

# 基于 ARIMA 和 LSTM 混合模型的时间序列预测

王英伟 马树才

(辽宁大学经济学院 辽宁 沈阳 110036)

**摘要** 由于现实中的时间序列通常同时具有线性和非线性特征,传统 ARIMA 模型在时间序列建模中常表现出一定局限性。对此,提出基于 ARIMA 和 LSTM 混合模型进行时间序列预测。应用线性 ARIMA 模型进行时间序列预测,用支持向量回归(SVR)模型对误差序列进行预测,采用深度 LSTM 模型对 ARIMA 模型和 SVR 模型的预测结果组合,并将贝叶斯优化算法用于选择深度 LSTM 模型的超参数。实验结果表明,与其他混合模型相比,该模型在五种不同时间序列预测中能够有效提高预测精度。

**关键词** ARIMA 模型 SVR 模型 深度 LSTM 模型 贝叶斯优化算法 时间序列预测

中图分类号 TP302.7

文献标志码 A

DOI:10.3969/j.issn.1000-386x.2021.02.047

## TIME SERIES FORECASTING BASED ON ARIMA\_DLSTM HYBRID MODEL

Wang Yingwei Ma Shucai

(Institute of Economics, Liaoning University, Shenyang 110036, Liaoning, China)

**Abstract** Because real-world time series usually contain both linear and nonlinear patterns, traditional ARIMA model has a limited performance in the time series modeling. In view of this, we propose ARIMA\_DLSTM hybrid model for time series forecasting. Linear ARIMA model was used for time series prediction firstly, and then support vector regression (SVR) was used for error series prediction. The deep LSTM model was introduced to combine the forecasts of ARIMA model and SVR model, and Bayesian optimization algorithm was adopted to obtain the optimal hyper-parameter of deep LSTM model. The experimental results of five time series forecasting show that ARIMA\_DLSTM model can effectively improve the prediction accuracy compared with other hybrid models.

**Keywords** ARIMA model SVR model Deep LSTM model Bayesian optimization algorithm Time series forecasting

## 0 引言

时间序列预测在众多领域有广泛应用,如金融、经济、工程和航空等,并成为机器学习领域的重要研究课题<sup>[1]</sup>。现实中时间序列通常同时具有线性和非线性特征,因此对不同时间序列现象建立模型并准确预测已成为最具挑战的应用之一。

ARIMA 等统计模型以简单和灵活性被大量用于时间序列预测<sup>[2-3]</sup>,但现实中的时间序列通常具有非线性特征,而传统时间序列预测方法是线性模型,在对时间序列建模中表现出一定局限性<sup>[4]</sup>。因此,支持向量机和神经网络等非线性模型,被广泛用于时间序列预测领域<sup>[5]</sup>。其中神经网络具有强大的非线性、映射

性和自适应性等特征,可以有效提高预测精度,以 LSTM 为代表的深度神经网络成为近年来的研究热点。但单一非线性模型对同时具有线性和非线性特征的时间序列并不能获得最优结果<sup>[6]</sup>。

由于单一模型的局限性,众多学者提出由线性和非线性模型组成的混合模型。Zhang<sup>[7]</sup>和 Oliveira 等<sup>[1]</sup>假设时间序列同时具有线性和非线性特征,并分别提出 ARIMA-ANN 混合模型和 ARIMA-SVR 混合模型,ARIMA 用于提取线性特征,而 ANN 和 SVR 用于提取残差特征,最后对两者进行组合,后者应用 PSO 选择模型参数。文献[8]提出 ARIMA-SVR\_s 混合模型,首先将时间序列分解为高波动率和低波动率两部分,并分别使用 ARIMA 和 AR\_SVR 对两部分建模,最后将两者结果组合。实验结果证明,混合模型可以有效

提高预测精度。以上模型均假设时间序列是线性特征和非线性特征的线性组合,但在实际应用中,两者可能存在非线性关系,从而影响混合模型的性能。

针对上述问题,本文对线性和非线性模型预测结果的组合方式提出新的策略。首先采用 ARIMA 模型提取时间序列线性特征,然后用 SVR 模型对 ARIMA 模型的预测值和实际值间的误差序列进行预测,最后将前两者预测结果作为 LSTM 模型的输入对时间序列进行预测,并应用贝叶斯优化算法选择深度 LSTM 模型的超参数。

### 1 ARIMA 模型

ARIMA 模型是由 Box 和 Jenkins 提出的用于时间序列分析和预测的线性模型。在时间序列分析中,需要设置 3 个参数,分别为自回归阶数( $p$ )、差分阶数( $d$ )和移动平均阶数( $q$ ),ARIMA( $p, d, q$ )的一般形式如下:

$$\begin{cases} \phi(B) \nabla^d y_t = \theta(B) \varepsilon_t \\ E(\varepsilon_t) = 0 \quad \text{var}(\varepsilon_t) = \sigma_\varepsilon^2, E(\varepsilon_t \varepsilon_s) = 0, s \neq t \\ E(x_t \varepsilon_s) = 0 \quad s < t \end{cases} \quad (1)$$

$$\phi(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \dots - \varphi_p B^p \quad (2)$$

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q \quad (3)$$

式中: $B$  为后移算子; $\nabla^d = (1 - B)^d$  为高阶差分; $\varphi_i (i = 1, 2, \dots, p)$  和  $\theta_j (j = 1, 2, \dots, q)$  分别为自回归参数和移动平均参数; $\varepsilon_t$  为符合  $N(0, \sigma^2)$  正态分布的误差项。式(2)和式(3)分别为自回归相关系数多项式和移动平均系数多项式。

时间序列  $y_t$  应先采用 ADF 检验选取差分阶数,将时间序列变为平稳序列,再根据赤池信息准则(Akaike Information Criterion, AIC)选择最佳模型。

### 2 支持向量回归

支持向量回归是基于结构风险最小化(SRM)原则进行泛化误差上界最小化,最初由文献[9]提出的机器学习方法。

在训练集  $\{x_i, y_i\}_{i=1}^l$  中,共有  $l$  个训练样本,  $x_i \in \mathbf{R}^d$  表示第  $i$  个输入向量,  $y_i \in \mathbf{R}$  表示第  $i$  个输出向量,则非线性回归函数可以表示为:

$$f(x) = \omega \varphi(x) + b \quad (4)$$

式中: $\omega$  为权值向量; $\varphi$  是将样本空间映射到高维空间的非线性变换函数; $b$  为偏差。

为保证支持向量回归具有稀疏性,引入  $\varepsilon$  不敏感

损失函数,允许样本值落入不敏感损失带,即允许最大为  $\varepsilon$  的误差,公式如下:

$$L(y, f(x)) = \begin{cases} 0 & |f(x) - y| < \varepsilon \\ |f(x) - y| - \varepsilon & \text{其他} \end{cases} \quad (5)$$

参数  $\omega$  和  $b$  通过式(6)求得。

$$\begin{aligned} \min \quad & \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^l (\xi_i - \xi_i^*) \\ \text{s. t.} \quad & y_i - \omega \varphi(x) - b \leq \varepsilon + \xi \\ & \omega \varphi(x) + b - y_i \leq \varepsilon + \xi^* \\ & \xi, \xi^* \geq 0, i = 0, 1, \dots, l \end{aligned} \quad (6)$$

式中: $\|\omega\|^2$  为模型复杂度项; $C$  为惩罚因子,用于平衡模型复杂度和最小训练误差; $\xi$  和  $\xi^*$  表示松弛变量,用以度量不敏感损失带外的训练样本偏离程度。将式(6)引入拉格朗日函数和核函数,则非线性回归函数表示为:

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) k(x_i, x_j) + b \quad (7)$$

式中: $\alpha_i$  和  $\alpha_i^*$  为拉格朗日算子; $k(x_i, x_j)$  为核函数,可以表示为  $\varphi(x_i)$  和  $\varphi(x_j)$  的内积,即  $k(x_i, x_j) = \varphi(x_i) \varphi(x_j)$ 。本文采用径向基核函数,即  $k(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / 2\gamma^2)$ ,  $\gamma$  为核参数。

### 3 LSTM 模型

循环神经网络(RNN)是用于处理序列数据的神经网络。其最大特点是神经元在某一时刻的输出能够作为输入再次输入到神经元,从而使网络具有记忆能力,但 RNN 无法解决长期时间序列依赖问题。因此, Hochreiter 等<sup>[10]</sup> 提出长短期记忆(LSTM)神经网络。

LSTM 模型是标准 RNN 的一种变体,通过引入记忆单元(Memory Cell)解决长期依赖问题,即梯度消失和梯度爆炸<sup>[11]</sup>,记忆单元结构如图 1 所示。

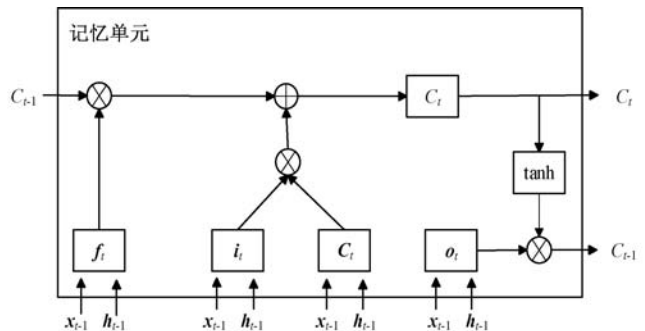


图 1 LSTM 记忆单元图

每个记忆单元包含三个门控结构:遗忘门( $f_t$ )、输入门( $i_t$ )和输出门( $o_t$ ),用于对类似传送带的单元状态(Cell State)进行移除和添加信息。每个门控结构由一个 sigmoid 神经网络层和一个点乘操作组成。假设

$x_t$  表示  $t$  时刻的输入向量,  $W_f$ 、 $W_s$ 、 $W_i$  和  $W_o$  表示循环层权重矩阵,  $U_f$ 、 $U_s$ 、 $U_i$  和  $U_o$  表示输入层权重矩阵,  $b_f$ 、 $b_s$ 、 $b_i$  和  $b_o$  表示偏置向量,  $h_t$  表示 LSTM 层的输出向量, 计算过程如下:

1) 通过遗忘门移除单元状态  $S_{t-1}$  中的无用历史信息, 并计算遗忘门激活值:

$$f_t = \text{sigmoid}(U_f x_t + W_f h_{t-1} + b_f) \quad (8)$$

2) 通过输入门决定将哪些新信息存储到单元状态:

$$i_t = \text{sigmoid}(U_i x_t + W_i h_{t-1} + b_i) \quad (9)$$

$$\tilde{S}_t = \tanh(U_s x_t + W_s h_{t-1} + b_s) \quad (10)$$

3) 将旧单元状态  $S_{t-1}$  更新为新单元状态  $S_t$ :

$$S_t = f_t \otimes S_{t-1} \oplus i_t \otimes \tilde{S}_t \quad (11)$$

4) 输出门通过 sigmoid 层决定需要输出的单元状态, 并将输出结果  $O_t$  和经过 tanh 层的新单元状态  $\tanh(S_t)$  相乘计算记忆单元的输出  $h_t$ :

$$O_t = \text{sigmoid}(U_o x_t + W_o h_{t-1} + b_o) \quad (12)$$

$$h_t = O_t \otimes \tanh(S_t) \quad (13)$$

LSTM 神经网络要求输入连续时间步上的特征向量值以便进行训练。假设每个序列共有  $N$  个时间步, 并且  $S_N$  表示第  $N$  个时间步的特征向量, 所以第  $M$  个序列可以表示为  $\{S_M, S_{M+1}, \dots, S_{M+N-1}\}$ , 输入特征向量序列的构造方式如图 2 所示。

特征向量

时间	1	2	3	...	$N$	$N+1$	$N+2$	...
向量	$S_1$	$S_2$	$S_3$	...	$S_n$	$S_{n+1}$	$S_{n+2}$	...

1	2	3	...	$N$	序列 1
$S_1$	$S_2$	$S_3$	...	$S_n$	

2	3	...	$N$	$N+1$	序列 2
$S_2$	$S_3$	...	$S_n$	$S_{n+1}$	

图 2 LSTM 神经网络的输入特征向量序列构造方式

## 4 深度 LSTM 神经网络

深度 LSTM 神经网络是由多个基本 LSTM 模块组成, 核心思想是在较低 LSTM 层建立输入数据的局部特征, 并在较高 LSTM 层进行整合。实验证明, 深度 LSTM 神经网络结构的学习能力更强<sup>[12]</sup>。

在如图 3 所示的深度 LSTM 网络结构中:  $h_{t-1}^1$  表示 LSTM 神经网络第 1 层在  $t-1$  时刻的输出向量; 在  $t$  时刻的输入向量  $X_t$  和在  $t-1$  时刻的输出向量  $h_{t-1}^1$ , 经过 LSTM 第 1 层后输出状态向量为  $h_t^1$ , 并作为 LSTM 第 2

层的输入和在  $t+1$  时刻的 LSTM 第 1 层输入, 迭代过程在 LSTM 第  $n$  层结束。

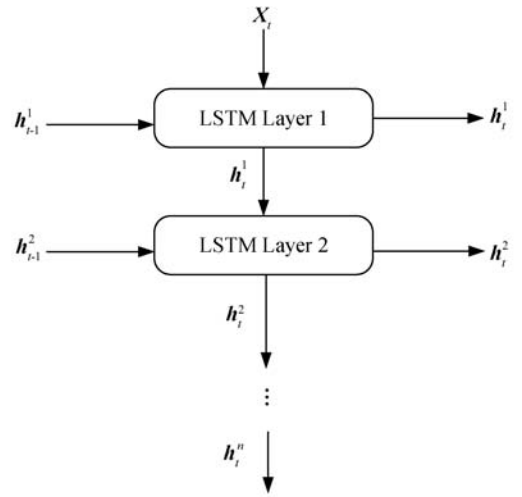


图 3 深度 LSTM 网络结构

## 5 贝叶斯优化算法

贝叶斯优化算法是一种高效的全局优化算法<sup>[13]</sup>。它能够在每次迭代中, 根据代理模型拟合实际目标函数的结果选择最优评估点, 减少目标函数的迭代次数, 尤其适合对评估代价高昂的黑盒目标函数进行优化, 因此被广泛用于机器学习和深度学习的超参数优化。

假设目标函数  $f: X \rightarrow R$ , 贝叶斯优化采用基于模型的序贯优化法对式 (14) 中的最优化问题求解。

$$x^* = \arg \min_{x \in X} f(x) \quad (14)$$

贝叶斯优化算法的迭代过程可以分为三个部分。(1) 根据最大化采集函数选择最优评估点, 即  $x_{n+1} = \text{argmax}_p(x)$ , 其中  $a_{p(f)}: X \rightarrow R$  为采集函数,  $p(f)$  为  $f$  的先验概率分布; (2) 评估目标函数  $y_{n+1} = f(x_{n+1}) + \varepsilon$ , 并将新的点  $(x_{n+1}, y_{n+1})$  加入观测数据  $D_n = (x_j, y_j)$ ,  $j = 1, 2, \dots, n$ ; (3) 更新  $f$  的后验概率分布和采集函数, 分别表示为  $p(f|D_{n+1})$  和  $a_{p(f|D_{n+1})}$ 。

本文采用高斯过程 (GP) 作为代理模型, 而采集函数则使用期望提高 (Expected Improvement, EI) 函数。

## 6 ARIMA\_DLSTM 模型

本文提出一种基于 ARIMA 模型、SVR 模型和深度 LSTM 模型的混合模型: ARIMA\_DLSTM 模型。ARIMA\_DLSTM 模型由三部分组成。第一部分是线性 ARIMA 模型, 对时间序列  $L_t$  进行单步预测, 预测结果表示为  $\hat{R}_t$ ; 第二部分是对 ARIMA 模型的误差  $\varepsilon_t$  进行预测, 预测结果表示为  $\hat{\varepsilon}_t$ , 其中  $\varepsilon_t = L_t - \hat{R}_t$ ; 第三部分是利用

DLSTM 模型将预测结果  $\hat{R}_t$  和  $\hat{\varepsilon}_t$  进行非线性组合, 预测结果表示为  $\hat{L}_t$ 。图 4 给出了 ARIMA\_DLSTM 模型流程。

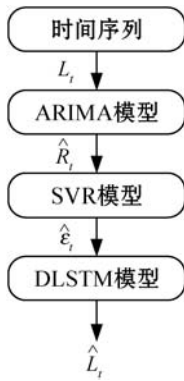


图 4 ARIMA\_DLSTM 模型流程

## 7 实验

### 7.1 实验数据与预处理

本文采用 5 组来自不同领域的时间序列数据集进行实证研究, 分别为 Canadian Lynx、Colorado River flow、Airline Passengers、比特币兑美元汇率和国内糖期货价格指数。以上具有不同统计特征的数据集在时间序列预测研究的文献中被广泛应用<sup>[7,14-15]</sup>。

首先对 Canadian Lynx 数据集进行对数化处理, 即将所有值取常用对数, 再对每个数据集按 4:1 的比例划分为训练集和测试集, 训练集用于训练模型, 测试集用于评估模型性能。Canadian Lynx 数据集共有 114 个样本, 其中前 100 个样本作为训练集, 后 14 个样本作为测试集。Colorado River flow 数据集共有 744 个样本, 前 595 个样本作为训练集, 后 149 个样本作为测试集。Airline Passengers 数据集共有 144 个样本, 前 115 个样本作为训练集, 后 29 个样本作为测试集。比特币兑美元汇率采用 2014 年 12 月至 2019 年 7 月间的数据, 共有 1 672 个样本, 前 1 472 个样本作为训练集, 后 200 个样本作为测试集。国内糖期货价格指数采用 2006 年 3 月至 2019 年 4 月间共 3 190 个样本, 前 3 000 个样本作为训练集, 后 190 个样本作为测试集。

在神经网络训练中, 数据间的量纲差别对网络训练是否能收敛以及预测准确性起到关键作用<sup>[16]</sup>。因此, 在建模前需要对输入数据进行预处理, 本文利用式 (15) 将样本数据归一化至  $[0, 1]$  区间。

$$I(t)' = \frac{I(t) - \min I(t)}{\max I(t) - \min I(t)} \quad (15)$$

式中:  $\min I(t)$  和  $\max I(t)$  分别为训练数据集的最小值

和最大值。训练后的输出数据进行反归一化处理, 以便得到预测值。

为提高模型泛化能力, 采用如下两种方式避免过拟合。首先, 采用失活 (dropout) 正则化, 在网络训练的每次更新中, 以一定失活率随机选择一部分单元失活, 包括输入连接和递归连接, 可以有效防止过拟合。如果采用深度 LSTM 网络, 可以同时每层间采用失活正则化, 所以每层网络共有三个失活参数。其次采用早停法, 将训练样本划分为训练集和验证集, 在每次迭代中, 分别计算训练集和验证集的损失值, 如果验证集损失值在步数  $k$  内不再减少, 则停止训练, 并返回最低验证损失值的模型参数。本文中使用两层 LSTM 神经网络, 因此共有六个失活参数。在训练集中, 80% 样本作为训练集, 20% 样本作为验证集, 步数  $k$  设置为 50。

### 7.2 衡量指标

为评估 ARIMA\_DLSTM 模型的预测性能, 本文采用均方误差 (MSE)、平均绝对百分比误差 (MAPE) 和平均绝对误差 (MAE) 作为衡量预测精度的指标。计算公式分别为:

$$MSE = \frac{1}{N} \sum_{t=1}^N (X_t - F_t)^2 \quad (16)$$

$$MAPE = \frac{1}{N} \sum_{t=1}^N \left| \frac{X_t - F_t}{X_t} \right| \quad (17)$$

$$MAE = \frac{1}{N} \sum_{t=1}^N |X_t - F_t| \quad (18)$$

式中:  $X_t$  表示实际值;  $F_t$  表示预测值;  $N$  为时间序列数据集的样本数量。

### 7.3 实验结果

表 1 中列出 ARIMA\_DLSTM 混合模型中 SVR 模型和 DLSTM 模型的参数及其选择范围。SVR 模型采用 RBF 核函数, 超参数包括惩罚系数  $C$ 、不敏感损失系数  $\varepsilon$ 、宽度系数  $\gamma$  和时间步 timestep, 可以通过网格搜索进行选择, 时间步数为输入序列的长度。DLSTM 模型的层数为 2, 超参数包括每层神经元数量和 dropout 参数, 其中 dropout 参数包括输入连接、递归连接和每层连接之间失活参数, 所以 2 层网络共有 6 个参数, 并采用贝叶斯优化算法<sup>[17]</sup> 进行超参数选择, 算法迭代次数为 50。为评估 ARIMA\_DLSTM 混合模型的性能, 将 ARIMA<sup>[7]</sup>、ARIMAMLP<sup>[18]</sup>、ARIMA\_MLP<sup>[7]</sup>、ARIMA\_SVR<sup>[8]</sup> 和 ARIMA\_SVR\_s<sup>[1]</sup> 预测模型作为对比模型。

表 1 模型参数及取值范围

模型	参数	范围
SVR	kernel	RBF
	$C$	0.1,1,100,1 000,10 000
	$\epsilon$	0.1, 0.01, 0.001
	$\gamma$	1,0.1,0.01,0.001
	timestep	[1,50]
DLSTM	layers	2
	layer1	[1,40]
	layer2	[1,40]
	dropout	[0.01,0.40]
	timestep	[1,50]

(1) Canadian Lynx 数据集。在 SVR 模型中,惩罚系数  $C$  为 1 000,不敏感损失系数  $\epsilon$  为 0.1,宽度系数  $\gamma$  为 1.0,时间步(输入序列长度)为 10。LSTM 模型的每层神经元个数分别为 26 和 21,dropout 为 [0.10, 0.26,0.16,0.12,0.27,0.26],时间步为 5,迭代次数为 2 000 次。

图 5 给出利用贝叶斯优化算法对 LSTM 模型进行超参数优化的最优验证误差迭代图。可以看出,算法在第 42 次迭代后误差值最优。

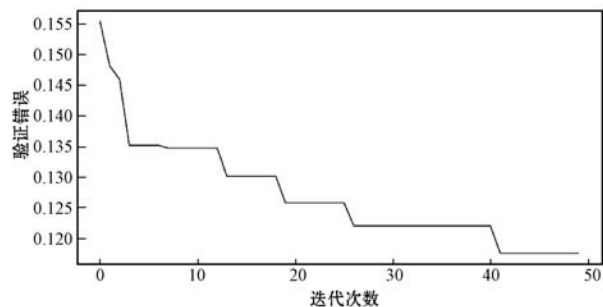


图 5 贝叶斯优化算法在 Canadian Lynx 的验证误差迭代图

表 2 给出了不同模型在 Canadian Lynx 时间序列测试集中的预测结果。可见,ARIMA\_SVR 模型在比较模型中最优,而 ARIMA\_DLSTM 模型的测试结果比 ARIMA\_SVR 模型 MSE、MAPE 和 MAE 值分别降低 53.47%、24.84% 和 26.80%。

表 2 对 Canadian Lynx 时间序列预测结果

模型	MSE	MAPE	MAE
ARIMA	0.020 4	—	0.112 2
ARIMAMLP	0.018 7	4.20	0.125 0
ARIMA_MLP	0.017 2	—	0.103 9
ARIMA_SVR	0.014 4	3.10	0.094 4
ARIMA_SVR_s	0.020 4	4.02	0.121 3
ARIMA_DLSTM	<b>0.006 7</b>	<b>2.33</b>	<b>0.069 1</b>

图 6 给出 ARIMA 模型和 ARIMA\_DLSTM 模型在 Canadian Lynx 测试集上的预测结果。由图 6 可知,ARIMA\_DLSTM 模型的预测结果和真实值更加接近,有效提高了 ARIMA 模型的性能。

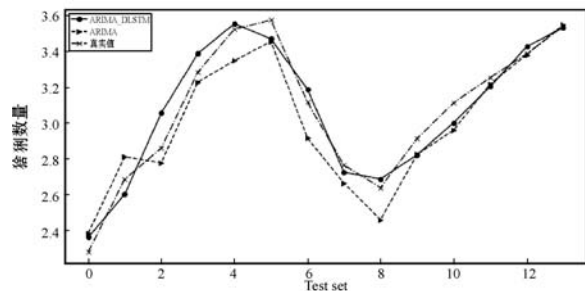


图 6 ARIMA 模型和 ARIMA\_DLSTM 模型在 Canadian Lynx 的时序列测结果

(2) Colorado River flow 数据集。在 SVR 模型中,惩罚系数  $C$  为 1 000,不敏感损失系数  $\epsilon$  为 0.001,宽度系数  $\gamma$  为 0.01,时间步(输入序列长度)为 24。LSTM 模型的每层神经元个数分别为 24 和 34,dropout 为 [0.14,0.32,0.22,0.18,0.28,0.18],时间步为 43,迭代次数为 2 000 次。

图 7 给出利用贝叶斯优化算法对 LSTM 模型进行超参数优化的最优验证误差迭代图。可见,算法在第 30 次迭代后误差值最优。

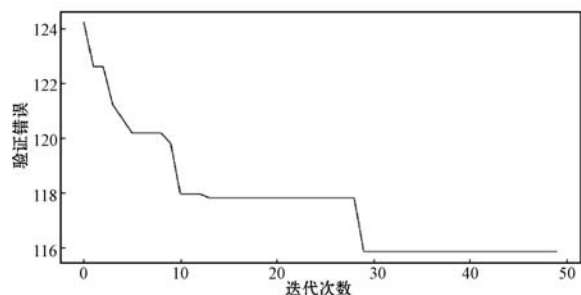


图 7 贝叶斯优化算法在 Colorado River flow 的验证误差迭代图

表 3 给出了不同模型在 Colorado River flow 时间序列测试集中的预测结果。可见,ARIMA\_MLP 模型 MSE 和 MAE 值在比较模型中最优,ARIMA\_SVR\_s 模型的 MAPE 值最优,而 ARIMA\_DLSTM 模型的测试结果比最优 MSE、MAPE 和 MAE 值分别降低 82.15%、41.58% 和 38.26%。

表 3 对 Colorado River flow 时间序列预测结果

模型	MSE	MAPE	MAE
ARIMA	0.286 9	75.44	0.287 9
ARIMAMLP	0.237 6	49.34	0.291 7
ARIMA_MLP	0.223 0	65.16	0.255 9
ARIMA_SVR	0.261 2	33.95	0.316 7
ARIMA_SVR_s	0.286 7	32.56	0.281 0
ARIMA_DLSTM	<b>0.039 8</b>	<b>19.02</b>	<b>0.158 0</b>

图 8 给出 ARIMA 模型和 ARIMA\_DLSTM 模型在 Colorado River flow 测试集上的预测结果。可以看出, ARIMA\_DLSTM 模型预测值存在滞后现象, 但和真实值更接近, 优于 ARIMA 模型。

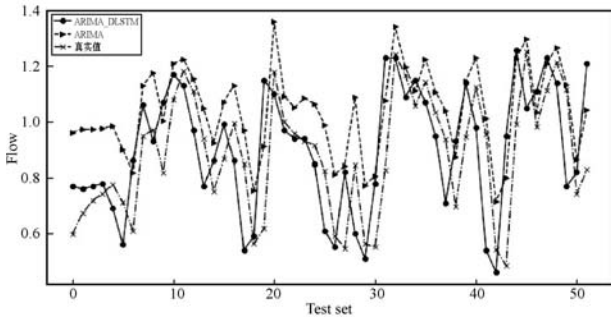


图 8 ARIMA 模型和 ARIMA\_DLSTM 模型在 Colorado River flow 的时序预测结果

(3) Airline Passengers 数据集。在 SVR 模型中, 惩罚系数  $C$  为 1 000, 不敏感损失系数  $\epsilon$  为 0.01, 宽度系数  $\gamma$  为 0.01, 时间步(输入序列长度)为 16。LSTM 模型的每层神经元个数分别为 35 和 22, dropout 为 [0.14, 0.27, 0.11, 0.10, 0.27, 0.25], 时间步为 47, 迭代次数为 2 000 次。

图 9 给出利用贝叶斯优化算法对 LSTM 模型进行超参数优化的最优验证误差迭代图。可以看出, 算法在第 41 次迭代后误差值最优。

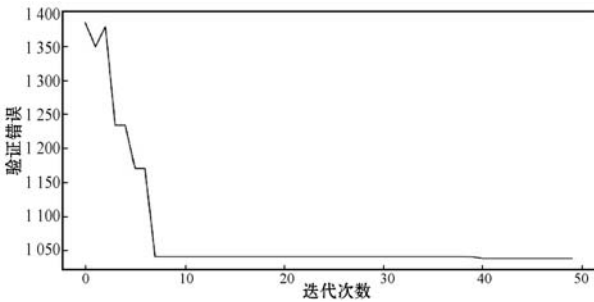


图 9 贝叶斯优化算法在 Airline Passengers 的验证误差迭代图

表 4 给出了不同模型在 Airline Passengers 时间序列测试集中的预测结果。可以看出, ARIMA\_SVR\_s 模型预测结果在比较模型中最优, 而 ARIMA\_DLSTM 模型的测试结果比最优 MSE、MAPE 和 MAE 值分别降低 23.61%、8.09% 和 5.72%。

表 4 对 Airline Passengers 时间序列预测结果

模型	MSE	MAPE	MAE
ARIMA	1 918.6	8.15	36.02
ARIMAMLP	793.3	4.90	22.05
ARIMA_MLP	485.7	3.93	16.59
ARIMA_SVR	405.4	3.81	16.77
ARIMA_SVR_s	388.9	3.71	16.08
ARIMA_DLSTM	<b>297.1</b>	<b>3.41</b>	<b>15.16</b>

图 10 给出 ARIMA 模型和 ARIMA\_DLSTM 模型在 Airline Passengers 测试集上的预测结果。可以看出, ARIMA\_DLSTM 模型预测值和真实值间趋势基本吻合, 有效提高了 ARIMA 模型的预测精度。

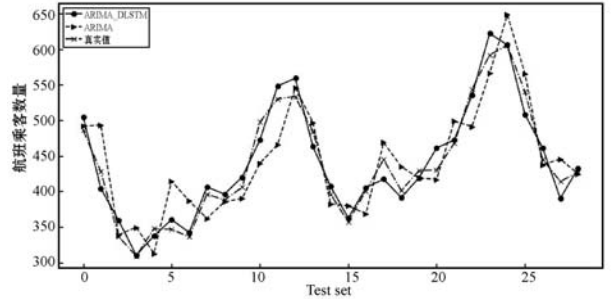


图 10 ARIMA 模型和 ARIMA\_DLSTM 模型在 Airline Passengers 时间序列预测结果

(4) 比特币汇率数据集。在 SVR 模型中, 惩罚系数  $C$  为 1 000, 不敏感损失系数  $\epsilon$  为 0.001, 宽度系数  $\gamma$  为 0.01, 时间步(输入序列长度)为 7。LSTM 模型的每层神经元个数分别为 23 和 22, dropout 为 [0.23, 0.11, 0.17, 0.16, 0.32, 0.16], 时间步为 40, 迭代次数为 2 000 次。

图 11 给出利用贝叶斯优化算法对 LSTM 模型进行超参数优化的最优验证误差迭代图。可以看出, 算法在第 16 次迭代后误差值最优。

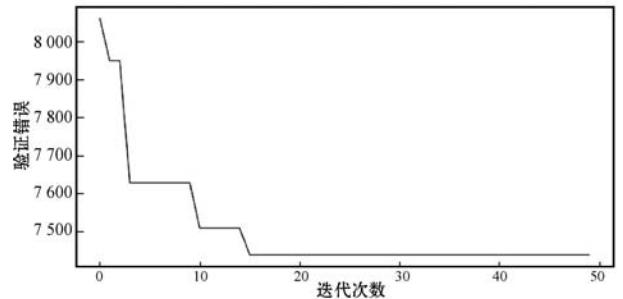


图 11 贝叶斯优化算法在 Bitcoin 汇率的验证误差迭代图

表 5 给出了不同模型在 Bitcoin 汇率时间序列测试集中的预测结果。可以看出, ARIMA\_SVR\_s 模型预测结果在比较模型中最优, 而 ARIMA\_DLSTM 模型的测试结果比最优 MSE、MAPE 和 MAE 值分别降低 17.8%、17.13% 和 10.4%。

表 5 对 Bitcoin 汇率时间序列预测结果

模型	MSE	MAPE	MAE
ARIMA	142 341.9	2.79	218.54
ARIMAMLP	139 805.1	3.38	228.90
ARIMA_MLP	139 370.7	3.30	227.02
ARIMA_SVR	137 082.3	3.23	224.61
ARIMA_SVR_s	136 752.4	3.21	223.23
ARIMA_DLSTM	<b>112 362.7</b>	<b>2.66</b>	<b>200.75</b>

图 12 给出 ARIMA 模型和 ARIMA\_DLSTM 模型在 Bitcoin 汇率测试集上后 20 组数据的预测结果。可以看出,ARIMA\_DLSTM 模型预测值和真实值之间虽有滞后现象,但趋势吻合,能够提高 ARIMA 模型的预测精度。

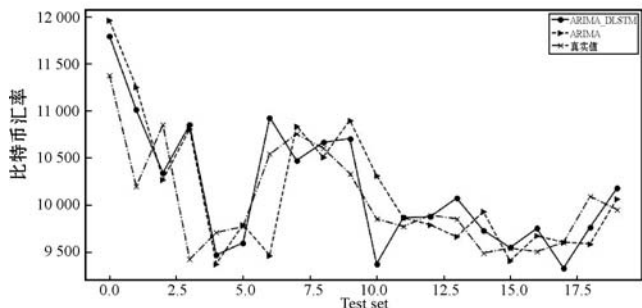


图 12 ARIMA 模型和 ARIMA\_DLSTM 模型在 Bitcoin 汇率时间序列预测结果

(5) 糖价格指数数据集。在 SVR 模型中,惩罚系数  $C$  为 1 000,不敏感损失系数  $\epsilon$  为 0.001,宽度系数  $\gamma$  为 1,时间步(输入序列长度)为 3。LSTM 模型的每层神经元个数分别为 27 和 5,dropout 为 [0.03,0.18,0.19,0.06,0.01,0.3],时间步为 20,迭代次数为 2 000 次。

图 13 给出利用贝叶斯优化算法对 LSTM 模型进行超参数优化的最优验证误差迭代图。可以看出,算法在第 25 次迭代后误差值最优。

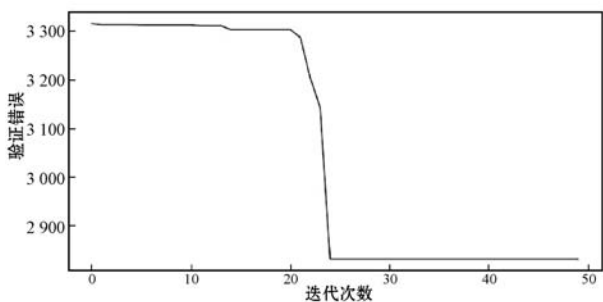


图 13 贝叶斯优化算法在糖价格指数的验证误差迭代图

表 6 给出了不同模型在糖价格指数时间序列测试集中的预测结果。可以看出,ARIMA\_SVR\_s 模型预测结果在比较模型中最优,而 ARIMA\_DLSTM 模型的测试结果比最优 MSE、MAPE 和 MAE 值分别降低 12.48%、3.17% 和 3.88%。

表 6 对糖价格指数时间序列预测结果

模型	MSE	MAPE	MAE
ARIMA	1 961.0	0.66	33.24
ARIMAMLP	1 912.3	0.66	33.41
ARIMA_MLP	1 908.1	0.66	33.49
ARIMA_SVR	1 873.2	0.63	32.35
ARIMA_SVR_s	1 865.7	0.63	32.22
ARIMA_DLSTM	<b>1 632.9</b>	<b>0.61</b>	<b>30.97</b>

图 14 给出 ARIMA 模型和 ARIMA\_DLSTM 模型在糖价格指数测试集上后 20 组数据的预测结果。可以看出,ARIMA\_DLSTM 模型预测值和真实值更接近,精度更高。

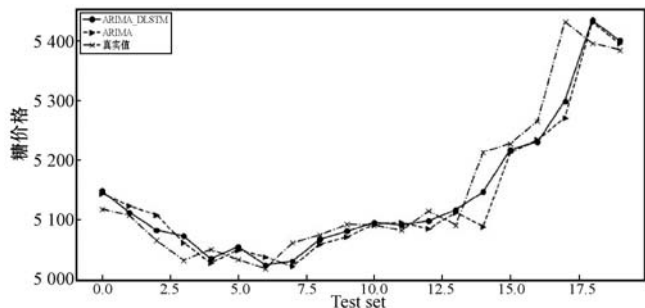


图 14 ARIMA 模型和 ARIMA\_DLSTM 模型在该时间序列预测结果

图 15 给出了 5 个数据集下的 ARIMA 模型和 5 种混合模型 MSE 误差比雷达图,图中曲线的顶点位置越靠近边缘则显示 MSE 误差比越大,说明相应的混合模型相比于 ARIMA 模型 MSE 误差更低。可以看出,ARIMA\_DLSTM 模型在所有时间序列中 MSE 误差最低,其中在 Colorado River flow 时间序列中的误差比最大,优势最为明显。

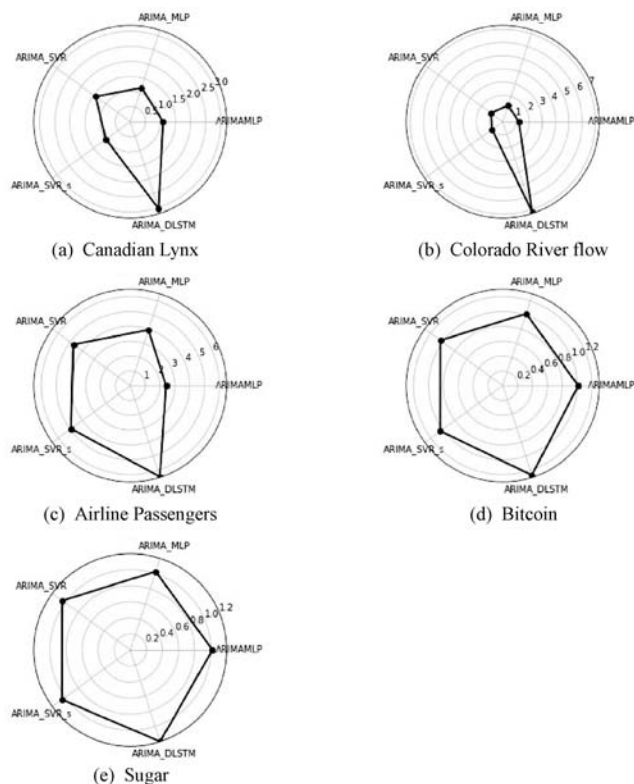


图 15 5 个数据集下的 MSE 误差比雷达图

## 8 结 语

本文提出了一种基于 ARIMA 模型、SVR 模型和深度 LSTM 模型的 ARIMA\_DLSTM 时间序列混合预测模

型。ARIMA 模型和 SVR 模型能够分别提取时间序列的线性特征和误差序列的非线性特征,深度 LSTM 模型对线性和非线性预测结果进行非线性组合。对来自不同领域的时间序列进行实证分析,实验结果表明,提出的 ARIMA\_DLSTM 模型和其他混合模型相比,能够有效提高预测精度,有一定的实际应用价值。

## 参 考 文 献

- [ 1 ] Oliveira J F L, Ludermit T B. A hybrid evolutionary decomposition system for time series forecasting[J]. *Neurocomputing*,2016,180:27 – 34.
- [ 2 ] Shu Y T, Yu M F, Liu J, et al. CQ08-3 wireless traffic modeling and prediction using seasonal ARIMA models [C]//2003 IEEE International Conference on Communications, 2003:1675 – 1679.
- [ 3 ] Liu C H, Hoi S C H, Zhao P L, et al. Online ARIMA algorithms for time series prediction [C]//Proceedings of the 30th AAAI Conference on Artificial Intelligence. ACM, 2016:1867 – 1873.
- [ 4 ] Taskaya-Temizel T, Casey M C. A comparative study of autoregressive neural network hybrids[J]. *Neural Networks*, 2005,18(5/6):781 – 789.
- [ 5 ] Deng Y F, Jin X, Zhong Y X. Ensemble SVR for prediction of time series [C]//2005 International Conference on Machine Learning and Cybernetics. IEEE,2005:3528 – 3534.
- [ 6 ] Panigrahi S, Behera H S. A hybrid ETS – ANN model for time series forecasting[J]. *Engineering Applications of Artificial Intelligence*,2017,66:49 – 59.
- [ 7 ] Zhang G P. Time series forecasting using a hybrid ARIMA and neural network model[J]. *Neurocomputing*,2003,50:159 – 175.
- [ 8 ] Oliveira J F L, Ludermit T B. A hybrid evolutionary system for parameter optimization and lag selection in time series forecasting [C]//2014 Brazilian Conference on Intelligent Systems. IEEE,2014:73 – 78.
- [ 9 ] Vapnik V N. *The nature of statistical learning theory*[M]. Springer,1995.
- [10] Hochreiter S, Schmidhuber J. Long short-term memory[J]. *Neural Computation*,1997,9(8):1735 – 1780.
- [11] 倪铮,梁萍. 基于 LSTM 深度神经网络的精细化气温预报初探[J]. *计算机应用与软件*,2018,35(11):239 – 242,277.
- [12] Lecun Y, Bengio Y, Hinton G E. Deep learning[J]. *Nature*, 2015,521:436 – 444.
- [13] Pelikan M. BOA:The Bayesian optimization algorithm[C]//Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation1999 Genetic and Evolutionary Computation Conference. US: Margan Kaufmann,1999: 525 – 532.
- [14] Wang L, Zou H F, Su J, et al. An ARIMA-ANN hybrid model for time series forecasting[J]. *Systems Research & Behavioral Science*,2013,30(3):244 – 259.
- [15] Oliveira J F L, Ludermit T B. A hybrid evolutionary system for parameter optimization and lag selection in time series forecasting [C]//2014 Brazilian Conference on Intelligent Systems. IEEE, 2014:73 – 78.
- [16] Azadeh A, Asadzadeh S M, Ghanbari A. An adaptive network-based fuzzy inference system for short-term natural gas demand estimation: Uncertain and complex environments [J]. *Energy Policy*,2010,38(3):1529 – 1536.
- [17] Snoek J, Larochelle H, Adams R P. Practical Bayesian optimization of machine learning algorithms[C]//Proceedings of the 25th International Conference on Neural Information Processing Systems. ACM,2012:2951 – 2959.
- [18] Babu C N, Reddy B E. A moving-average filter based hybrid ARIMA-ANN model for forecasting time series data[J]. *Applied Soft Computing*,2014,23:27 – 38.

## (上接第 243 页)

- [ 7 ] 黄德威,陆健强,陈可昕,等. 基于大气光值图的自适应去雾算法[J]. *计算机工程与设计*,2018,39(4):1130 – 1135.
- [ 8 ] Siddig A, Guo Z, Zhou Z, et al. An image denoising model based on a fourth-order nonlinear partial differential equation [J]. *Computers & Mathematics with Applications*, 2018, 76(5):1056 – 1074.
- [ 9 ] Nnolim U A. Partial differential equation-based hazy image contrast enhancement[J]. *Computers & Electrical Engineering*, 2018, 72:670 – 681.
- [10] Bai J, Feng X C. Image denoising using generalized anisotropic diffusion [J]. *Journal of Mathematical Imaging and Vision*, 2018, 60(7): 994 – 1007.
- [11] Pushparaj J, Malarvel M. Panchromatic image denoising by a log-normal-distribution-based anisotropic diffusion model [J]. *Journal of Applied Remote Sensing*,2019,13(1):1 – 15.
- [12] Yin X, Chen S, Wang L, et al. Fractional-order difference curvature-driven fractional anisotropic diffusion equation for image super-resolution[J]. *International Journal of Modeling, Simulation, and Scientific Computing*, 2019, 10(1): 1 – 13.
- [13] Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990, 12(7): 629 – 639.
- [14] Singh D, Kumar V. Dehazing of remote sensing images using improved restoration model based dark channel prior [J]. *The Imaging Science Journal*, 2017, 65(5): 282 – 292.
- [15] Xie C H, Qiao W W, Liu Z, et al. Single image dehazing using kernel regression model and dark channel prior [J]. *Signal, Image and Video Processing*,2017,11(4):705 – 712.