第 45 卷 第 3 期 计 算 机 学 报 Vol. 45 No. 3
2022 年 3 月 CHINESE JOURNAL OF COMPUTERS Mar. 2022

一种基于自演化预训练的多变量时间序列预测算法

万 晨[1),2)] 李文中[1),2)] 丁望祥[1),2)] 张治杰[1),2)] 叶保留[1),2)] 陆桑璐[1),2)]

[1)](南京大学计算机软件新技术国家重点实验室 南京 210023)

[2)](南京大学计算机科学与技术系 南京 210023)

摘 要 时间序列预测是典型的时间序列分析任务，对于辅助决策、资源配置、提前采取止损措施等方面有重要意义，在包括电力、气象、交通、商业等领域有广泛应用. 近年来，时间序列预测算法一直是机器学习的热门研究领域，其中多变量时间序列预测是一个具有挑战性的任务. 本文研究多变量时间序列预测的局部变量预测精度问题，即多变量预测需要在提升整体预测性能的同时保证局部单变量的预测精度. 针对现有多变量时间序列预测算法不能保障局部变量预测精度的局限性，我们设计并实现了一种基于自演化预训练的多变量时间序列预测算法 SEPNets. 基于预训练的思想，SEPNets 首先构建和训练单变量时间序列模型作为后续建模的基准. 然后，通过拓展时序卷积网络和长短记忆（LSTM）单元来建模变量间复杂的时序依赖关系. 通过将预训练模型和拓展模型进行融合再训练，SEPNets 可以保障多变量时间序列预测的局部变量预测精度，并提升总体的预测性能. 我们在 5 个真实数据集上对所提模型进行性能评估. 实验结果表明，本文提出的 SEPNets 算法比现有算法获得相对最高的预测精度，同时在保障局部变量预测精度上具有更好的性能.

关键词 时间序列预测；多变量时间序列；机器学习；神经网络；预训练

中图法分类号 TP391 **DOI 号** 10.11897/SP.J.1016.2022.00513

A Multivariate Time Series Forecasting Algorithm Based on Self-Evolution and Pre-training

WAN Chen[1),2)] LI Wen-Zhong[1),2)] DING Wang-Xiang[1),2)] ZHANG Zhi-Jie[1),2)]
YE Bao-Liu[1),2)] LU Sang-Lu[1),2)]

[1)](*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing,* 210023)

[2)](*Department of Computer Science, Nanjing University,* 210023)

Abstract Time series forecasting is a typical time series analysis task, which is of great significance for assisting decision-making, resource allocation, and taking stop loss measures in advance. It is widely used in fields including power, weather, transportation, and business. In recent years, time series forecasting algorithms based on machine learning have been a hot research field, among which multivariate time series forecasting is a challenging task. This paper focuses on the local variable forecasting accuracy of multivariate time series forecasting, that is, multivariate forecasting needs to improve the overall forecasting performance while ensuring the forecasting accuracy of local univariate forecastings. In view of the above problems, this paper analyzes the limitations of the multivariate time series forecasting algorithm. The multivariate time series forecasting algorithm SEPNets based on self-evolution pretraining is designed

收稿日期：2020-05-19；在线发布日期：2021-07-29. 本课题得到本课题得到国家重点研发计划（2018YFB1004704）、国家自然科学基金（61972196, 61672278, 61832008, 61832005）、江苏省重点研发计划项目（BE2018116, BE2017152）、软件新技术与产业化协同创新中心、中德社会计算研究所资助. **万 晨**，硕士研究生，主要研究领域为时间序列分析、深度学习. E-mail: mf1733062 @smail.nju.edu.cn. **李文中**（**通信作者**），博士，教授，中国计算机学会（CCF）会员，主要研究领域为分布式计算、社交网络分析、深度学习. E-mail: lwz@nju.edu.cn. **丁望祥**，硕士研究生，主要研究领域为时间序列分析、深度学习. **张治杰**，硕士研究生，主要研究领域为时间序列分析、深度学习. **叶保留**（**通信作者**），博士，教授，中国计算机学会（CCF）会员，主要研究领域为分布计算与并行处理. E-mail: yebl@nju.edu.cn. **陆桑璐**（**通信作者**），博士，教授，中国计算机学会（CCF）会员，主要研究领域为分布计算与并行处理. E-mail: sanglu@nju.edu.cn.

and implemented. Inspired by the pre-trained model, SEPNets first constructs and trains a univariate time series model as a benchmark for subsequent modeling; then expands the complex temporal dependence between the time-series convolutional network (1-dimensional convolution) and the long and short memory (LSTM) unit modeling variables Relationship; using the former as the latter pre-training model for fusion and retraining, SEPNets solves the problem of local variable forecasting accuracy for multivariate time series forecasting. The experimental results show that the SEPNets algorithm proposed in this paper ensures the forecasting accuracy of local variables to a certain extent while obtaining the relatively highest forecasting accuracy.

Keywords time series forecasting; multivariate time series; machine learning; neural networks; pre-train

1 引 言

1.1 研究背景

近年来，时间序列挖掘成为了备受关注的研究领域，其研究问题包括时间序列的分类、预测、关联、聚类、回归、摘要、异常检测等. 时间序列预测 (time series forecasting, TSF) 的主要目的是基于历史观测的时间序列数据，分析序列的时序模式(包括趋势、周期性、波动性等)，建立合适的序列模型，然后使用该模型为对应的时间序列生成未知的未来数据，即进行预测. TSF 是时间序列建模中通过理解过去来预测未来的算法[1].

采用时间序列预测方法、获得未来时刻的预测值，对于辅助决策、优化资源分配、提前采取止损措施等具有基础且重要的意义. 在电力、气象、交通、商业等领域，时间序列预测算法有着广泛应用. 例如，亚马逊等大型商业组织依据需求信息及时做出正确的决策，在不断变化的商业环境中增强商业竞争优势、降低成本、提高商业利润，所以对整个供应链中的商品需求量、供应量、价格等进行预测具有必要性. 此外，实时运行的信息系统产生了大量的时序数据，包括系统性能监测数据等，对这些监测数据进行预测是很多应用的需求. 例如预测网络流量用于资源规划与异常发现[2]，预测数据中心的磁盘故障以提前替换[3]等. 因此预测系统性能数据的趋势可以为现代信息系统的日常运维工作提供重要的决策依据.

为建立有效的时间序列预测模型以提高预测性能，研究人员做出了许多努力. 长期以来，线性统计类算法（例如 AR[4]、ARIMA[5]）对 TSF 领域产生了影响. 由于线性模型在很多实际应用中缺乏适用性[6]，非线性时间序列分析和预测被提出并逐渐应用[7].

除了经典的统计方法模型，机器学习模型已经成为时间序列预测领域的主流技术[6]. 研究人员将线性回归模型、支持向量机[8]、决策树等机器学习模型应用于时间序列分析. 基于机器学习的时间序列预测算法把时序预测作为监督学习任务，其中输入属性为历史观测的时间序列数据，输出标签为(训练集上的）未来数据. 机器学习模型以数据驱动的方式，学习序列特征，建模历史数据和未来数据之间的映射关系. 在实际应用中，基于机器学习的时间序列预测算法表现优于基于统计的算法[9]. 为了改善时间序列预测算法，人们尝试在模型复杂度和序列特征表示方面进行突破. 以神经网络为代表的深度学习算法是机器学习研究的新兴领域[10]，其逐层构建与处理的机制具有显著的模型内特征转换和足够的模型复杂度的优势[11]. 最近在时间序列建模预测中采用深度神经网络算法成为了研究热点，特别是在大数据和复杂任务的场景. 无论在对模型的理论理解和在模型优化方面，都取得了令人瞩目的成果. 例如，基于递归神经网络的算法通过使用有状态模型建模时间序列数据的序列特征，获得了出色的预测性能[2,12].

尽管基于机器学习的时间序列预测问题已被广泛研究，多变量时间序列预测仍然是具有挑战性的研究问题. 在多变量时间序列预测任务中，相关的多个时间序列被并行地统一建模和预测. 例如，预测多个家庭的能源消耗、数据中心中多台服务器的负载、大型零售商提供的所有产品的需求等. 在这些情况下，可以利用时间序列间的相似性，建模多变量时间序列依赖关系，以提高整体的预测精度. 捕捉复杂时序模式与变量间依赖关系是多变量时间序列预测研究的重点和实现精准预测模型的关键[13].

1.2 研究动因

同一系统中，多个序列组成的多变量时间序列

本质上具有相关性. 不同于单变量预测任务，利用变量间的相关性进行建模，可以提高整体预测精度. 多变量序列预测有着广泛的应用场景，为了获得较好的预测精度，需要有效建模复杂时序模式与变量间的依赖关系. 代表性多变量时间序列预测算法包括时序分析类统计算法 VAR[14]，基于 LSTM 网络的时序建模[15,16]，和融合卷积神经网络和递归神经网络的 CNN-GRU-Input(All)-Att[17]和 LSTNets[18]算法等. 然而，现有的多变量时间序列预测算法主要关注整体预测目标优化，会带来局部变量预测精度损失的问题. 例如，在多变量预测的回归任务中，现有算法以所有变量的整体预测误差加权和为最小化目标,通过优化该目标提高整体预测精度. 然而，整体优化目标可能忽略局部变量的预测精度损失，造成牺牲局部变量的精度来提升整体预测精度的问题. 在实际应用中，对于一个复杂的多变量系统，可能会存在某些非常重要的指标，如果为了提高整体的预测精度而牺牲了重要指标的预测精度，可能会带来重大损失. 例如，在亚马逊等电商系统中记录了许多指标数据，如商品需求量、供应量、价格、评星等，其中一旦如商品需求量、价格等重要局部指标的预测精度得不到保障，就会对企业的决策造成较大的影响. 此外，在性能监测系统中需要监测多种数据，比如 CPU 利用率、内存利用率、响应时间、吞吐量等. 如果不能保障响应时间和吞吐量等重要指标的预测精度，可能会导致运维人员无法针对这些重要的指标提前合理地采取止损措施. 因此，多变量时间序列预测算法在优化整体预测精度的同时，需要保证局部变量的预测精度不会造成损失.

我们通过实验评估来展示局部变量预测精度的损失问题. 图 1 给出对部分多变量时间序列预测算法在 Electricity 数据集（具体描述见第 4 节）上的精度评估结果. 横坐标表示四种不同的多变量预测算法，分别为 VAR[14]、LSTM[19]、LSTNets[18]、Expected（表示理想算法）；纵坐标表示对应算法在所有变量上的精度. 其中，对每个变量单独采用单变量预测算法 AR 进行建模和预测时，其平均精度记为 P_0.当采用多变量时间序列预测算法后，四个算法的整体平均预测精度分别为 P_1,P_2,P_3,P_4.对每种算法，蓝色条形表示预测精度下降的局部变量的比例，橘色条形表示预测精度上升或不变的局部变量的比例. 如图所示，尽管各种算法的整体平均精度都高于 P_0，但是 VAR 算法有大部分局部变量的预测精度降低了，而 LSTM 和 LSTNets 算法也有很大比例局部变量的预测精度低于单变量预测算法. 究其原因，已有算法以提升多变量整体预测精度为优化目标，并不能保障局部变量的精度损失.

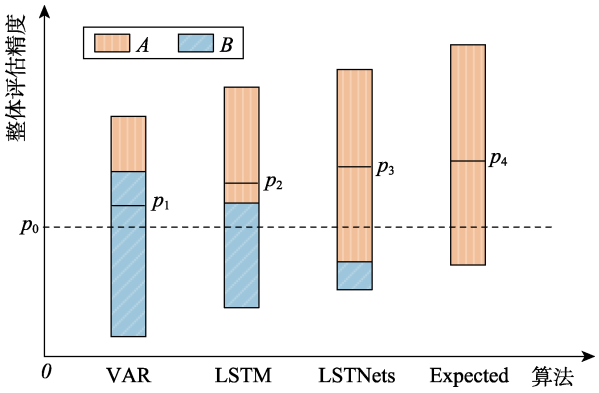

图 1　多变量时间序列预测算法的精度估计（Electricity 数据集）

将单变量序列融入多变量序列预测中，其目的是获得自身预测精度的提升，或者至少不比单序列自身预测精度差. 因此，理想的多变量时间序列预测算法应如图 1 的 Expected 算法所示，在提升整体预测精度的同时，保障局部变量的预测精度不低于单变量预测算法的性能. 为此，需要研制具有局部变量性能保障的多变量时间序列预测算法.

1.3　解决思路与贡献

为了解决多变量时间序列预测的局部变量精度损失问题，针对任务的复杂性以及多变量和单变量优化目标的差异性，我们采用深度学习模型进行多时间序列建模，并提出基于自演化预训练的预测算法. 在以上问题中，单序列预测模型的精度是对局部变量预测性能保障的基准. 因此，在多变量时间序列模型的整体优化中如何有效地融合单序列预测模型，是解决问题的关键. 预训练（pretrain）策略可以在类似任务的已训练模型的基础上，构建当前任务的模型，并继续训练. 由于当前模型采用了已训练模型表示（包括网络结构与权值），等价于迁移已有的知识表示来实现后续任务（downstream task）. 预训练机制已被用于计算机视觉（CV）与自然语言处理(NLP）领域，在节省资源的同时能获得良好的性能[20].

基于预训练策略，结合多变量序列预测背景，本文首先设计单序列基准模型，并在每个单序列上进行单独训练，作为后续整体任务的“基准知识”，其蕴含了单变量自身演化（Self-Evolution）模式；然后，设计多层次 CNN-RNN 级联神经网络单元作

为多变量建模模块，先后对多变量间相互依赖（Inter-Dependency）关系进行局部空域信息（local spatio）建模和时序（temporal）信息建模；最后，把已训练完成的单变量模型作为预训练模块，与多变量建模模块的输出一一对应融合，从而利用已训练完成的单序列模型来优化整体模型的性能. 该思路能够在多变量模型的整体优化中有效地融合已训练完成的单序列模型，解决局部单变量的预测精度比对应的单序列模型的预测精度差的问题.

本文的主要贡献总结如下：

• 本文提出并研究了多变量时间序列预测的局部变量预测精度问题. 通过实验观测，评估已有基于机器学习的多变量时间序列预测算法的局限性，这些算法在优化整体预测精度时，可能造成局部单变量的预测精度损失.

• 为保障局部变量的预测精度，提出了基于自演化预训练的多变量时间序列预测算法. 基于预训练策略与卷积网络、长短记忆网络深度序列模型，结合单变量自演化信息与多变量依赖关系信息进行建模，实现了多变量时间序列预测的优化算法，同时兼顾了整体预测精度和局部单变量的预测精度.

• 基于五个真实数据集进行对比实验，评估了所提算法的有效性. 实验结果表明，基于自演化预训练的多变量时序预测算法，具有较好的整体预测精度，对于局部单变量的预测精度保障性方面远优于现有的多变量时间序列预测算法.

本文的组织结构如下. 第 2 章介绍相关工作. 第 3 章介绍基于自演化预训练的多变量时间序列预测模型和算法. 第 4 章对所提算法进行性能评估和分析. 第 5 章对全文进行总结.

2 相关工作

多变量时间序列预测方法利用多个时间序列间的相关性，并以此建模提高整体预测性能. 预测多变量时间序列的关键是如何从时间序列中精确捕捉复杂的时序模式与变量间依赖关系[13]. 向量自回归模型（Vector Autoregression，VAR）是常用的典型统计时间分析类的多变量时间序列分析算法. VAR 将多变量时间序列看作多维随机过程，在单变量 AR 模型的基础上对变量依赖关系进行线性假设. VAR 描述的随机过程是线性的，用于捕捉多变量序列间常见的线性模式，适用于平稳序列建模. 向量差分整合移动平均自回归模型（Vector ARIMA, VARIMA）是 ARIMA 算法的多变量扩展. 该模型能够以随机过程描述更多的信息，但一般结构复杂优化困难，其实用效果不如 VAR（VAR 拟合前可进行简单的差分平稳化）. 高斯随机过程（Gaussian Proccess, GP）是另一类多变量时间序列分析的统计算法[21]. 该模型先验假设多变量时间序列之间服从多维高斯分布，其由均值向量与协方差矩阵共同决定. 尽管这些统计类时间序列分析模型在多个领域得到应用，但它们假设了线性的时序模式，并且不适合建模长期依赖关系.

由于深度学习模型在复杂度和序列特征表示方面表现优秀，目前多变量时间序列预测算法越来越多地采用深度神经网络进行构建，以捕捉复杂的时序模式与变量间依赖关系[6]. 全连接神经网络（Nerual Networks, NNs）最早用于对时间序列进行建模预测[22]. 层间全连接的神经网络能够建模非线性等时序关系，并且十分灵活. 受到序列建模与权值共享思路的启发，循环神经网络（Recurrent Neural Network, RNN）被提出. RNN 为每个时刻增加了状态向量 $\boldsymbol{h}_t$，并且增加了层内前一时刻状态到当前时刻状态的有向连接，这有效建模了序列信息，即当前时刻的状态不仅取决于当前输入，而且依赖于之前时刻的信息，而 $\boldsymbol{h}_t$ 是时刻 t 前所有信息的表示. 基于递归神经网络的算法通过使用有状态模型建模时间序列数据的序列特征，并获得较好的预测性能[2] [12].

为了解决 RNN 中梯度消失问题[23]带来的训练困难以及长时段依赖关系的捕捉，人们提出了基于门机制的 Long-Short Term Memory（LSTM）算法[19]. LSTM 是一种性能出色的深度序列模型，常用作大数据、复杂任务场景下的序列建模[24,25]，或者作为基本单元与其他算法融合进行序列建模[26]. LSTM 还有多种变体，诸如简化门控制机制的 GRU[27]、关注序列相位信息的 phi-LSTM[28]、显示地跨时间步连接的 skip-LSTM[29]等.

为了获得多变量时间序列的有效特征表示，研究人员采用自编码器（Auto-Encoder, AE）对多变量时间序列进行编码. Wang 等人提出了一种基于优化的时序卷积网络并带有堆叠式自动编码器的预测算法，可以有效表达多变量时间序列中时序关系的复杂性和非线性[30]. 基于编解码架构，Qin 等人提出基于两阶段注意力（Attention）机制的循环神经网络多变量预测算法 DA-RNN[31]，分别在时间序列编码阶段与解码预测阶段加入注意力机制，自适应的选择相关的输入特征表示，有利于捕捉长时间的时序信息与依赖关系. Huang 等人提出了一种双重自注意力网络（DSANet），用于高效的多元时间序列

预测，适用于动态周期或非周期序列的预测[32].

除了典型的深度序列模型 RNN，人们发现诸如卷积神经网络(CNN)等也可以作为序列建模单元[33]. 为了建模时间序列之间存在的强相关性，Hu 等人提出了一种基于时序卷积网络的多变量时间序列预测模型[34]. Ong 等人提出了一种时间张量转换网络（Temporal Tensor Transformation Network）[12]深度学习架构，以解决现有算法的强统计假设、高维数值、手动特征工程和可伸缩性等问题. 另外，Assaf 等人证明了 CNN 深度神经网络不仅可以用于多元时间序列数据预测，还可以用于解释这些预测[35].

研究人员试图结合时间序列分析算法与深度神经网络来构建混合模型，从而利用两者的优势来提高性能. 对于多变量时间序列建模通常存在的显式线性相关性成分和非线性成分信息，Goel 等人提出了基于残差学习的混合模型 R2N2[36]. 针对多变量时间序列间短期与长期依赖关系的混合问题，Lai 等人提出一种长期和短期时间序列网络深度学习框架[18]，使用卷积神经网络（CNN）和递归神经网络（RNN）来提取变量之间的短期局部依赖模式，拓展 skip-LSTM 捕获时间序列趋势的长期模式，进行模型构建. Thongniran 等人提出了将注意力机制（attention）加入 CNN 与 RNN 结合的模型中来进行多变量时间序列的建模[17].

近年来，国内学者也对多变量时间序列进行了大量研究. 张鹏程等人提出了一种基于多元时间序列的 QoS 预测方法 MulA-LMRBF[37]，通过网络开源数据和自测数据的实验结果表明，该方法与传统方法相比有较好预测效果，更适合动态多步预测. 欧阳红兵等人提出将小波分析与纳入时间序列依赖特征的长短期记忆（LSTM）神经网络相结合，构建时间序列数据预测模型[38]. 王双成等人在基于高斯函数估计属性密度的基础上，结合转换数据集构建、回归变量的离散化、类变量的数量化、属性联合密度的分解计算等，建立用于时间序列预测的具有超父结点贝叶斯网络回归模型[39]. 此外，李文海等人提出一种基于 DTW 距离的相似样本度量方法[40]，通过误差调谐函数对 SVM 的核函数进行调整来提升预测的精度.

总之，已有基于机器学习的时间序列预测算法关注于如何有效的捕获复杂时序模式和变量间关系，并建立整体优化预测模型. 在整体模型优化过程中，往往以所有变量的预测误差和（或者加权和）为最小化目标. 正如本文引言部分分析，现有算法在优化全部变量的整体预测精度时，不保证每个单变量时间序列的预测精度，可能导致局部变量预测精度损失的问题. 本文提出的自演化预训练的多变量时间序列预测算法，可以在提升整体预测精度的同时，保障单变量时间序列的预测精度不会降低.

3　自演化预训练的多变量时间序列预测算法

3.1　问题建模

本文将多变量时间序列 $\boldsymbol{x}_t \in R^N$ 形式化描述如下：

$$\begin{aligned}\boldsymbol{x}_t &= process\left(\boldsymbol{x}_{t-T:t-1}\right)\\ &= merge\left(\left[SE\left(\boldsymbol{x}_{t-T:t-1}\right), ID\left(\boldsymbol{x}_{t-T:t-1}\right)\right]\right)+\varepsilon_t\end{aligned} \tag{1}$$

其中 N 为多变量时序数据的变量数，T 为通过自相关函数 ACF 来评估时间序列的周期性而得到的历史输入序列长度，$\boldsymbol{x}_{t-T}$ 表示时刻 t 前 T 长度的历史时间序列. $Process(\cdot)$ 把多变量序列抽象为随机过程，而 $merge(\cdot)$ 进一步假设这样的随机过程是两部分的融合结果；$SE(\cdot)$ 表示单变量的自演化模式（self-evolution），$ID(\cdot)$ 表示多变量间的依赖关系（inter-dependency）；ε_t 为多维随机噪声.

本文针对 $SE(\cdot)$ 和 $ID(\cdot)$，进行建模、构建学习模块，然后在 $merge(\cdot)$ 过程进行融合训练，具体介绍如下.

3.2　模型框架

本文提出基于自演化预训练的神经网络（Self-Evolution Pretrained Neural Network，简称为 SEP-Nets）模型，整体框架如图 2 所示.

该框架将具有 N 个变量数的时间序列 $\left[x^{(1)},\cdots,x^{(N)}\right]$ 作为输入，经过两个模块的处理融合后，输出所有变量序列下一时间步的预测值向量 $\boldsymbol{x}_t$. 具体步骤如下.

第一步，将 N 个单序列 $\left[x^{(1)},...,x^{(N)}\right]$ 输入到自演化预训练模型 $SE(\cdot)$ 中，进行单序列建模、预测，形成对应输出 $\boldsymbol{x}_t^{(se)}$；第二步，将 N 个单序列组成的二维张量 $\left[x^{(1)},\cdots,x^{(N)}\right]$ 先进行大小不同的卷积操作，提取变量间依赖关系信息得到多个特征图 (feature map)，然后输入到长短记忆网络（LSTM）中进行时序建模，这样的多变量建模的网络组件（Nets）形成对应输出 $\boldsymbol{x}_t^{(id)}$；最后，将前两步的 $\boldsymbol{x}_t^{(se)}$ 和 $\boldsymbol{x}_t^{(id)}$ 进行向量元素的一一对应相加融合，构建出多变量预测输出 $\boldsymbol{x}_t$. 以下介绍各部分的细节.

3.3　自演化预训练模型

自演化预训练建模任务是单序列预测任务，为

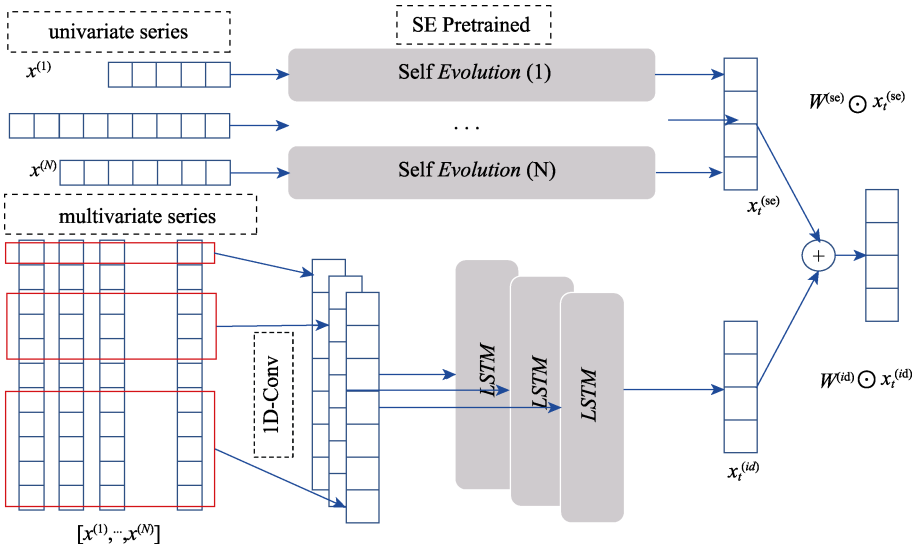

图 2 SEPNets 框架图

整体多变量任务建模、训练提供基准，并可以作为局部单变量预测精度的评估标准. 在大量各异的单序列预测任务的情况下，需要平衡预测精度和模型复杂度，应对各时序模式各异的单序列，算法必须具有广泛的适用性.

从简单、有效的时序分析的统计类算法出发，本文提出基于差分特征构造的线性自回归模型（enhanced autoregression, EAR）. 首先，建立线性自回归模型. 对于第 i 个序列有 $x_t^{(i)}=\sum_{j=1}^{T}w_j^{(i)}x_{t-j}^{(i)}$，其中 $w_j^{(i)}$ 为线性权值. 借鉴差分平稳化的思路，线性回归模型构造“多阶差分特征”. 考虑差分阶数 $q=1$ 时，有 $(x_t^{(i)}-x_{t-1}^{(i)})=\sum_{j=1}^{T}w_j^{(i)}(x_{t-j}^{(i)}-x_{t-j-1}^{(i)})$ 即 $x_t^{(i)}=x_{t-1}^{(i)}+w_1^{(i)}x_{t-1}^{(i)}+\sum_{j=2}^{T}(w_j^{(i)}-w_{j-1}^{(i)})x_{t-j}^{(i)}$，可见一阶差分下预测值含有 $x_{t-1}^{(i)}=q_1^{(i)}$ 常数项（其余项含有权值变量）；类似地，差分阶数 $q=2$ 时，有常数项 $2x_{t-1}^{(i)}-x_{t-2}^{(i)}=q_2^{(i)}$；类似地，差分阶数 $q=3$ 时，有常数项 $3x_{t-1}^{(i)}-3x_{t-2}^{(i)}+x_{t-3}^{(i)}=q_3^{(i)}$. 考虑多阶差分信息可以在平稳、非平稳序列的预测中提高适用性、鲁棒性，把 $q_1^{(i)}$、$q_2^{(i)}$、$q_3^{(i)}$ 等差分特征显示地融入到线性自回归模型中，形式化如下：

$$\boldsymbol{x}_t^{(se,i)}=softmax\left(\left[a_1^{(i)},\cdots,a_Q^{(i)}\right]\right)\left[q_1^{(i)},\cdots,q_Q^{(i)}\right]^T+AR(x_{t-T:t-1}) \tag{2}$$

其中 $\left[a_1^{(i)},a_2^{(i)},\cdots,a_Q^{(i)}\right]$ 为差分特征的融合权值，限制为非负实数，$softmax(\cdot)$ 为对应的归一化操作，Q 为最大差分阶数；$AR(x_{t-T:t-1})=\sum_{j=1}^{T}w_{t-j}^{(i)}x_{t-j}^{(i)}$ 为标准的线性自回归项，T 为历史输入序列长度.

上述基于“多阶差分特征”的线性自回归模型为整体多变量任务建模、训练提供基准的同时评估各个变量的情况. 如图 2 中上半部分，第 i 个时间序列 $x^{(i)}$ 采用 Self Evolution (i)进行建模，由于不同的单变量时间序列的周期性各不相同，所以选取的历史窗口长度 T 也可能不相同，因此在图上方所画的单变量序列长度是不一样的.

3.4 基于CNN-LSTM的多变量依赖关系建模

捕捉复杂动态的时序模式与变量间依赖关系是多变量序列预测的关键. 相关变量序列的相关性往往体现在变量间存在着局部依赖关系，即单变量序列在某一时间点的取值依赖于相邻时间段其他相关序列的取值. 本文采用多个卷积核的一维CNN层的时序卷积对多元时间序列之间的复杂依赖关系进行表示，得到蕴含多变量序的局部依赖关系多层次信息的特征图（feature map）. 对于输入序列 $X_{t-T:t-1}=\left[x_{t-T:t-1}^{(1)},x_{t-T:t-1}^{(2)},\cdots,x_{t-T:t-1}^{(N)}\right]$ 进行变量维度上的一维卷积操作，表示如下：

$$C^{(k)}=W^{(k)}*X_{t-T:t-1} \tag{3}$$

其中 $W^{(k)}$ 为第 k 个卷积核，$C^{(k)}$ 是卷积结果，即从变量间依赖关系中提取的特征图.

在卷积网络中，通常采用 K 个卷积核 $W^{(1)},W^{(2)},\cdots,W^{(K)}$ 形成多个通道的特征图 $C^{(1)},C^{(2)},\cdots,C^{(K)}$. 不同于现有算法将所有卷积核设定为相同尺寸，本文采用可变尺寸的卷积核. 如图 2 中一维卷积部分，采用

$W^{(\cdot)} \in R^{1\times N}, W^{(2,c)} \in R^{3\times N}, W^{(3,c)} \in R^{5\times N} (c=1,2,\cdots,C)$ 卷积核组，以此建模多变量序列间的局部依赖关系，其中 C 为每个卷积核的通道数. 与同尺度卷积核相比，多尺度卷积核显式地描述了不同尺度的关联时间段特征，如公式（3）计算出的多尺度特征图记为 $C^{(1)}, C^{(2)}, \cdots, C^{(K)} \in R^{T\times C}$，蕴含着多变量序的局部依赖关系的多层次信息. 上述卷积操作是对变量维度进行的，所以多尺度特征图依然保留着原有的时序关系.

完成多变量序列一维卷积的特征提取后，SEPNets 采用长短记忆网络 LSTM 对多尺度特征图进行时序建模，来捕捉多变量依赖关系的时序信息. 如图 3 所示，第 k 个卷积核所有通道的特征图按时间维度进行拼接，然后作为对应的第 k 个长短记忆网络的输入；对应网络单元的隐藏输出输入到全连接层，形成输出向量 $h^{(k)} \in R^{N\times 1}$. 同样地，$K$ 个相同的网络结构总计输出 $h^{(1)}, h^{(2)}, \cdots, h^{(K)}$，据此将所有向量相加形成多变量依赖关系建模输出结果 $x_t^{(id)}$.

其中 $\hat{\boldsymbol{x}}_t^{(se)} = \left[x_t^{(se,1)}, x_t^{(se,2)}, \cdots, x_t^{(se,N)}\right]$ 表示自演化预训练模型的输出向量. $\hat{\boldsymbol{x}}_t^{(id)} = [x_t^{(id,1)}, x_t^{(id,2)}, \cdots, x_t^{(id,N)}]$ 表示多变量依赖关系模型的输出向量. 融合方法如下：

$$\hat{\boldsymbol{x}}_t = W^{(se)} \odot \hat{\boldsymbol{x}}_t^{(se)} + W^{(id)} \odot \hat{\boldsymbol{x}}_t^{(id)} \tag{4}$$

其中 $W^{(se)}, W^{(id)}$ 为融合权值，$\odot$ 表示元素乘法. 这样的两部分加权融合具有明确实际含义，增加了算法的可解释性. 例如，第 i 个变量的预测值 $\hat{x}_t^{(i)} = W^{(se,i)} \times \hat{x}_t^{(se,i)} + W^{(id,i)} \times \hat{x}_t^{(id,i)}$，那么对于该变量的预测，两项取值的相对大小便直接反映了两部分信息在预测中的占比.

完成模型构建后，为了训练模型，我们构造优化目标对应的损失函数如下：N 维变量预测训练模型的损失函数记为 $\mathcal{L}^{(se)} = \left[\mathcal{L}^{(se,1)}, \mathcal{L}^{(se,2)}, \cdots, \mathcal{L}^{(se,N)}\right]$，其中 $\mathcal{L}^{(se,i)}$ 为真实值与预测值的平方误差：

$$\mathcal{L}^{(se,i)} = \left(\hat{x}_t^{(se,i)} - x_t^{(se,i)}\right)^2 \tag{5}$$

该损失函数用于训练每个单变量预测模型 $\mathrm{SE}^{(i)}(\cdot)$，而整体的多变量预测损失函数为

$$L = \frac{1}{N}\sum_{i=1}^{N}(\hat{x}_t^{(i)} - x_t^{(i)})^2 \tag{6}$$

整体模型可基于梯度下降法进行优化，其具体流程如算法 1 所示.

算法 1.　SEPNets 训练过程.

输入：训练数据集 $\mathcal{D}: \left\{\left(X_{t-T:t-1}, X_t\right) \in \mathcal{D}\right\}$

1. FOR each　$i = 1,2,...N$　DO
2. 评估 $X^{(i)}$ 序列的周期性，选取历史窗口 $T^{(i)}$
3. 依据 $T^{(i)}$ 构造单序列训练集 $\mathcal{D}_{train}^{i}$
4. 依据 $\mathcal{L}^{(se,i)}$ 在 $\mathcal{D}_{train}^{i}$ 上训练得到序列 $X^{(i)}$ 的预训练模型 $SE^{(i)}(\cdot)$
5. END FOR
6. 合并单序列预训练模型，得整体预训练模型 $SE(\cdot)$
7. 选取 $T = max(T^{(1)}, T^{(2)}, \cdots, T^{(N)})$ 作为整体的历史窗口
8. 依据 T 构造多变量序列训练集 $\mathcal{D}_{train}$
9. 依据 $\mathcal{L}$ 在 $\mathcal{D}_{train}$ 上训练 $SE(\cdot)$ 和 CNN-LSTM 构建的网络模型 $Nets$，得到训练完毕的 SEPNets 模型参数.

输出：SEPNets 模型参数

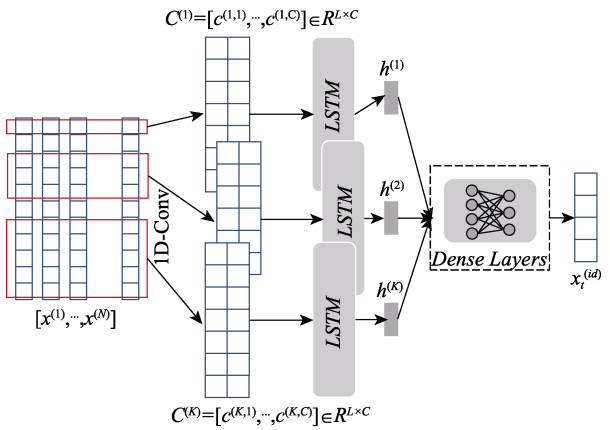

图 3　多变量时序依赖关系建模图

4　实验验证

4.1　实验数据集

本文基于以下五个真实数据集评估多变量预测算法的性能.

• 用电量数据集 Electricity①

该开放数据集记录了某一区域 2012 年至 2014 年约 300 个用户的用电量数据，时间间隔为 15 分钟，每个用户有约 14 万个采样点，取值单位为 kWh. 由于同一区域内，多个用户间具有相邻性，所以对应产生的用电量序列间具有相关性；不少工作采用此数据集进行多变量序列预测模型的训练以及评估[18] [13]. 首先，我们对数据集的每个用电量序列进行归一化处理，以减小不同用户数据不同数量级的影响，方式如下：

$$x = \frac{x - u_x}{1 + \sigma_x} \tag{7}$$

其中，u_x 为 x 的均值，σ_x 为 x 的标准差. 然后，

① https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

从300多个用户中随机选出32个用户形成本文分析的数据集；最后，在时间维度上对数据集进行划分：将 2012-2013 年历史数据作为训练集，将 2014 年全年数据作为测试集.

• 数据库系统数据集

包括从一家 IT 公司的 Oracle DBMS（数据库管理系统）收集的三个数据库的真实数据集.

这三个数据库支持着两种不同的服务，并分别部署在一个国家的两个省份. DBMS 监视 70 个预定义的关键性能指标（KPI），这些 KPI 涵盖了广泛的性能指标，包括硬件运行状况（例如，CPU 利用率），软件系统性能（例如，SQL 服务响应时间）以及其他一些与用户相关的指标（例如，活动会话数）. 监测系统每 3 分钟记录一次所有 KPI 测量值，并在相同的时间间隔内生成对应的数据库性能分数. 我们选择的三个数据库的名称分别为 JSDB、GSDB 和 GSDB2（处于隐私保护，我们隐去了公司的名称和数据库所采集的地点等信息）. 我们对三个数据库的 DBMS 采集数据，每隔 30 分钟计算高维系统 KPI 和性能分数的相关性量化值（线性相关系数），得到70个KPI时间序列与系统性能分数序列的相关系数时间序列. 所有 KPI 共同描述了同一系统，所以不同 KPI 本质上具有相关性，对应组成 KPI 相关性度量序列是相关多变量序列，且所有序列的取值范围都是 $[-1,1]$. 同样地，按照时间维度对数据集进行划分，前面 80%部分构成训练集，而后面的 20%部分构成测试集. 三个数据集的统计信息如表 1 所示.

表 1 DBMS 数据集统计信息

数据集名称	JSDB	GSDB	GSDB2
时间段	Sep.-Nov. 2018	Dec.2018-Jan.2019	Sep.-Nov. 2018
服务类型	Data Analysis	Web	Data Analysis
数据库实例数	15	2	2
数据样本数	28638	12956	14452
性能分数值域	[64,96]	[0,98]	[0,100]
性能分数均值	92.8	87.3	96.6
性能分数标准差	2.42	6.76	13.1
KPI 数	70	70	70

• 太阳能数据集 Solar-Energy①

数据集为 2006 年 Alabama 州选择的 70 个光伏电站采集的太阳能数据，时间间隔为 10 分钟. 同样地，将数据集按照时间维度进行划分，前面 80% 的数据集作为训练集，而之后的 20%的数据集作为测试集，然后对数据集的每个光伏电站的数据进行 min-max 归一化处理，使得数据的值映射到[0,1]，归一化方式如下：

$$\boldsymbol{x} = \frac{\boldsymbol{x} - min_{\boldsymbol{x}}}{max_{\boldsymbol{x}} - min_{\boldsymbol{x}}} \tag{8}$$

其中，$min_{\boldsymbol{x}}$ 为 $\boldsymbol{x}$ 的最小值，$max_{\boldsymbol{x}}$ 为 $\boldsymbol{x}$ 中的最大值.

4.2 基准算法与评估指标

本文将提出的 SEPNets 算法与下列算法进行对比：

• Autoregression(AR)[41]：一个基本的单变量时间序列建模、预测模型，是具有代表性的随机过程方法，其将当前时刻的输出变量当做历史时间段变量的线性加权和进行预测；

• Vector Autoregression (VAR)[14]：一个典型的统计类多变量时间序列分析算法，它将多变量时间序列看作多维随机过程，在单变量 AR 模型的基础上对变量依赖关系进行线性建模；

• LSTM[19]：一个基本的深度序列预测模型，它用 LSTM 基本单元建模多变量时间序列并基于多时间布历史数据进行预测；

• CNN-GRU-Input(All)-Att[17]：一个结合卷积神经网络 CNN(1d)和 GRU 来对多维时间序列进行预测的深度序列模型，并利用软注意力机制（Soft Attention）来提高预测性能，文章后面将其简称为 CGIA；

• LSTNets[18]：一个融合卷积神经网络（CNN）和递归神经网络（RNN）的深度序列模型，能够提取变量之间的长短期局部依赖关系，建模多变量预测任务中复杂的时序依赖模式.

此外，我们将第 3.3 节介绍的 $SE(\cdot)$ 的实现算法 EAR 也加入到对比算法中，并以单变量模型的预测精度作为各变量精度评估的基准.

评估多变量预测精度时，对于单变量序列预测值的平均绝对误差，第 i 个变量的平均绝对误差记为 $MAE^{(i)}$；所有变量的平均绝对误差的整体平均值记为 MAE，即

$$MAE = \frac{1}{N}\sum_{i=1}^{N} MAE^{(i)} \tag{9}$$

为了计算同一时刻下多变量预测值向量与真实值向量的相似度，我们采用向量间的余弦距离，记为 $Corr$. 两个向量 $\boldsymbol{u}$ 和 $\boldsymbol{v}$ 的余弦距离计算公式如下：

① https://www.nrel.gov/grid/solar-power-data.html

$$Corr = 1 - \frac{(\boldsymbol{u}-\bar{\boldsymbol{u}})\cdot(\boldsymbol{v}-\bar{\boldsymbol{v}})}{\left\|(\boldsymbol{u}-\bar{\boldsymbol{u}})\right\|_2 \left\|(\boldsymbol{v}-\bar{\boldsymbol{v}})\right\|_2} \tag{10}$$

其中 $\bar{\boldsymbol{u}}$ 是 $\boldsymbol{u}$ 元素的均值，$\boldsymbol{x}\cdot\boldsymbol{y}$ 是向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 的点积. $Corr \in [0,2]$，取值越小，预测值向量与真实值向量的相似度距离越小，相似度越大. 为了衡量模型预测能力的好坏，我们采用了方程的确定性系数 $R2$，公式如下所示：

$$R2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{y}_i - y_i)^2} \tag{11}$$

其中 $\hat{y}_i$ 为预测的值，y_i 为真实值，$\bar{y}_i$ 为 y_i 的均值，$R2$ 值越大，表示预测能力越好.

为了判断各变量的预测精度与对应单变量模型的预测精度相比是否显著变差，首先，我们计算得到各变量预测值的平方误差，第 i 个变量的平方误差记为集合 $\mathcal{SE}^{(i)} = \left\{SE^{(i,1)}, SE^{(i,2)}, \ldots, SE^{(i,N)}\right\}$；然后，先验地假设 $\mathcal{SE}^{(i)}$ 是对应分布的采样结果，该分布为 $x \sim \mathcal{X}^2(1)$（因为平方误差是误差的平方，并且一般假设误差服从正态分布）；最后，基于假设检验，构造统计量，对“各变量的预测精度与对应单变量模型的预测精度相比变差”进行显著性检验. 所有变量中预测误差比单变量模型误差显著大的计数为 B（“差”），剩余变量计数为 A（“优”），且 $B + A = N$ 表示所有变量总数. 为了评估多变量预测算法与单变量模型相比预测精度的提升情况，我们采用所有变量上的误差降低率，记为 $Ratio$，其计算公式如下：

$$Ratio = \frac{1}{N}\Sigma_{i=1}^{N} \frac{\left(MAE^{(se,i)} - MAE^{(i)}\right)}{1 + MAE^{(se,i)}} \times 100\% \tag{12}$$

其中 $MAE^{(se,i)}$ 为第 i 个变量单序列模型的预测误差；$Ratio$ 为任意取值的实数，取值越大算法对应的预测精度提升度越大.

4.3　实验环境及参数配置

实验物理机配置为：CPU 核心 i5，内存 32G 和 GPU GTX1070Ti，基于 TensorFlow 并使用 Keras 深度学习框架进行搭建.

根据 3.3 节与 3.4 节对模型的理论描述，我们通过 Keras 搭建了 SEPNets 的模型①，表 2 展示了其架构和相应的参数.

此外，我们在训练阶段利用了 EarlyStopping 机制，即在模型训练过程中，若模型在数据集中的一部分上表现开始下降时，则停止训练，从而避免继续训练模型导致过拟合的现象. 训练过程中利用 Adam 优化器，并选择了平均绝对误差（MAE）作为损失函数进行模型的训练.

表 2　SEPNets 模型架构及相应参数信息

Operation	Parameters	Inputs	Output
	Self Evolution		
Dense	units:1	$\boldsymbol{x}$	
Dense	units:1	$\boldsymbol{x}$	*se_pred*
	CNN-LSTM		
Conv1D	filters:1; kernel_size:1	$\boldsymbol{x}$	*c*1
Conv1D	filters:1; kernel_size:3	$\boldsymbol{x}$	*c*3
Conv1D	filters:1; kernel_size:5	$\boldsymbol{x}$	*c*5
LSTM	units:16	*c*1	*r*1
LSTM	units:16	*c*3	*r*3
LSTM	units:16	*c*5	*r*5
Add		*r*1,*r*3,*r*5	*a*
Dense	units:num_var	*a*	*res_pred*
Multiply	units:num_var	*se_pred*	*se*
Multiply	units:num_var	*res_pred*	*res*
Add		*se*,*res*	*y_pred*

4.4　实验结果及分析

图 4 显示了 SEPNets 模型在五个数据集上随着训练 epoch 的平均损失变化情况. 为了将所有数据集训练的损失规定在 0 到 0.1 之间进行显示，对其中的 Electricity 数据集的损失除以 10. 结果显示在 SEPNets 的训练阶段，模型的训练平均损失一直下降直到收敛.

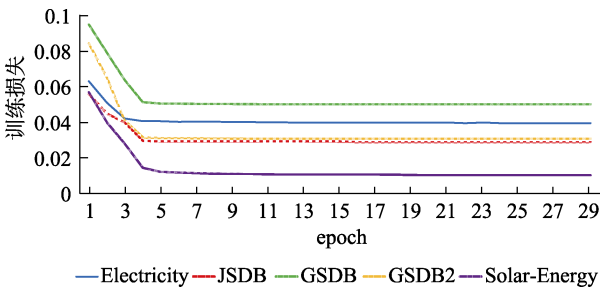

图 4　训练阶段 SEPNets 模型平均损失随 epoch 变化图

表 3 对比了五个数据集上不同算法的多变量预测平均绝对误差 MAE、相似度距离 $Corr$ 和 $R2$，MAE 和 $Corr$ 都是取值越小，表明对应算法多变量预

① https://github.com/weanl/SEPNets

测性能越好，而 $R2$ 取值越大表明多变量预测算法越有效. 如表所示，在五个时间序列数据集上，多变量算法的预测性能总体上好于单变量算法 AR 和 EAR. 在多变量算法中，VAR 和 LSTM 算法在不同数据集上相对的预测性能表现不同，但基本上差于 CNN-GRU-Input(All)-Att(CGIA) 和 LSTNets 算法. 与其他算法相比，本文提出的算法 SEPNets 无论在 MAE，$Corr$ 还是 $R2$ 上基本都有较好的提升，在多变量预测任务中获得最好的整体预测精度.

若将 SEPNets 第一部分的基于差分特征构造的线性自回归模型 EAR 替换成简单的自回归模型 AR，经过实验验证，采用 EAR 的 SEPNets 模型在多变量时序预测性能上将比采用 AR 的 SEPNets 模型平均提高 2%左右，从结果可以证明多阶差分信息可以在 AR 的基础上提高模型预测的适用性和鲁棒性.

表 3 多变量时序预测误差对比表

算法	Electricity			JSDB			GSDB			GSDB2			Solar-Energy		
	MAE	*Corr*	*R2*	*MAE*	*Corr*	*R2*	*MAE*	*Corr*	*R2*	*MAE*	*Corr*	*R2*	*MAE*	*Corr*	*R2*
AR	0.514	0.524	0.232	0.0619	0.126	0.703	0.0537	0.180	0.701	0.0560	0.144	0.712	0.0267	0.0718	0.985
EAR	0.485	0.449	0.340	0.0422	0.121	0.764	0.0501	0.640	0.701	0.0459	0.135	0.720	0.0215	0.0687	0.986
VAR	0.455	0.428	0.361	0.0378	0.119	0.679	0.0502	0.153	0.678	0.0386	0.132	0.718	0.0217	0.0655	0.980
LSTM	0.442	0.431	0.372	0.0374	0.118	0.622	0.0510	**0.151**	0.639	0.0394	0.126	0.718	0.0234	0.0648	0.982
CGIA	0.434	0.432	0.343	0.0372	0.113	0.762	0.0501	0.152	0.691	0.0391	0.124	0.717	0.0221	0.0636	0.986
LSTNets	0.435	0.435	0.343	0.0372	0.109	**0.765**	0.0497	**0.151**	0.696	0.0385	0.120	0.720	0.0220	0.0612	0.987
SEPNets	**0.418**	**0.405**	**0.379**	**0.0371**	**0.107**	0.764	**0.0493**	**0.151**	**0.704**	**0.0378**	**0.110**	**0.722**	**0.0199**	**0.0563**	**0.989**

如图 5 所示，我们选择了 VAR,LSTM 和 LSTNets 这三个典型的多变量基准算法和提出的 SEPNets 算法，以此对比不同算法对于所有变量预测误差比单变量模型误差显著大的计数 B（“差”）以及剩余变量计数为 A（“优”）的结果. 图 6 中的曲线显示了四个数据集上四种多变量预测算法整体的误差降低率.

图 5(a)展示了四种多变量预测算法在 Electricity 数据集上的性能. 从误差降低率来看，VAR 为 2.00%，LSTM 为 2.82%，LSTNets 为 3.25%，SEPNets 为 4.40%，这说明所有多变量算法都一定程度上降低了预测误差，并且 SEPNets 取得了最好的降低率. 值得注意的是，Electricity 数据集的总变量数为 32. SEPNets 算法对应的“差”变量数为 0、“优”变量数为 32，这表明所有的 32 个变量的预测精度都没有比单变量模型的预测精度显著地变差；而 VAR 分别为 13 和 19，LSTM 分别为 11 和 21，LSTNets 分别为 4 和 28，这表明现有多变量算法虽然都能一定程度上降低预测误差，但出现了不少局部单变量的预测精度比对应的单序列模型的预测精度差的情况.

图 5（b）、图 5（c）、图 5（d）、图 5（e）分别展示了其余的四个数据集结果，他们的总变量数都为 70. 多变量预测算法中，VAR 和 LSTM 算法在不同数据集上相对的预测误差降低率表现不同，但都低于 LSTNets 算法. 本文所提的 SEPNets 算法获得最高的误差降低率，同时对应的“差”变量数和“优”变量数分别为 0 和 70、1 和 69、1 和 69、0 和 70. 综上所述，SEPNets 可以极大减少单变量的预测精度比对应的单序列模型的预测精度差的情况，与现有多变量时间序列预测算法相比，在保障局部变量预测精度上有显著的性能提升.

经过实验分析，通过保障局部变量预测精度，多变量整体预测性能也变得更好，这说明局部变量的预测与整体变量的预测是相关的，公式（4）对自演化预训练模型和多变量依赖关系模型两部分结果按权重进行融合，通过让损失函数 $\mathcal{L}$ 达到最小，这样既可以控制局部单变量的精度损失不会变大，同时也可以控制整体预测精度损失，从而达到我们所预期的目标.

图 6（a）和图 6（b）分别展示了 EAR, LSTM, LSTNets 和 SEPNets 这四种算法在五个数据集上的训练时间与测试时间. 结果表示 SEPNets 的训练时间和测试时间最长. 比如对于数据集 Electricity，在训练过程中 EAR，LSTM 和 LSTNets 的训练时间分别为 0.4h，1.2h，1.1h，而 SEPNets 的训练时间为 4.1h; 而在测试过程中 EAR，LSTM 和 LSTNets 总的测试时间分别为 1.25ms，1.20ms，1.14ms，SEPNets 的测试时间为 2.6ms. 虽然 SEPNets 与其他多变量时间

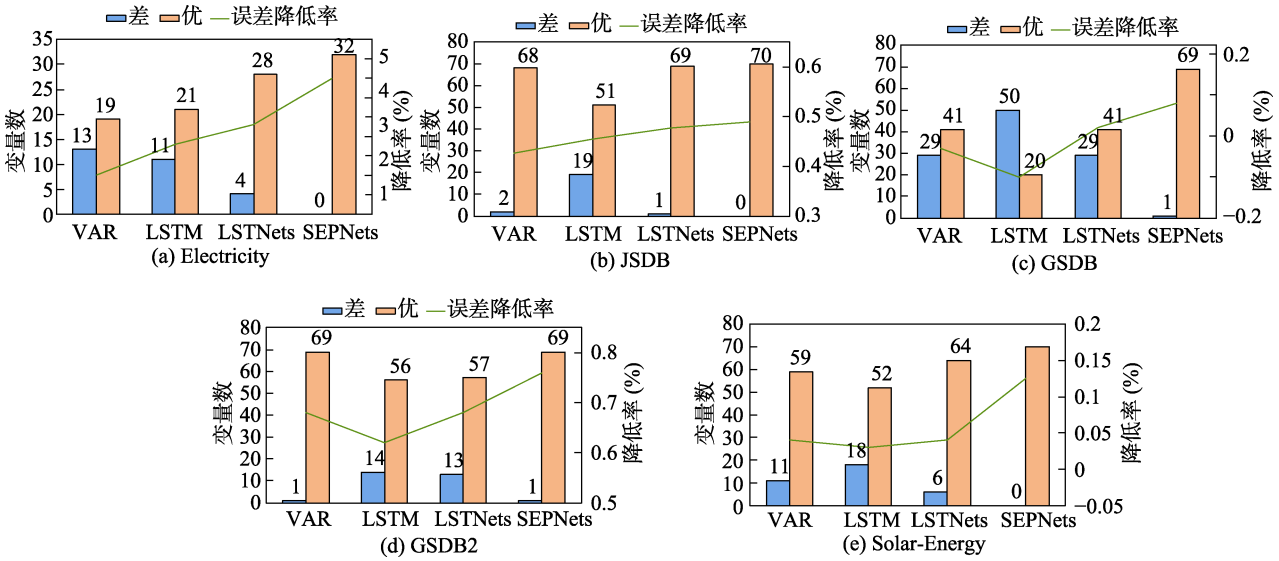

图 5　各变量精度判别结果与误差降低率图

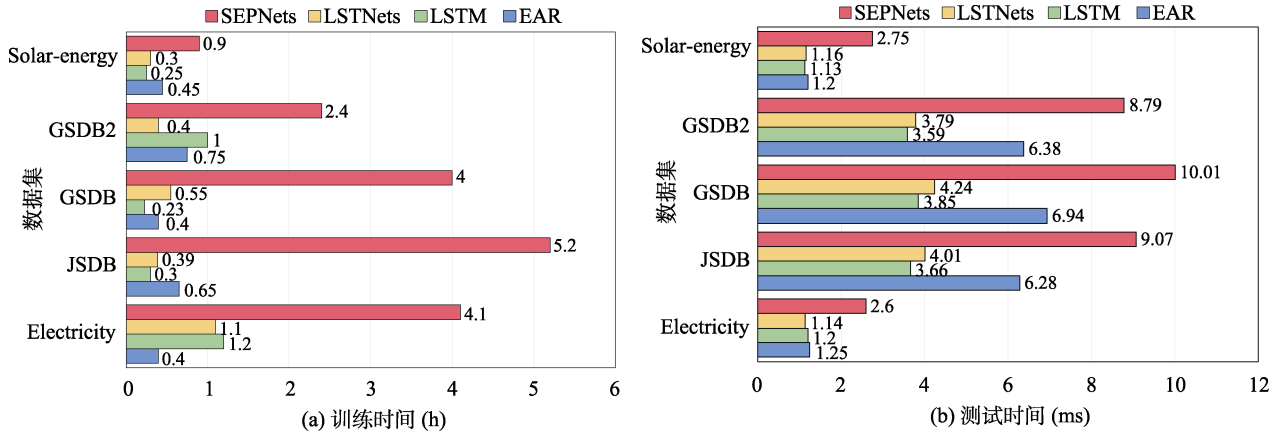

图 6　算法在不同数据集上训练和测试时间对比

序列预测算法在对多变量依赖关系建模上是相似的，但不同之处在于还需要先对单变量进行预训练，所以使得训练和测试时间相比较而言要更长.

5　总结与展望

多变量时序预测是典型的时间序列分析任务，对于辅助决策、资源配置、提前采取止损措施等方面有重要意义，在包括电力、气象、交通、商业等领域有广泛应用. 然而，现有多变量时间序列预测算法关注整体预测目标的优化，会带来局部变量预测精度损失的问题. 针对局部变量预测精度保障问题，本文提出了一种基于自演化预训练的多变量时间序列预测算法 SEPNets. 首先，基于预训练策略，设计和训练单序列预测基准模型，作为后续整体任务的基准知识；然后，设计多层次 CNN-RNN 级联神经网络模型，对多变量之间的相互时-空依赖关系进行建模；最后，将已训练的单变量模型作为预训练模块，与多变量建模模块的输出一一对应融合，从而利用已训练的单序列模型来优化整体模型的性能，解决局部单变量的预测精度比对应的单序列模型的预测精度差的问题. 本文基于五个真实数据集进行对比试验，结果表明，基于自演化预训练的多变量时序预测算法，具有较好的整体预测精度，对于局部单变量的预测精度保障性方面远优于现有的多变量时间序列预测算法.

尽管本文的算法取得了较好的实验结果，它仍然具有以下局限性. 首先，算法需要先对单变量进行自演化预训练，再对多变量依赖关系进行建模，最后对自演化预训练模型和多变量依赖关系模型两部分结果进行融合，因此在训练时间和测试时间上，本算法相较于其他算法消耗的时间更长. 其次，本算法只能支持单步时间序列预测，不能适用于长期的多步时间序列预测问题.

在未来的工作中，我们会在本算法基础上进一

步研究如何提升训练的和测试的效率，降低时间复杂度. 其次，我们将研究在不牺牲局部变量多步预测精度的同时，设计出一个更优的多变量时间序列多步预测算法. 此外，我们拟融合多任务学习等深度学习技术，进一步优化多变量时间序列的预测精度和模型的泛化能力.

参 考 文 献

[1] Raicharoen T, Lursinsap C, Sanguanbhokai P. Application of critical support vector machine to time series prediction// Proceedings of the 2003 International Symposium on Circuits and Systems. Bangkok, Thailand, 2003: 741-744

[2] Cortez P, RIO M, Rocha M, et al. Multi-scale internet traffic forecasting using neural networks and time series methods. Expert Systems, 2012, 29(2): 143-155

[3] Botezatu M M, Giurgiu I, Bogojeska J, et al. Predicting disk replacement towards reliable data centers//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, USA, 2016: 39-48

[4] Box G E, Jenkins G M, Reinsel G C, et al. Time Series Analysis: Forecasting and Control.NJ, USA: John Wiley & Sons, 2015

[5] Zhang G P. Time series forecasting using a hybrid arima and neural network model. Neurocomputing, 2003, 50: 159-175

[6] Ahmed N K, Atiya A F, Gayar N E, et al. An empirical comparison of machine learning models for time series forecasting. Econometric Reviews, 2010, 29(5-6): 594-621

[7] De Gooijer J G, Hyndman R J. 25 years of time series forecasting. International Journal of Forecasting, 2006, 22(3): 443-473

[8] Jin-Fang Y, Yong-Jie Z, Dong-Feng W, et al. Time series prediction based on support vector regression. Proceedings of the CSEE, 2005, 25(017): 110-114(in Chinese)
(杨金芳，翟永杰，王东风，徐大平. 基于支持向量回归的时间序列预测. 中国电机工程学报, 2005, 25(017): 110-114)

[9] Bontempi G, Taieb S B, Le Borgne Y A. Machine learning strategies for time series forecasting//European Business Intelligence Summer School. Brussels, Belgium: Springer, 2012: 62-77

[10] Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge, UK : MIT Press, 2016

[11] Zhou Z H, Feng J. Deep forest: Towards An Alternative to Deep Neural Networks//Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. Melbourne, Australia, 2017: 3553-3559

[12] Ong Y J, Qiao M, Jadav D. Temporal tensor transformation network for multivariate time series prediction//International Conference on Big Data. Atlanta, USA, 2020: 1594-1603

[13] Yin J, Rao W, Yuan M, et al. Experimental study of multivariate time series forecasting models//Proceedings of the 28th ACM International Conference on Information and Knowledge Management. Beijing, China, 2019: 2833-2839

[14] Lütkepohl H. New introduction to multiple time series analysis. Berlin,Germany: Springer Science & Business Media, 2005

[15] Guo T, Xu Z, Yao X, et al. Robust online time series prediction with recurrent neural networks//2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA). Montreal, Canada, 2016: 816825

[16] Lipton Z C, Kale D C, Elkan C, et al. Learning to diagnose with lstm recurrent neural networks. arXiv preprint arXiv:1511.03677, 2015

[17] Thongniran N, Jitkajornwanich K, Lawawirojwong S, et al. Combining attentional cnn and gru networks for ocean current prediction based on hf radar observations//Proceedings of the 2019 8th International Conference on Computing and Pattern Recognition. New York, USA, 2019: 440–446

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

[19] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735-1780

[20] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, USA, 2019: 4171-4186

[21] Wang J, Hertzmann A, Fleet D J. Gaussian process dynamical models//Advances in Neural Information Processing Systems. Vancouver, Canada, 2006: 1441-1448

[22] Kolarik T, Rudorfer G. Time series forecasting using neural networks. ACM SIGAPL APL Quote Quad, 1994, 25(1): 86-94

[23] Marino D L, Amarasinghe K, Manic M. Building energy load forecasting using deep neural networks//IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society. Florence, Italy. 2016: 7046-7051

[24] Narayanan A, Verma S, Ramadan E, et al. Deepcache: A deep learning based framework for content caching//Proceedings of the 2018 Workshop on Network Meets AI & ML. Budapest, Hungary ,2018: 48-53

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

[26] Gensler A, Henze J, Sick B, et al. Deep learning for solar power forecasting—an approach using autoencoder and lstm neural networks //2016 IEEE international conference on systems, man, and cybernetics (SMC). Budapest, Hungary, 2016: 002858-002865

[27] Cho K, Van Merriënboer B, Bahdanau D, et al. On the properties of neural machine translation: Encoder-decoder approaches// Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. Doha, Qatar, 2014: 103-111

[28] Tan Y H, Chan C S. phi-lstm: a phrase-based hierarchical lstm model for image captioning//Asian Conference on Computer Vision. Taipei, China: Springer, 2016: 101-117

[29] Campos V, Jou B, Giró-I Nieto X, et al. Skip rnn: Learning to skip state updates in recurrent neural networks. arXiv preprint arXiv:1708.06834, 2017

[30] Wang Y, Liu Z, Hu D, et al. Multivariate time series prediction based on optimized temporal convolutional networks with stacked autoencoders//Asian Conference on Machine Learning. Nagoya, Japan, 2019: 157-172

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

[32] 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

[33] Bai S, Kolter J Z, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling// 2018 International Joint Conference on Neural Networks. Rio de Janeiro, Brazil, 2018: 1-7

[34] Hu C, Hu Y, Seo S. A deep structural model for analyzing correlated multivariate time series//2019 18th IEEE International Conference On Machine Learning And Applications. Orlando, USA. 2019: 69-74

[35] Assaf R, Schuman n A. Explainable deep neural networks for multivariate time series predictions//Proceedings of the 28th International Joint Conference on Artificial Intelligence. Macao, China, 2019: 6488-6490

[36] Goel H, Melnyk I, Banerjee A. R2n2: residual recurrent neural networks for multivariate time series forecasting. arXiv preprint arXiv:1709.03159, 2017

[37] Peng-Cheng Z, Li-Yan W, Shun-Hui J, et al. Web service qos forecasting approach using multivariate time series. Journal of Software, 2019(6): 1742-1758(in Chinese)
(张鹏程, 王丽艳, 吉顺慧, 等. 多元时间序列的 Web Service QoS 预测方法. 软件学报, 2019(6): 1742-1758)

[38] Hong-Bing O, Kang H, Hong-Ju Y. Prediction of financial time series based on lstm neural network. Chinese Journal of Management Science, 2020, 28(4): 30-38(in Chinese)
(欧阳红兵, 黄亢, 闫洪举. 基于 LSTM 神经网络的金融时间序列预测. 中国管理科学, 2020, 28(4): 30-38)

[39] Shuang-Cheng W, Gao-Rui, Rui-Jie D. With super parent node bayesian network ensemble regression model for time series. Chinese Journal of Computers, 2017, 40(12): 2748-2761 (in Chinese)
(王双成, 高瑞, 杜瑞杰. 具有超父结点时间序列贝叶斯网络集成回归模型. 计算机学报, 2017, 40(12): 2748-2761)

[40] Li W H, Jia-Yu C, Chen-Yang X. Prediction method of cyclic time series based on dtw similarity. Computer Science, 2019, 46(05): 157-162(in Chinese)
(李文海，程佳宇，谢晨阳. 基于 DTW 相似判定的周期性时间序列预测方法. 计算机科学, 2019, 46(05): 157-162)

[41] Hamilton J D. Time Series Analysis. Volume 2. New Jersey, USA: Princeton New Jersey, 1994

WAN Chen, master student. His research interests include time series analysis, and deep learning.

LI Wen-Zhong, Ph.D., professor. His research interests include distributed computing, social network analysis, and deep learning.

DING Wang-Xiang, master student. His research interests include time series analysis, and deep learning.

ZHANG Zhi-Jie, master student. His research interests include time series analysis, and deep learning.

YE Bao-Liu, Ph.D., professor. His research interests include distributed computing and parallel processing.

LU Sang-Lu, Ph.D., professor. her research interests include distributed computing and parallel processing.

Background

Time series forecasting is a typical time series analysis task, which is of great significance for assisting decision-making, resource allocation, and taking stop loss measures in advance. It is widely used in fields including power, weather, transportation, and business. In recent years, time series forecasting algorithms based on machine learning have been a hot research field, among which multivariate time series forecasting is a challenging task. This paper focuses on the local variable forecasting accuracy of multivariate time series forecasting, that is, multivariate forecasting needs to improve the overall forecasting performance while ensuring the forecasting accuracy of local univariate forecastings. In view of the above problems, this paper analyzes the limitations of the multivariate time series forecasting algorithm. The multivariate time series forecasting algorithm SEPNets based on self-evolution pre-training is designed and implemented. Inspired by the pre-trained model, SEPNets first constructs and trains a univariate time series model as a benchmark for subsequent modeling; then expands the complex temporal dependence between the time-series convolutional network (1-dimensional convolution) and the long and short memory (LSTM) unit modeling variables Relationship; using the former as the latter pre-training model for fusion and retraining, SEPNets solves the problem of local variable forecasting accuracy for multivariate time series forecasting. The experimental results show that the SEPNets algorithm proposed in this paper ensures the forecasting accuracy of local variables to a certain extent while obtaining the relatively highest forecasting accuracy.