测方法. Salinas 等人[6]提出了一种基于 RNN 的多步概率预测模型 DeepAR，将协变量作为每个时间步的输入，并输出未来时间步预测值的概率分布. Qin 等人[7]使用 RNN 提取时序信息，随后通过双阶段的注意力机制分别挖掘不同协变量和不同时间步的重要程度，并对未来时间步的序列进行预测. Rangapuram 等人[8]将神经网络与状态空间模型相结合，提出了深度状态空间模型（Deep State Space Model，DSSM），模型将时间序列输入 RNN 中以获得线性状态空间模型的参数，并以此进行未来若干步序列的概率预测. Fan 等人[9]使用了多模态的注意力机制以更好地结合不同阶段的历史信息，并将其用于未来时间步的预测. 最近，在自然语言处理领域提出的多层自注意力机制（Transformer）也被应用于时间序列预测任务，Li 等人[10]设计了卷积自注意力以更好地融入上下文信息，并解决了经典 Transformer 用于时间序列预测的内存瓶颈. Zhou[11]等人主要面向长序列的长距离依赖的捕捉问题，解决 transformer 用于时间序列预测时的时间和空间复杂度高的问题. Feng 等人[12] 研究河流流量预测方法，解决人工神经网络模型学习过程的收敛速度慢和易陷入局部极小点问题，提出将合作搜索算法与神经网络模型结合优化模型参数的方法. 文献[13, 14]主要研究多模态输入的异质性和如何有效融合异质特征的方法. 文献[15] 研究时间序列的非平稳性等问题，提出一种新的数据标准化方法，以消除数据的非平稳性. 文献[16]提出使用时域卷积和普通卷积相结合的方法提取单时间序列的特征，结合注意力机制进行时序预测. 文献[17]则研究地铁站点客流时间序列的预测方法，将地铁站点客流时间序列进行聚类后，比较用 LSTM 模型进行预测的性能的区别和影响因素. 没有考虑全局模式以及可解释性. 这类局部方法能够结合序列本身信息和协变量中蕴藏的额外信息，但是却忽略了跨序列的全局变化特征.

Sen 等人[18]于 2019 年提出了一种能够结合序列信息和全局特征的模型 DeepGLO，该模型通过将矩阵分解的预测值同协变量一同输入时序卷积网络（Temporal Convolutional Network，TCN），以获得下个时间步的预测值，但未能充分利用全局特征. Yang 等人[19]研究金融时间序列的预测，人为构建了反映跨地区相互关联的特征，而不是从多个时间序列中自动提取全局特征. 文献[20, 21]研究如何利用图神经网络获得更多的序列之间的关联信息，这类方法需要借助外部或领域知识构建序列信息之间的关系网络，限制了应用范围. 文献[22]研究时间序列预测中的迁移学习方法. 文献[23，24]主要研究时间序列预测的集成方法，本文所提模型也可以作为集成方法中的基模型.

虽然深度学习模型在时间序列预测任务上能够取得更精准的效果，但是由于深度学习的“黑盒”

性质，预测结果往往得不到解释[25]. 而在现实的时间序列预测任务中，如果缺少解释性的证据，预测结果的可信性就会大打折扣，因此时间序列预测任务中的可解释性是一个重要的研究话题. 也有一些工作关注深度时间序列预测模型的解释性. Guo 等人[26]使用张量化长短期记忆网络编码序列信息，并通过注意力机制计算每个时间步的权重，以及每个协变量对于预测结果的影响，以提供可解释的单步预测结果. Pantiskas 等人[27]使用 TCN 提取多变量时间序列中的特征，并通过自注意力机制融合不同协变量的影响，以预测未来的协变量序列，并通过将预测的协变量序列输入其他机器学习算法以实现可解释的预测. Lim 等人[28]提出了时序融合自注意力模型（Temporal Fusion Transformers，TFT）用于可解释的多步时间序列预测，该工作首先使用 RNN 编码协变量信息，并通过多头自注意力机制融合不同时间步的信息. 作者设计了变量选择模块来衡量不同协变量的重要性，并通过自注意力部分的注意力权重得到每个时间步的时序重要性.

除了注意力机制之外，还有一些工作利用显著性方法对预测结果进行解释. Assaf 等人[29]通过分箱将时间序列预测任务转化成分类问题，利用 CNN 提取多变量序列特征得到分类预测结果，并通过反向计算特征图的平均梯度得到不同协变量在不同时刻的重要性热力图. Ismail 等人[30]提出已有的显著性方法虽然能识别出时间序列预测任务中的重要时间步，却无法区分重要事件步中的重要特征. 为此提出了两步时间显著性重构方法，大幅提高了显著图结果的质量.

此外，Li 等人[31]提出了一种以 KL 散度作为目标函数的深度状态空间模型，通过为每个输入的协变量学习一个权重来解释预测结果. Oreshkin 等人[32]提出了一种名为 N-BEATS 的可解释性时间序列预测模型，该模型包括若干个叠加的神经网络模块，每个模块输入并预测上层模块的预测残差，并将所有模块的预测结果累加之后得到最终预测结果，该模型还可以通过将时间序列分解为趋势子序列和周期子序列，以解释预测结果. 然而，这些工作在预测的精准性和全局特征的利用上仍有提升的空间.

3　融合全局和序列特征的预测模型

3.1　问题定义

考虑一个变量数为 M 的历史时间序列矩阵 $\boldsymbol{Y}\in\mathbb{R}^{M\times T}$，其中序列 $\boldsymbol{y}^i$ 是第 i 个变量的时间序列. 已知 $\boldsymbol{Y}$ 在过去 T 个时间步的值为 $\boldsymbol{Y}_{1:T}$. 与此同时，已知和 $\boldsymbol{Y}$ 相关的协变量张量 $\boldsymbol{X}\in\mathbb{R}^{M\times N\times(T+\tau)}$，其中，N 是协变量的个数，每个子矩阵 $\boldsymbol{X}^i$ 都是随时间变化的序列，代表与序列 $\boldsymbol{y}^i$ 相关的协变量矩阵. 与时间序列 $\boldsymbol{Y}$ 不同，假设所有的协变量 $\boldsymbol{X}$ 在未来的时间步上是已知的或者是可以被推断出来的，故可以在预测时使用.

本文的目标是学习一个模型 $\mathcal{F}(\cdot)$，可以利用过去的时间序列 $\boldsymbol{Y}_{1:T}$，以及过去和未来的协变量 $\boldsymbol{X}_{1:T+\tau}$，预测出未来 τ 个时间步的时间序列 $\hat{\boldsymbol{Y}}_{T+1:T+\tau}$，即

$$\hat{\boldsymbol{Y}}_{T+1:T+\tau}=F(\boldsymbol{Y}_{1:T},\boldsymbol{X}_{1:T+\tau}\,|\,\boldsymbol{\theta}) \tag{1}$$

$\boldsymbol{\theta}$ 是模型的参数. 与此同时，模型 $\mathcal{F}(\cdot)$ 能够对预测结果进行解释. 为了解决该问题，本文提出了预测模型 TEDGER，其基本架构图如图 1 所示.

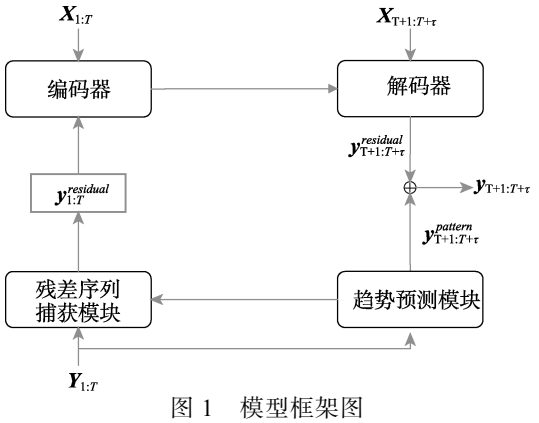

图 1　模型框架图

3.2　预测模型TEDGER

预测模型 TEDGER 主要由四个模块构成：编码器、解码器、残差序列捕获和趋势预测模块. 历史时间序列 $Y_{1:T}$ 经残差序列捕获模块和趋势模块处理之后得到原时间序列的残差序列，作为编码器的输入，编码器对残差序列和协变量进行编码后得到序列的隐状态，输入解码器得到未来 τ 时间步的残差预测. 趋势预测模块捕获历史时间序列的全局特征和未来变化趋势模式. 最后，结合未来残差的预测和变化趋势的预测得到最终的预测结果. 下面分别对各模块进行介绍.

3.2.1　编码器和解码器

编码器用于对历史时间序列的残差序列的每个时间步进行编码，捕获每个时间步的隐状态，解码器则根据编码器获得的隐状态预测未来多个时间步的残差值. 为解决已有工作存在的问题，编码器-解码器基于张量长短期记忆网络（Tensorized Long

Short-Term Memory network，TLSTM）构建[13]，以避免每个时间步的不同协变量中的信息被混合，且保持普通的编码器-解码器框架在时间序列预测任务的精度. 模型框架如图 2 所示.

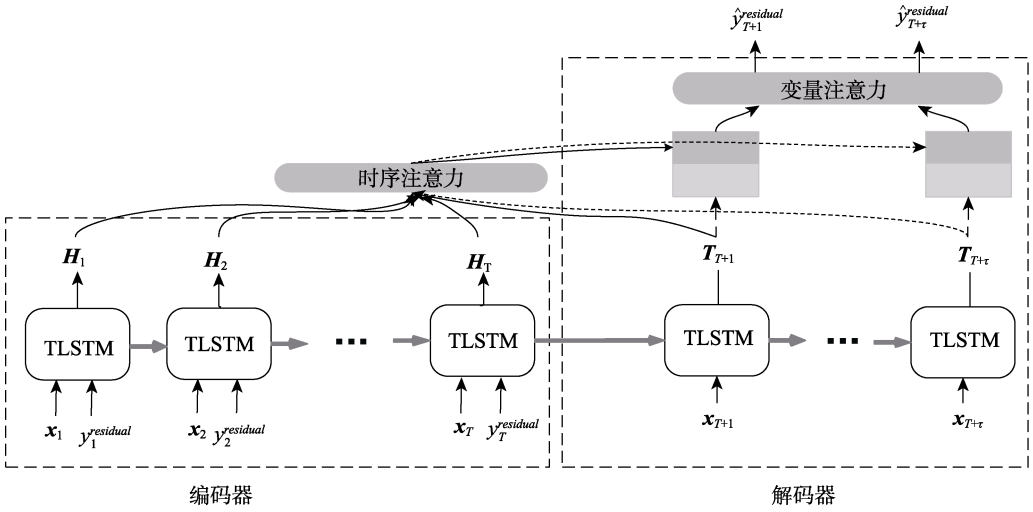

图 2 编码器和解码器模型

编码器部分提取每个时间步的输入的信息，获得编码器阶段的所有历史时间步的特征：

$$\boldsymbol{H}_{1:T}=[\boldsymbol{H}_1,\cdots,\boldsymbol{H}_T]^{\top}$$

解码器使用 TLSTM 提取未来时间步的协变量的信息，且与编码器部分共享相同的参数. 通过解码器得到未来 τ 步的特征：

$$\boldsymbol{H}_{T+1:T+\tau}=[\boldsymbol{H}_{T+1},\boldsymbol{H}_{T+2},\cdots,\boldsymbol{H}_{T+\tau}]^{\top}$$

以长短期记忆网络（Long Short-Term Memory networks，LSTM）为代表的循环神经网络被广泛应用于序列建模任务中，然而，将传统的 LSTM 网络应用于时间序列预测任务时，存在一些问题. 当时间序列与协变量被输入到 LSTM 单元后，每个时间步的所有协变量信息都被混合在隐状态向量中. 因此，使用传统的 LSTM 网络将无法得知每个时间步的每个协变量对于预测结果的不同贡献，这极大地削弱了模型的预测结果的可解释性. 为此，本文改用张量长短期记忆网络.

张量长短期记忆网络（TLSTM）用于编码每个残差序列 $y^{i,residual}$ 的信息，以及该序列所对应的协变量中的信息 $\boldsymbol{X}^i$，为了表述简便起见，上标 i 将在本章的后续部分省略. TLSTM 将传统 LSTM 中每个时间步 t 的隐状态向量 $\boldsymbol{h}_t$ 替换为隐状态矩阵 $\boldsymbol{H}_t=[\boldsymbol{h}_t^1,\boldsymbol{h}_t^2,\cdots,\boldsymbol{h}_t^N]^{\top}\in\mathbb{R}^{N\times d}$. 在隐状态矩阵中，每一 $\boldsymbol{h}_t^j\in\mathbb{R}^d$ 表示的是仅与第 j 个协变量 $\boldsymbol{x}^j$ 相关的隐状态部分. 通过这种方法，每个协变量只影响隐状态空间中的对应部分，从而保证了不同协变量的影响在隐状态空间中的独立. 一个 TLSTM 单元的结构[13]如图 3 所示.

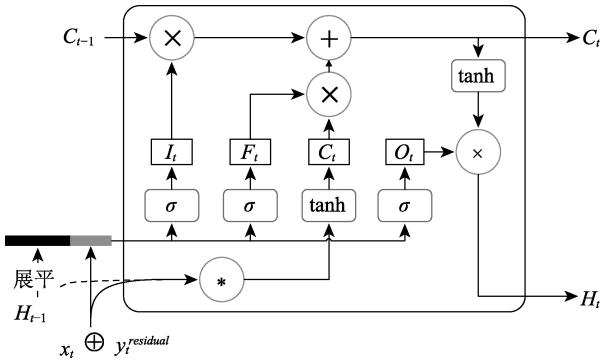

图 3 TLSTM 单元

TLSTM 单元的状态转移方程如下：

$$\begin{aligned}\boldsymbol{I}_t&=mat(\sigma(\boldsymbol{W}_i[\boldsymbol{x}_t\oplus y_t^{\text{residual}}\oplus \text{vec}(\boldsymbol{H}_{t-1})]+\boldsymbol{b}_i))\\ \boldsymbol{F}_t&=mat(\sigma(\boldsymbol{W}_f[\boldsymbol{x}_t\oplus y_t^{\text{residual}}\oplus \text{vec}(\boldsymbol{H}_{t-1})]+\boldsymbol{b}_f))\quad(1)\\ \boldsymbol{O}_t&=mat(\sigma(\boldsymbol{W}_o[\boldsymbol{x}_t\oplus y_t^{\text{residual}}\oplus \text{vec}(\boldsymbol{H}_{t-1})]+\boldsymbol{b}_o))\end{aligned}$$

$$\tilde{\boldsymbol{C}}_t=\tanh(\boldsymbol{W}_c\otimes\boldsymbol{H}_{t-1}+\boldsymbol{U}_c\circledast(\boldsymbol{x}_t\oplus y_t^{\text{residual}})+\boldsymbol{B}_c)\quad(2)$$

$$\boldsymbol{C}_t=\boldsymbol{F}_t\odot\boldsymbol{C}_{t-1}+\boldsymbol{I}_t\odot\tilde{\boldsymbol{C}}_t\quad(3)$$

$$\boldsymbol{H}_t=\boldsymbol{O}_t\odot tanh(\boldsymbol{C}_t)\quad(4)$$

在 TLSTM 中，不仅是隐状态(hidden state) $\boldsymbol{H}_t\in\mathbb{R}^{(N+1)\times d}$ 和细胞状态（cell state）$\boldsymbol{C}_t\in\mathbb{R}^{(N+1)\times d}$ 是矩阵形式，输入门 $\boldsymbol{I}_t\in\mathbb{R}^{(N+1)\times d}$、输出门 $\boldsymbol{O}_t\in\mathbb{R}^{(N+1)\times d}$ 和遗忘门 $\boldsymbol{F}_t\in\mathbb{R}^{(N+1)\times d}$ 也为矩阵形式.

在编号为（1）的三个公式中，$\boldsymbol{W}_i$、$\boldsymbol{W}_f$ 和 $\boldsymbol{W}_o \in \mathbb{R}^{(N+1)d\times((N+1)d+N+1)}$ 表示计算不同门控矩阵的权重矩阵，$\boldsymbol{b}_i$、$\boldsymbol{b}_f$ 和 $\boldsymbol{b}_o \in \mathbb{R}^{(N+1)d}$ 表示偏差向量. vec(·) 表示将矩阵展开成向量的操作，而 mat(·) 表示将向量复原成矩阵的操作. $\sigma(\cdot)$ 代表 sigmoid 激活函数，符号 $\oplus$ 表示拼接两个向量的操作.

在公式（2）中，$\boldsymbol{W}_c \in \mathbb{R}^{(N+1)\times d\times d}$ 是隐状态矩阵的转移张量，$\boldsymbol{W}_c=[\boldsymbol{W}_c^1,\cdots,\boldsymbol{W}_c^{N+1}]^\top$，$\otimes$ 代表张量转移操作，其含义为 $\boldsymbol{W}_c\otimes\boldsymbol{H}=[\boldsymbol{W}_c^1\boldsymbol{h}^1,\cdots,\boldsymbol{W}_c^{N+1}\boldsymbol{h}^{N+1}]^\top$，即对于 $\boldsymbol{H}$ 中的每一行 $\boldsymbol{h}^j$，都乘以对应的转移矩阵 $\boldsymbol{W}_c^j$，再将每一行的结果拼起来得到结果矩阵，以避免隐状态矩阵不同行之间的相互影响. $\boldsymbol{U}_c \in \mathbb{R}^{(N+1)\times d\times 1}$ 表示输入的转移矩阵，$\boldsymbol{U}_c=[u_c^1,\cdots,u_c^{N+1}]^\top$，$\circledast$ 代表输入转移操作，其含义为

$$\boldsymbol{U}_c\circledast(\boldsymbol{x}_t\oplus y_t^{\text{residual}})=[u_c^1x_t^1,\cdots,u_c^Nx_t^N,u_c^{N+1}y_t^{\text{residual}}]^\top$$

与张量转移操作类似，输入转移操作也保证了每个输入的协变量 x_t^j 只会影响对应的隐状态 $\boldsymbol{h}^j$. 因此，候选细胞状态 $\tilde{\boldsymbol{C}}_t=[\tilde{\boldsymbol{c}}_t^1,\cdots,\tilde{\boldsymbol{c}}_t^{N+1}]^\top$ 中的每个子状态 $\tilde{\boldsymbol{c}}_t^j$ 只根据协变量 x_t^j 以及 $\boldsymbol{h}_{t-1}^j$ 的值来确定，这个过程是完全与其他协变量隔绝的，保证了隐状态矩阵中每一行之间的独立性，并且每个协变量对于预测的贡献都是相互独立、互不干扰的，避免了传统 LSTM 网络在可解释性方面的问题.

为了简洁起见，本文将公式（1）~（4）中 TLSTM 单元的内部操作统一归纳表示为

$$\boldsymbol{H}_t=TLSTM(\boldsymbol{x}_t,\boldsymbol{H}_{t-1}) \tag{5}$$

在解码器部分使用了两种注意力机制，以充分地利用历史序列信息和协变量信息. 首先，使用时序注意力机制来关注哪一个历史时间步对于未来的第 t 个时间步的预测最相关. 对于第 j 个协变量，$\alpha_{i,t}^j$ 表示的是其在第 i 个历史时间步的值与第 t 个未来时间步的相关性，具体计算方法为

$$r_{i,t}^j=\boldsymbol{v}^\top\tanh(\boldsymbol{W}_r[\boldsymbol{h}_i^j\oplus\boldsymbol{h}_t^j]+\boldsymbol{b}_r)$$

$$\alpha_{i,t}^j=\frac{\exp(r_{i,t}^j)}{\sum_{k=1}^T\exp(r_{k,t}^j)} \tag{6}$$

$$\boldsymbol{c}_t^j=\sum_{i=1}^T\alpha_{i,t}^j\boldsymbol{h}_i^j$$

其中 $\boldsymbol{c}_t^j$ 代表第 j 个协变量在解码器的第 t 个时间步的上下文向量，参数 $\boldsymbol{v}\in\mathbb{R}^d$ 和 $\boldsymbol{W}_r\in\mathbb{R}^{d\times 2d}$ 表示权重矩阵. 通过将 $\boldsymbol{h}_t^j$ 与 $\boldsymbol{c}_t^j$ 拼接得到：

$$\boldsymbol{g}_t^j=[\boldsymbol{h}_t^j\oplus\boldsymbol{c}_t^j] \tag{7}$$

随后可以计算基于第 j 个协变量的相关信息，得到未来的第 t 个时间步预测值 μ_t^j：

$$\mu_t^j=\boldsymbol{w}_\mu^\top\boldsymbol{g}_t^j+b_\mu \tag{8}$$

同时，使用变量注意力机制以融合基于不同协变量得到的不同预测值：

$$s_t^j=\boldsymbol{w}_s^\top\boldsymbol{g}_t^j+b_s\ \ \beta_t^j=\frac{\exp(s_t^j)}{\sum_{n=1}^N\exp(s_t^n)} \tag{9}$$

其中 β_t^j 代表基于第 j 个协变量得到的未来的第 t 个时间步预测值在最终预测结果中的贡献. 这样就得到了对于未来第 t 个时间步的最终预测值 $\hat{y}_t$：

$$\hat{y}_t^{\text{residual}}=\sum_{j=1}^N\beta_t^j\mu_t^j \tag{10}$$

通过以上的处理方法可以得到未来每个时间步的最终预测结果：

$$\hat{y}_{T+1:T+t}^{\text{residual}}=\left[\hat{y}_{T+1}^{\text{residual}},\hat{y}_{T+2}^{\text{residual}},\ldots,\hat{y}_{T+\tau}^{\text{residual}}\right] \tag{11}$$

3.2.2　趋势预测模块

为了捕获多变量时间序列间的关系，本文使用时序正则化矩阵分解（Temporal Regularized Matrix Factorization，TRMF）[33]来提取跨序列的全局演化趋势. 将已知的时间序列矩阵 $\boldsymbol{Y}\in\mathbb{R}^{M\times T}$ 作为输入，矩阵分解技术可以将其分解为两个低秩矩阵 $\boldsymbol{P}\in\mathbb{R}^{M\times k}$ 和 $\boldsymbol{G}\in\mathbb{R}^{k\times T}$，矩阵 $\boldsymbol{P}$ 中的每一行 $\boldsymbol{p}_i$ 表示第 i 个序列的特征，矩阵 $\boldsymbol{G}$ 中的每一列 $\boldsymbol{g}_t$ 表示的是 t 时刻的全局时序特征，如图 4 所示. 传统的矩阵分解技术缺乏预测未来的能力，TRMF 可以通过引入时序依赖，由已知的时序特征 $\boldsymbol{G}_{1:T}$ 推断出未来的时序特征 $\boldsymbol{G}_{T+1:T+\tau}$.

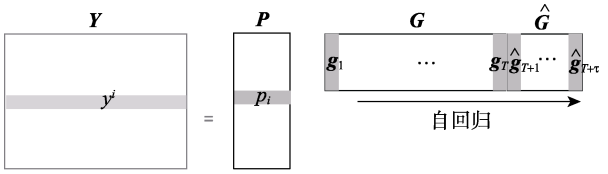

图 4　时序正则化矩阵分解示意

矩阵分解的损失函数为

$$\mathcal{L}_{TRMF}=\sum_{i,t}(y_t^i-\boldsymbol{p}_i^\top g_t)^2+\lambda_1\left\|P\right|_2+\lambda_2\left\|G\right|_2+\lambda_3\mathcal{R}_{AR}(\boldsymbol{G})$$

其中，$\lambda_3\mathcal{R}_{AR}(\boldsymbol{G})$ 是基于线性自回归的时序正则化项：

$$\mathcal{R}_{AR}(\boldsymbol{G})=\frac{1}{2}\sum_{t}\left(\boldsymbol{g}_t-\sum_{l}w_r^l\boldsymbol{g}_{t-l}\right)^2+\lambda_4\left\|\boldsymbol{w}_r\right\|_2$$

该项的含义是，在训练时该模型期望每一个 t 时刻的时序特征 $\boldsymbol{g}_t$ 都可以用过去若干步的时序特征 $\boldsymbol{g}_{t-l}$，以 w_r^l 为权重经过线性回归推断得到. 这样可以预测未来任意时刻 t 的时序特征：

$$\hat{\boldsymbol{g}}_t=\sum_{l}w_r^l\boldsymbol{g}_{t-l},\ t=T+1,\ldots,T+\tau \tag{12}$$

3.2.3 残差序列捕获模块

为了将提取到的全局特征融合至模型中，一种比较直接的方法是将序列特征 $\boldsymbol{p}_i$ 和全局特征 $\boldsymbol{G}$ 输入至模型中[34]. 在对序列 $\boldsymbol{y}^i$ 进行预测时，将 $\boldsymbol{p}_i\in\mathbb{R}^k$ 作为额外的 k 个变量输入到编码器和解码器的每一个时间步中. 与此同时，将每个时间步 t 的全局特征 $\boldsymbol{g}_t\in\mathbb{R}^k$ 作为另外的 k 个变量输入. 需要注意的是，在编码器阶段，每个时间步的全局特征 $\boldsymbol{g}_t$ 是通过 TRMF 分解得到的；而在解码器阶段，为了避免来自未来的信息泄漏，未来时间步的全局特征是未知的，需要通过自回归方法进行预测.

将提取出的序列和全局特征同其他协变量一同输入到预测模型的方法存在不足，矩阵分解得到的全局特征不同于其他的协变量，对于预测应该可以发挥更重要的作用. Sen 等人[11]认为，可以将矩阵分解视为将原序列表达成若干个基本序列的线性组合. 因此，本文进一步提出了一种新的融合模型，将 TRMF 得到的全局特征重新组合成新的序列，反映目标序列在未来的基本变化趋势，将其作为未来预测的基准，在此基础上进行残差预测. 本文提出两种不同的趋势序列的构建方法，对应模型 TEDGER 的两种不同版本：TEDGER I 和 TEDGER II.

对于序列 $\boldsymbol{y}^i$，给定 TRMF 得到的序列特征 $\boldsymbol{p}_i$ 和全局特征 $\boldsymbol{G}_{1:T}$，首先以 $\boldsymbol{p}_i$ 为权重，得到合成的趋势序列：

$$\boldsymbol{y}_{1:T}^{\text{pattern}}=\boldsymbol{p}_i\boldsymbol{G}_{1:T} \tag{13}$$

随后将原始序列减去趋势序列得到残差序列：

$$\boldsymbol{y}_{1:T}^{\text{residual}}=\boldsymbol{y}_{1:T}-\boldsymbol{y}_{1:T}^{\text{pattern}} \tag{14}$$

并将残差序列 $\boldsymbol{y}_{1:T}^{\text{residual}}$ 代替原始序列 $\boldsymbol{y}_{1:T}$ 输入编码器，相应的，在预测阶段解码器也仅预测未来序列的残差：

$$\hat{\boldsymbol{y}}_{T+1:T+\tau}^{\text{residual}}=decode(\boldsymbol{y}_{1:T}^{\text{residual}},\boldsymbol{X}_{1:T+\tau}) \tag{15}$$

随后通过时序依赖计算未来的全局特征及趋势序列：

$$\hat{\boldsymbol{y}}_{T+1:T+\tau}^{\text{pattern}}=\boldsymbol{p}_i\hat{\boldsymbol{G}}_{T+1:T+\tau} \tag{16}$$

累加未来的趋势序列与预测的残差序列得到最终的预测结果为：

$$\hat{\boldsymbol{y}}=\hat{\boldsymbol{y}}_{T+1:T+\tau}^{\text{residual}}+\hat{\boldsymbol{y}}_{T+1:T+\tau}^{\text{pattern}} \tag{17}$$

在此，本文认为全局特征实际上是若干个所有序列共同遵循的基本趋势，在通过全局特征合成趋势序列之后，只需根据未来的协变量信息，进一步预测未来的真实序列与趋势序列之间的差值（即残差）即可. 这样比直接预测未来序列要更为精准，因为趋势序列的应用能够为预测未来的序列提供基本的引导. 本文将这种新的融合全局特征的残差预测模型命名为 TEDGER I.

在 TEDGER I 中，由于趋势序列是直接以 TRMF 得到的序列特征作为权重合成的，而序列特征不随时间变化，这可能导致在计算趋势序列时，未能有效利用时间变化的信息. 因此本文设计了另外一种合成趋势序列的方法. 利用一个 LSTM 编码器来编码已知的目标序列 $\boldsymbol{y}_{1:T}$，并以此计算将全局特征合成的权重：

$$\begin{aligned}\boldsymbol{h}_g&=\text{LSTM}(\boldsymbol{y}_{1:T})\\ \boldsymbol{w}_g&=W\boldsymbol{h}_g\\ \boldsymbol{y}_{1:T}^{\text{pattern}}&=\boldsymbol{w}_g\boldsymbol{G}_{1:T}\end{aligned} \tag{18}$$

此 LSTM 网络的参数同原有模型的参数一同学习. 通过这个额外的编码器能够充分利用原始序列中的时序信息，将 TRMF 矩阵分解得到的若干基本序列合成为一个趋势序列，使其相比于直接利用序列特征作为权重，能够更符合当前窗口的情况. 使用此方法进行趋势序列合成的预测模型称为 TEDGER II.

通过 TEDGER 模型分别预测所有的 M 个序列，就可以得到所有序列的预测结果 $\hat{\boldsymbol{Y}}_{T+1:T+\tau}=[\hat{\boldsymbol{y}}_{T+1:T+\tau}^1,\cdots,\hat{\boldsymbol{y}}_{T+1:T+\tau}^M]^\top$. 本文选择均方误差作为模型训练的损失函数：

$$\mathcal{L}_{\text{TEDGER}}=\frac{1}{M\times\tau}\sum_{m=1}^{M}\sum_{t=T+1}^{T+\tau}(y_t^m-\hat{y}_t^m)^2 \tag{19}$$

3.3 模型预测结果的解释

除了给出未来序列在一段时间内的预测结果之外，模型的可解释性非常重要. 在现有的可解释性工作中，一个重要的关注点就是试图研究深度学习模型输入与输出的之间的相关性，换句话说，哪部分输入对于预测结果贡献最大. 本文尝试从这个角

度出发去解释模型的预测结果.

在 TEDGER 模型中，时序和变量两种注意力机制提供了有关时序和变量对于结果的重要性. 具体来说，可以从模型的参数中提取出时序注意力权重 $\alpha(m,w,i,j,n)$ 和变量注意力权重 $\beta(m,w,n,j)$. 其中 m,w,n 分别代表序列、滑动窗口和协变量的索引，而 i 和 j 则分别对应每个滑动窗口中编码器和解码器的时间步数. $\alpha(m,w,i,j,n)$ 代表的是对于第 m 个序列的第 w 个滑动窗口中的第 n 个协变量，历史时间步 i 对于解码器中的第 j 个时间步的预测的贡献. 而 $\beta(m,w,n,j)$ 表示的是在第 m 个序列的第 w 个滑动窗口中，协变量 n 对于解码器中的第 j 个时间步的最终预测结果 y_j 的贡献. 通过这种方式可以得知，在历史输入中，哪些时段的哪些协变量对于与模型的预测结果是最为相关的，以此对解释进行模型.

基于上述注意力权重，使用以下三种方法来解释所提模型的预测结果：

（1）变量重要性（Variable Importance）

$$\bar{\beta}(n)=\frac{1}{M\times W\times\tau}\sum_{m=1}^{M}\sum_{w=1}^{W}\sum_{j=1}^{\tau}\beta(m,w,n,j) \tag{20}$$

通过公式（20），可以得到协变量 n 在该数据集中的平均贡献，即在输入的若干协变量中，哪个协变量对于得到最终的预测结果是最为重要的.

（2）时序重要性（Temporal Importance）

$$\bar{\alpha}(i,j)=\frac{1}{M\times W\times N}\sum_{m=1}^{M}\sum_{w=1}^{W}\sum_{n=1}^{N}\alpha(m,w,i,j,n) \tag{21}$$

公式（21）计算的是编码器阶段的时间步 i 对于解码器阶段第 j 个时间步的预测的平均贡献，这反映了模型中哪部分的输入被考虑的更多.

（3）重要事件挖掘（Significant Events Mining）.

上面所分析的时序重要性关注的是在一个滑动窗口内各时间步的时序重要性，然而在一个数据集中，时间序列的突然波动往往对预测结果有着重要的影响. 本文参考文献[15]所提出的方法，分析时间序列中的重要事件（突然的剧烈波动）.

对于滑动窗口 w，首先计算该窗口对于未来时间步 j 的时序重要性的向量：

$$\tilde{\boldsymbol{\alpha}}(w,i,j)=\frac{1}{M\times N}\sum_{m=1}^{M}\sum_{n=1}^{N}\alpha(m,w,i,j,n) \tag{22}$$

$$\tilde{\boldsymbol{\alpha}}(w,j)=\left[\tilde{\boldsymbol{\alpha}}(w,1,j),\cdots,\tilde{\boldsymbol{\alpha}}(w,T,j)\right]^{\top} \tag{23}$$

考虑所有的滑动窗口，则计算得到平均向量：

$$\bar{\boldsymbol{\alpha}}(j)=\left[\bar{\boldsymbol{\alpha}}(1,j),\cdots,\bar{\boldsymbol{\alpha}}(T,j)\right]^{\top} \tag{24}$$

随后利用巴氏系数计算两个向量之间的距离：

$$dist\left(\tilde{\boldsymbol{\alpha}}(w,j),\bar{\boldsymbol{\alpha}}(j)\right)=\sqrt{1-\sum_{i=1}^{T}\sqrt{\tilde{\boldsymbol{\alpha}}(w,i,j)\bar{\boldsymbol{\alpha}}(i,j)}} \tag{25}$$

如果在某个窗口序列出现了剧烈的波动，在该窗口的注意力权重向量就会与所有窗口的平均权重向量有较大的距离，通过该方式可以探测到这种波动.

考虑该窗口预测的所有未来时间步，计算得到该滑动窗口的重要程度：

$$\gamma(w)=\frac{1}{\tau}\sum_{j=1}^{\tau}\mathrm{dist}\left(\tilde{\boldsymbol{\alpha}}(w,j),\bar{\boldsymbol{\alpha}}(j)\right) \tag{26}$$

如果 $\gamma(w)$ 大于某个阈值，那么则认为在窗口 w 产生了重要事件.

3.4　模型复杂度分析

首先分析一下模型的空间复杂度，在公式（1）中，$\boldsymbol{W}_i$、$\boldsymbol{W}_f$ 和 $\boldsymbol{W}_o\in\mathbb{R}^{(N+1)d\times((N+1)d+N+1)}$，$\boldsymbol{b}_i$、$\boldsymbol{b}_f$ 和 $\boldsymbol{b}_o\in\mathbb{R}^{(N+1)\times d}$. 每个 TLSTM 单元中的每个门对应的参数量为 $(N+1)\cdot d\cdot((N+1)d+N+1)+(N+1)\cdot d=(N+1)^2\cdot d^2+(N+1)^2\cdot d+(N+1)\cdot d$，令 $D=(N+1)\cdot d$，则每个门对应的参数量为 $D^2+(N+1)D+D=D^2+ND+2D$. 因此，公式（1）对应的三个门的参数量为 $3(D^2+ND+2D)=3D^2+3ND+6D$.

公式（2）~（4）中，$\boldsymbol{W}_c\in\mathbb{R}^{(N+1)\times d\times d}$，$\boldsymbol{U}_c\in\mathbb{R}^{(N+1)\times d\times 1}$，参数量为 $(N+1)\cdot d\cdot d+(N+1)\cdot d+(N+1)\cdot d=D^2/(N+1)+2D$.

由于公式（5）~（11）只存在于解码器部分，而编码器部分的长度一般要远大于解码器，故其中的参数量可忽略不计. 因此每个 TLSTM 单元的总空间复杂度为 $(3+1/(N+1))D^2+3ND+8D$.

模型的时间复杂度方面，对公式（1）来说，每个 TLSTM 单元中的每个门的计算复杂度为 $D^2+ND+2D$. 公式（2）~（4）中，计算复杂度为 $(N+1)\cdot d\cdot d+(N+1)\cdot d+2(N+1)\cdot d=D^2/(N+1)+3D.$ 其余公式中的参数量可忽略不计，故模型的总计算复杂度约为 $O(D^2+ND)$.

4　实验与分析

4.1　实验设置

为了验证本文所提模型的预测效果，本文选取了两个现实世界的真实时间序列数据集，各个数据集的基本信息如表 1 所示.

表 1 实验数据集基本信息

数据集	间隔	长度	序列数	协变量数	输入窗口	预测步长
Live600	天	31	623	54	10	7
Electricity	小时	26304	321	0	72	24

Live600. 该数据集来源于某直播流平台，选取了该平台 623 个主要主播在 2017 年 12 月的每日直播的被观看时长作为预测目标. 除了观看时长序列之外，该数据还含有丰富的协变量信息，比如由观看者提供的主播的标签、直播分类和主播职业等.

Electricity. 该数据集是由 Lai 等人[3]提供的公开数据集，包括 321 个客户端在 2012 年至 2014 年的每小时电力消耗量，共计 26304 个时间步. 该数据集中没有提供协变量.

在这些数据集上采取滑动窗口预测的方法进行采样，滑动窗口预测方法是一种在时间预测任务中常用的实验方法. 可以利用有限的数据构建多个训练集和测试集，提升性能检验效果的有效性. 给定一个总长度为 L_{all} 的数据集，按照一定的比例将数据集划分为训练集、验证集和测试集，长度分别为 L_{train}、L_{valid} 和 L_{test}. 采样训练集时，首先在训练集的头部选取出第一个滑动窗口，输入窗口时间步长为 T，输出预测时间步长为 τ. 随后将第一个窗口向后滑动 τ 步，选取出第二个窗口，以此不断类推，直到到达划分的训练集与验证集的界限. 采样验证集和测试集时，采用同样的滑动方法，这样就从原始数据集中采样出了分别属于训练集、验证集和测试集的若干个滑动窗口. 随后分别将每个滑动窗口作为模型的输入以及预测的目标.

以 Live600 数据集为例，选取输入窗口步长为 10，预测步长为 7 的滑动窗口. 即使用前 10 天的数据去预测后面一周的序列值. 对于 Electricity 数据集，如表 1 所示，使用前前三天（输入窗口 72）的数据去预测未来一天每个小时（预测步长 24 小时）的数据.

由于 Live600 数据集时间总长度有限，本文按照 80%（24 天）：16.7%（5 天）：6.7%（2 天）的比例将该数据集划分为训练集、验证集和测试集. 对于 electricity 数据集，按照 60%：20%：20%的比例进行划分.

对于所有输入的序列和协变量，采用最大值归一化的方法，将所有值归一到 $[-1,1]$ 的范围内：

$$y_t'^{i} = \frac{y_t^i}{max(|\boldsymbol{y}^i|)}$$

$$x_t'^{i,j} = \frac{x_t^{i,j}}{max(|\boldsymbol{x}^{i,j}|)} \tag{26}$$

除了原始数据集中自带的协变量外，还将时间特征作为额外的协变量输入. 以 Electricity 数据集为例，添加的协变量包括：年内第几月、月内第几天、周内第几天、是否是周末、天内第几小时以及是否工作时间（按照 9 至 17 时计算）. 对于以天为间隔的 Live600 数据集，则不考虑后面两个时间特征. 此外，模型还将上一时间步的序列值也作为一个额外的自回归协变量.

本文选取了两种常用的评估指标来衡量模型预测性能，分别是归一化均方根误差 NRMSE（Normalized Root Mean Squared Error）和归一化平均绝对误差 NMAE（Normalized Mean Absolute Error），其计算公式如下式（27）和（28）. 这两个指标是用来衡量预测值与真实值之间的误差的，因此越小越好.

$$NRMSE = \frac{\sqrt{\sum_{m=1}^{M}\sum_{t=T+1}^{T+\tau}(y_t^m - \hat{y}_t^m)^2}}{\sqrt{\sum_{m=1}^{M}\sum_{t=T+1}^{T+\tau}(y_t^m)^2}} \tag{27}$$

$$NMAE = \frac{\sum_{m=1}^{M}\sum_{t=T+1}^{T+\tau}\left|y_t^m - \hat{y}_t^m\right|}{\sum_{m=1}^{M}\sum_{t=T+1}^{T+\tau}\left|y_t^m\right|} \tag{28}$$

上式中 $\hat{y}_t^m$ 表示第 m 个时间序列第 t 个时间步的预测值，y_t^m 表示第 m 个时间序列第 t 个时间步的真实值.

NRMSE 和 NMAE 分别相当于均方根误差（Root Mean Squared Error，RMSE）和平均绝对误差（Mean Absolute Error，MAE）的缩小版本，因为随着序列变量值取值范围的变化，RMSE 和 MAE 的数值将变得没有参考意义，而 NRMSE 和 NMAE 能够将取值限定在一个可读的范围内.

所提模型以及其他基准模型的超参数均通过验证集进行了适当的调整. 对于 TED 模型，TLSTM 的隐状态维度 d 设置为 12，dropout 概率设置为 0.2. 对于全局特征提取部分，全局特征的维度 d_k 设定为 5（Live600 数据集）或者是 8（Electricity 数据集）.

本文使用 Python 3.7 语言，基于 PyTorch 框架编写了所提模型及实验的相关代码. 每个模型训练了 100 个轮次，并且选择在验证集上效果最好的对应轮次的模型来进行测试集的预测. 模型使用 Adam 来优化参数，学习率设置为 0.001.

4.2　对比模型

本文将所提出的模型与以下基准模型进行性能比较：

ARIMAX. ARIMAX 模型是差分整合滑动平均自回归（ARIMA）模型的考虑协变量的版本，该模型是最为经典的统计学方法之一，也是时间序列预测任务中最基本的线性方法.

DSSM[8]. 深度状态空间模型（Deep State Space Model，DSSM）使用 RNN 来编码时间序列，并生成线性状态空间模型的的参数，以此来预测未来序列的概率分布.

IMV-LSTM[26]. 该模型首次将 TLSTM 应用于单步时间序列预测任务中，并且依此给出可解释的预测结果.

TRMF[33]. 时序正则化矩阵分解（TRMF）模型将矩阵分解应用于高维时间序列预测任务中，使用线性自回归来建模矩阵分解时的时序依赖.

LSTNet[3]. 该模型使用CNN网络提取跨序列的全局特征，并通过RNN捕获全局特征中的时序信息. 该模型同时设计了一种跳跃 RNN 机制，以捕捉时间序列中的周期信息.

DSANet[5]. 该模型使用两个 CNN 网络来分别提取全局的和局部的时序特征，并且引入自注意力机制来建模跨序列的依赖.

DeepGLO[18]. 该模型采用了和 TRMF 类似的矩阵分解的思路，并使用时序卷积网络（TCN）代替线性自回归来建模矩阵分解中的时序依赖. 同时使用另一个 TCN 网络来预测未来的时间序列值，在每个时间步同时输入矩阵分解的预测值、协变量、上一时间步真实值.

TCN-MF[18]. 时序卷积网络矩阵分解（TCN-MF）模型是 DeepGLO 中的矩阵分解部分的直接预测结果.

以上基准模型大致可以分为三类：

TRMF、LSTNet、DASNet 和 TCN-MF 将所有的时间序列 $\boldsymbol{Y}$ 同时输入，而忽视了对于单个序列独立特征的建模，以及每个序列相关的协变量的作用. 并且，所有序列的预测结果 $\hat{\boldsymbol{Y}}$ 也被同时作为一个矩阵输出. 本文将此类模型称为全局模型（Global Models）.

ARIMAX、DSSM 和 IMV-LSTM 模型将每个序列 $\boldsymbol{y}^i$ 单独作为输入，且同时考虑了协变量 $\boldsymbol{X}^i$ 对于预测的作用，各个序列的预测结果 $\hat{\boldsymbol{y}}^i$ 也被单独给出. 这类模型本文称之为局部模型（Local Models），此类模型没有利用跨序列的全局特征.

DeepGLO 和本文所提出的 TEDGER 模型都能够同时利用局部的序列、协变量特征中的时序依赖，以及全局的跨序列特征. 本文将这类模型称为混合模型.

几个基准模型的参数设置如下. 在 DSSM 模型中，RNN 的 hidden state size 被设置成 128，RNN 层数为 2；IMV-LSTM 模型中，每个协变量对应的隐状态维度为 8；DeppGLO 模型中，通道数设置为 16、16、1，kernel 的大小为 2，全局模型的秩为 12；LSTNet 模型中，CNN 和 RNN 的隐状态维度为 32；DASNet 模型中，模型的隐状态维度为为 4，Key 和 Value 的维度为 8，head 数为 4，kernel 数为 4. dropout 概率统一设置为 0.2.

4.3　结果分析

4.3.1　与基准模型的对比

将本文提出的模型与其他基准模型在二个数据集上进行了预测效果的对比.

在 Live600 数据集上的实验对比结果如表 2 所示. 首先，在所有的全局模型中，LSTNet 和 DSANet 的效果较好，而基于矩阵分解的 TRMF 和 TCN-MF 效果一般. 其次，对于局部模型来说，ARIMAX 效果最差，这可能是由于 ARIMAX 模型过于简单，不能充分挖掘数据中的复杂关系；针对单步预测而设计的 IMV-LSTM 同样效果不佳，可能是由于其缺乏长距离预测的能力，不适用于多步预测任务，相比之下，DSSM 的预测效果比较好. 最后，在混合模型中，同样是融合了序列特征和全局特征的情况下，TEDGE 的预测效果远好于 DeepGLO，也超越了所有其他基准模型，本文提出的两种基于残差预测的改进模型 TEDGER 进一步提升了 NRMSE 预测指标，在这两种合成趋势序列的方法中，通过额外的 LSTM 编码器来计算权重的 TEDGER II 模型相比之下效果更佳.

表 2　LIVE600 数据集上的实验结果

	模型	NMAE	NRMSE
全局模型	TRMF	0.454	1.698
	TCN-MF	0.452	1.834
	LSTNet	0.383	1.428
	DSANet	0.411	1.338
局部模型	ARIMAX	0.485	1.631
	DSSM	0.299	0.782
	IMV-LSTM	0.416	1.333
混合模型	DeepGLO	0.411	1.784
	TEDGER I	0.238	**0.631**
	TEDGER II	**0.234**	0.641

在 Electricity 数据集上的实验对比结果如表 3 所示. 在全局模型中，TRMF 取得了最好的效果，LSTNet 和 DSANet 次之，说明即便是简单的矩阵分解模型，在特定的数据集上效果可能也优于复杂的全局神经网络模型. 在局部模型中，DSSM 的预测性能优于 ARIMAX 和 IMV-LSTM. 随着全局特征的融合，所提出的 TEDGE 模型在 NMAE 指标与 DSSM 相近的情况下，取得了更好的 NRMSE 指标. 在所提出的两种基于残差预测的改进模型中，TEDGER II 效果最佳，在 NMAE 和 NRMSE 两个指标上都进一步提升了预测性能.

表 3 Electricity 数据集上的实验结果

	模型	NMAE	NRMSE
全局模型	TRMF	0.088	0.774
	TCN-MF	0.279	0.972
	LSTNet	0.100	0.808
	DSANet	0.117	0.830
局部模型	ARIMAX	0.510	3.999
	DSSM	**0.061**	0.607
	IMV-LSTM	0.095	1.492
混合模型	DeepGLO	0.100	0.718
	TEDGER I	0.066	0.488
	TEDGER II	**0.061**	**0.486**

综上所述，首先，在所有的基准模型中，局部模型 DSSM 效果最佳. 其次，TEDGE 模型通过将矩阵分解得到的全局特征直接作为协变量引入，有效的融合了序列和协变量中的时序信息，以及跨序列的全局特征. 而本文提出的两种通过合成趋势序列进行残差预测的改进模型 TEDGER I 和 TEDGER II，能够更有效的利用全局特征，并实现了超越所有基准模型的优异的预测效果.

4.3.2 全局特征的影响

本文提出的 TEDGER 模型通过合成趋势序列提取了跨序列的全局信息，并将其与每个序列的时序信息进行了融合. 为了评估加入全局特征对于模型预测性能的影响，本文设计了相应的消融实验.

TEDGER 模型将潜在序列特征 $\boldsymbol{P}$ 和全局变化特征 $\boldsymbol{G}$ 合成得到趋势序列 $\boldsymbol{y}^{\text{pattern}}$. 如果移除模型中所有有关全局特征对应的模块，就得到了只考虑每个序列独有特征的局部模型，即将输入中的残差 $\boldsymbol{y}_{1:T}^{\text{residual}}$ 改为 $\boldsymbol{y}_{1:T}$，解码器部分也直接输出 $\boldsymbol{y}_{T+1:T+\tau}$，称该模型为 TED（Tensorized recurrent Encoder-Decoder）.

在 Live600 数据集上的对比实验结果如表 4 所示，在移除全局特征模块后，模型的预测性能发生了明显下降，说明了全局特征模块的有效性.

表 4 Live600 数据集上移除全局特征模块的实验

模型	NMAE	NRMSE
TEDGER I	0.238	0.631
TEDGER II	0.234	0.641
TED	0.282	0.871

4.3.3 输入窗口和预测步长的影响

为评估预测时输入窗口和预测步长的大小对于模型预测性能的影响，设计了变换输入窗口大小和预测步长的实验来进行评估. 由于 Live600 数据集上总时间步长有限，此实验选择在总长度较长的 Electricity 数据集上进行.

编码器阶段的输入窗口大小对于 TEDGERⅡ模型预测效果的影响如图 5 所示，在对 Electricity 数据集进行预测时，72 个时间步（3 天）是最佳的输入窗口. 一种可能的解释是，过短的输入窗口可能会遗漏重要的时序信息. 2 到 6 天的输入窗口对应的性能差别不大.

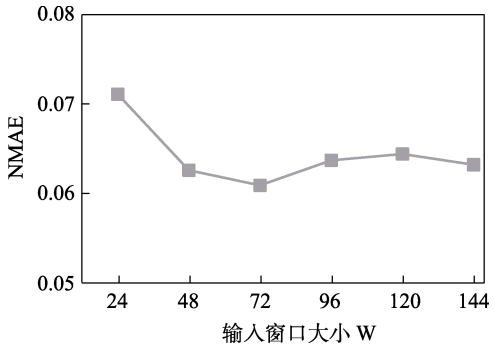

图 5 Electricity 数据集输入窗口大小对于预测的影响

另外，变换预测步长的实验结果如图 6 所示，总体上，模型预测的误差随着预测步长的增加而不

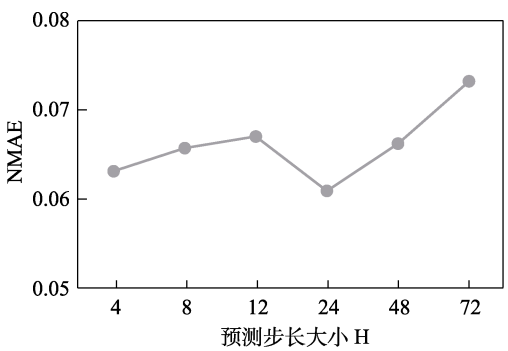

图 6 Electricity 数据集预测步对于预测效果的影响

断增大，步长越大越难以预测. 但是，在预测步长为 24（即 1 天）时，NMAE 误差较小，这可能是因为恰好预测后一天时，能够更好的利用模型中学习到的序列和全局信息.

4.4　可解释性分析

本文提出的模型的设计考虑到了可解释性，下面对 TEDGER II 模型的预测结果进行分析.

首先分析 Live600 数据集，其重要性排名前 10 的协变量如表 5 所示，标签开头的协变量代表主播是否有这个标签，如果有则为 1，反之为 0，在前 10 名的协变量中，有 8 个和标签相关，这说明对于预测模型，是否有相关的标签是一个重要的预测依据. 另外两个比较重要的协变量是“使用 iOS 系统”和“分类 0”，“分类 0”表示该主播是否属于唱歌分类.

表 5　Live600 数据集上重要性排名前 10 的协变量

排名	协变量	重要性
1	标签 14（“歌手”）	0.046
2	标签 29（“小鲜肉”）	0.042
3	标签 19（“闷骚”）	0.040
4	使用 iOS 系统	0.039
5	分类 0（“唱歌”）	0.038
6	标签 7（“帅哥”）	0.036
7	标签 31（“老师”）	0.035
8	标签 33（“才艺绝了”）	0.026
9	标签 5（“性感”）	0.026
10	标签 30（“文艺范”）	0.024

Live600 数据集的时序重要性如图 7 所示，可以看到，编码器阶段时间步的时序重要性呈现出快速上升的趋势，到第 6 个时间步，其平均重要程度达到最高，然后后续有所下降，但变化不大. 靠前的时间步相对作用小，靠后的作用更大，这是容易理解的. 由于 Live600 数据集的总长度过短，本文不再试图从中挖掘重要事件的信息.

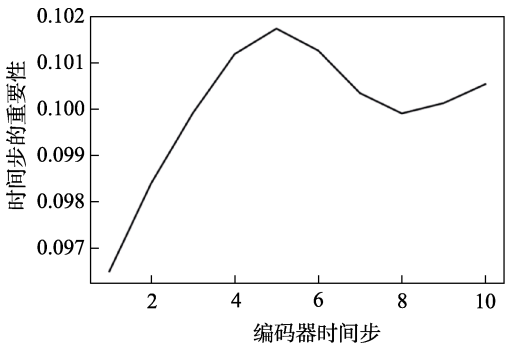

图 7　Live600 数据集上的平均时序重要性

其次分析 Electricity 数据集上的预测结果的可解释性. 该数据集上的平均协变量重要性如表 6 所示，可以看到，“是否是周末”对于未来预测贡献最大，占据了 18.42%的贡献，随后分别为“是否是工作时间”、“上一时刻自回归值”、“天内第几小时”以及“月内第几天”. 这说明虽然该数据集中没有内生的协变量，但设计加入的时间特征仍然具有意义.

表 6　Electricity 数据集上的协变量重要性

排名	协变量	重要性
1	是否是周末	0.184
2	是否是工作时间	0.169
3	上一时刻自回归值	0.145
4	天内第几小时	0.142
5	月内第几天	0.141
6	星期几	0.126
7	年内第几周	0.093

时间步的重要性如图 8 所示，在 Electricity 数据集上，编码器时间步的时序重要性呈现出明显的周期性特征，在 17~40 步和 41~64 步两个区间内，时序重要性曲线表现出来了高度的一致，而这一周期也恰好对应了该数据集中的一天，这表明本文所提模型能够挖掘数据集中的周期性信息. 两个峰值分别出现在第 28 和第 52 个时间步，相隔 24 小时，这说明在输入的历史窗口中，模型特别关注特定的一天中的某个小时输入的协变量.

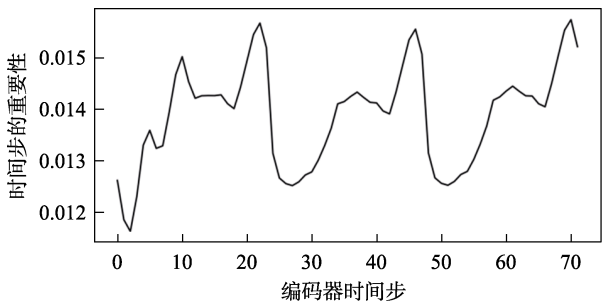

图 8　Electricity 数据集上的平均时序重要性

对 Electricity 数据集，本文选取了三个有代表性的序列样例来说明该方法的有效性，如图 9、10 和 11 所示，在每个样例序列中，上方的曲线代表的是序列在每个滑动窗口的平均值，而下方的曲线表示的是窗口的重要程度 $\gamma(w)$. 在第一组样例中，序列中的几次剧烈波动均能够被 $\gamma(w)$ 反映出来. 在后面的两组样例中，本文所提方法也捕捉到了序列值的突然急剧下降. 这说明本文所提方法能够正确挖

掘出序列中的重要事件，在序列值波动比较剧烈的窗口中，模型中设计的注意力机制能够捕捉到这种波动，反映了模型设计的有效性.

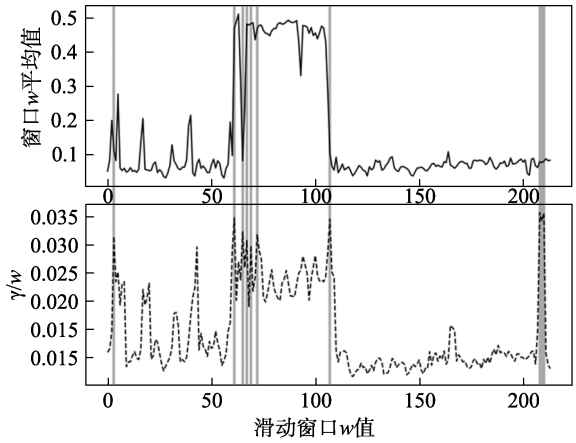

图 9 重要事件示例 1

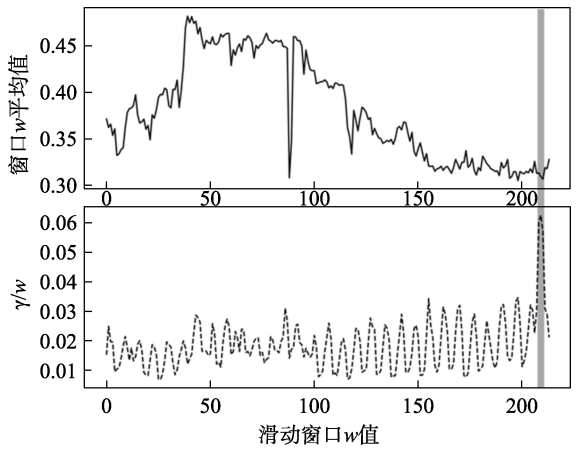

图 10 重要事件示例 2

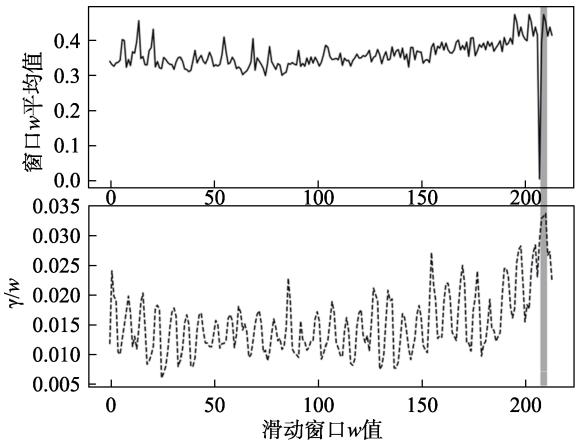

图 11 重要事件示例 3

5 结 论

本文研究多变量多步时间序列预测问题. 针对已有工作的不足，本文提出了一种新的多变量时间序列预测模型 TEDGER. 该模型基于张量循环神经网络的编码器提取时间序列的时序特征，解码器使用两种注意力机制，融合历史信息和不同协变量的影响，同时利用时序矩阵分解提取隐藏在多变量时间序列中的全局变化趋势特征. 在此基础上，设计了两种权重计算方法将全局特征重新组合成新的趋势序列，并将其作为未来预测的基准，在此基础上进行残差预测.

所提方法的性能通过实验进行了评估，与多个基准模型的性能进行了比较. 实验结果表明，本文提出的模型相比基准模型而言，能够提供更加精确的预测结果. 通过对注意力机制的权重进行分析，所提模型能够同时对预测结果进行一定程度的解释. 尽管本文在利用深度学习技术进行时间序列预测的研究上取得了一定进展，未来还可以在可解释性方面进行更深入的研究，比较不同可解释方法的有效性. 同时，可以进一步研究跨序列全局模式的其他提取方法，如结合聚类技术发现不同簇时间序列之间关系的方法，如何结合领域知识发现和利用全局和局部特征的方法，以及降低模型复杂度的方法等.

参 考 文 献

[1] Durbin J, Koopman S J. Time series analysis by state space methods. Second Edition. Oxford,UK: Oxford University Press, 2012

[2] Hyndman R, Koehler A B, Ord J K, et al. Forecasting with exponential smoothing: the state space approach. Berlin, Germany: Springer Science & Business Media, 2008

[3] Lai G, Chang W C, Yang Y, et al. Modeling long-and short-term temporal patterns with deep neural networks //Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. Ann Arbor, USA, 2018: 95-104

[4] Shih S Y, Sun F K, Lee H. Temporal pattern attention for multivariate time series forecasting. Machine Learning, 2019, 108(8): 1421-1441

[5] Huang S, Wang D, Wu X, et al. DSANet: Dual self-attention network for multivariate time series forecasting //Proceedings of the 28th ACM International Conference on Information and Knowledge Management. Beijing, China, 2019: 2129-2132

[6] Salinas D, Flunkert V, Gasthaus J, et al. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting, 2020, 36(3): 1181-1191

[7] Qin Y, Song D, Cheng H, et al. A dual-stage attention-based recurrent neural network for time series prediction//Proceedings of the 26th International Joint Conference on Artificial Intelligence. Melbourne, Australia, 2017: 2627-2633

[8] Rangapuram S S, Seeger M, Gasthaus J, et al. Deep state space models for time series forecasting //Proceedings of the 31st

International Conference on Neural Information Processing Systems. Montr, Canada, 2018: 7796-7805

[9] Fan C, Zhang Y, Pan Y, et al. Multi-horizon time series forecasting with temporal attention learning //Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Anchorage, USA, 2019: 2527-2535

[10] Li S, Jin X, Xuan Y, et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting //Proceedings of the 32nd International Conference on Neural Information Processing Systems. Vancouver, Canada, 2019: 5243-5253

[11] Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H, Zhang W. Informer: Beyond efficient transformer for long sequence time-series forecasting //Proceedings of the 35th AAAI Conference on Artificial Intelligence. Virtual, 2021: 11106-11115

[12] Feng, Z. K., & Niu, W. J. (2021). Hybrid artificial neural network and cooperation search algorithm for nonlinear river flow time series forecasting in humid and semi-humid regions. Knowledge-Based Systems, 2021, 211: 106580

[13] Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting. 2021, 37(4): 1748-1764

[14] Wu Minghui, Zhang Guangjie, Jin Canghong. Time series prediction model based on multimodal information fusion. Journal of Computer Applications, Web first released. Oct. 15, 2021 (in Chinese)
(吴明晖, 张广洁, 金苍宏.基于多模态信息融合的时间序列预测模型计算机应用, 网络首发时间：2021-10-15) https://kns.cnki.net/kcms/detail/51.1307.TP.20211014.1357.014.html

[15] Tang Tiantian, Zhou Wei. Research on commodity sales forecast oriented on deep learning. Journal of Chongqing University of Technology(Natural Science). Web first released Sep. 15, 2021 (in Chinese)
(唐甜甜, 周伟. 面向深度学习的商品销售额预测研究. 重庆理工大学学报 (自然科学). 网络首发时间: 2021-09-15) https://kns.cnki.net/kcms/detail/50.1205.T.20210914.1808.012.html.

[16] Sun Si-Yu, Zhang Biao-Biao, Wu Jun-Hong, Ma Shi-Qiang, Ren Jia. time series forecasting combining temporal convolution, Residual Structure and Attention Mechanism. Computer Systems & Applications, 2021, 30(9): 145−151 (in Chinese)
(孙思宇, 张标标, 吴俊宏, 马仕强, 任佳. 融合时域卷积、残差结构和注意力机制的时序预测. 计算机系统应用, 2021, 30(09): 145-151)

[17] Du Xiwang, Zhao Xing, Li Liang. Research on short-term prediction of inbound passenger flow of rail transit based on LSTM. Journal of Guizhou University (Natural Sciences). 2021, 38(5): 109-118 (in Chinese)
(杜希旺, 赵星, 李亮. 基于 LSTM 的轨道交通进站客流短时预测研究. 贵州大学学报(自然科学版), 2021, 38(05): 109-118

[18] Sen R, Yu H F, Dhillon I S. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting//Proceedings of the 32nd International Conference on Neural Information Processing Systems. Vancouver, Canada, 2019: 4837-4846

[19] Yang Z, Keung J, Kabir M A, Yu X, Tang Y, Zhang M, Feng S. AComNN: Attention enhanced compound neural network for financial time-series forecasting with cross-regional features. Applied Soft Computing, 2021, 111: 107649

[20] Cheng D, Yang F, Xiang S, Liu J. Financial time series forecasting with multi-modality graph neural network. Pattern Recognition, 2022, 121: 108218

[21] Hu Hexuan, Sui Huachao, Hu Qiang, Zhang Ye, Hu Zhenyun, Ma Nengwu. Runoff forecast model based on graph neural network and two-order attention mechanism. Journal of Computer Applications. DOI: 10.11772/j.issn.1001-9081.2021050829. (in Chinese)
(胡鹤轩, 隋华超, 胡强, 张晔, 胡震云, 马能武. 基于图神经网络与双阶注意力机制的径流预报模型. 计算机应用, DOI: 10.11772/j.issn.1001-9081.2021050829)

[22] Ye, R., & Dai, Q. Implementing transfer learning across different datasets for time series forecasting. Pattern Recognition, 2021, 109: 107617

[23] Godahewa, R., Bandara, K., Webb, G. I., Smyl, S., & Bergmeir, C. Ensembles of localised models for time series forecasting. Knowledge-Based Systems, 2021, 233: 107518

[24] Zhang, S., Chen, Y., Zhang, W., & Feng, R. A novel ensemble deep learning model with dynamic error correction and multi-objective ensemble pruning for time series forecasting. Information Sciences, 2021, 544: 427-445

[25] Lim B, Zohren S. Time Series Forecasting with Deep Learning: A Survey. arXiv preprint arXiv:2004.13408, 2020

[26] Guo T, Lin T, Antulov-Fantulin N. Exploring interpretable LSTM neural networks over multi-variable data //Proceedings of the International Conference on Machine Learning. Long Beach, USA, 2019: 2494-2504

[27] Pantiskas L, Verstoep K, Bal H. Interpretable multivariate time series forecasting with temporal attention convolutional neural networks//Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI). Canberra, Australia, 2020: 1687-1694

[28] Lim B, Arik S O, Loeff N, et al. Temporal fusion transformers for interpretable multi-horizon time series forecasting. arXiv preprint arXiv:1912.09363, 2019

[29] Assaf R, Giurgiu I, Bagehorn F, et al. MTEX-CNN: Multivariate time series explanations for predictions with convolutional neural networks//Proceedings of the 2019 IEEE International Conference on Data Mining. Beijing, China, 2019: 952-957

[30] Ismail A, Gunady M, Bravo H, et al. Benchmarking deep learning interpretability in time series predictions//Proceedings of the 33rd International Conference on Neural Information Processing Systems. Virtual, 2020: 6441-6452

[31] Li L, Yan J, Yang X, et al. Learning interpretable deep state space model for probabilistic time series forecasting //Proceedings of the 28th International Joint Conference on Artificial Intelligence. Macao, China, 2019: 2901-2908

[32] Oreshkin B N, Carpov D, Chapados N, et al. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting// Proceedings of the International Conference on Learning Representations (ICLR). New Orleans, USA, 2019: 1-31

[33] Yu H F, Rao N, Dhillon I S. Temporal regularized matrix factorization for high-dimensional time series prediction// Proceedings of the 29th International Conference on Neural Information Processing Systems. Barcelona Spain, 2016: 847-855

[34] Li Z, He J, Liu H, et al. combining global and sequential patterns for multivariate time series forecasting//IEEE International Conference on Big Data (Big Data 2020). Atlanta, USA, 2020: 180-187

LI Zhao-Xi, master. His research interests includes data mining, time series analysis and computer vision.

LIU Hong-Yan, Ph.D., professor. Her research interests include artificial intelligence, business intelligence, recommender systems, financial technology and computer vision.

Background

Time series exist in many areas of our daily life and work. For example, the daily sales of goods, hourly weather conditions, daily stock prices, daily subway traffic, etc. are all time series data. Time series forecasting can help us understand how things evolve over time. Prediction can give us chance to intervene what might happen next. Extensive studies have been conducted for accurate time series prediction. Many statistical models have been widely used in the past. In recent years, many deep learning models are proposed for time series forecasting, which show better accuracy than statistical models. However, how to utilize both the local and global patterns hidden in multivariate time series to further improve accuracy and provide interpretability of the model are still open questions. In this paper, we propose a new deep learning based multivariate time series forecasting model named TEDGER. TEDGER can capture both sequential patterns hidden in individual time series and global trends hidden across multivariate times series. Both sequential pattern and global trend are integrated to make residual prediction for final time series forecasting. We conducted experimental study to evaluate the performance of our proposed model on two real world datasets. We compare our model with eight benchmark models including both statistical and deep learning models. Results show that our proposed model shows better performance than any of other benchmark models. Meanwhile, our model can provide the importance of input variable based on attention mechanism.

This study was supported by the National Social Science Major Program with grant number 20&ZD161. This project aims to conduct research on methods to make good use of data. This study contributes to this project on methods of time series prediction with interpretability.