文章编号：2096-1472(2022)-12-01-07
DOI:10.19644/j.cnki.issn2096-1472.2022.012.001

基于机器学习算法的股指期货价格预测模型研究

杨学威

(青海民族大学经济与管理学院，青海 西宁 810007)
✉1661315325@qq.com

摘　要：人工智能技术和量化投资领域的结合，诞生了各类基于机器学习算法的价格预测模型。为研究不同机器学习算法在股指期货价格预测中的应用效果，采用支持向量回归、长短期记忆网络、随机森林及极端梯度提升树四种常用的机器学习算法构建价格预测模型，对沪深300股指期货价格进行预测研究，并利用贝叶斯算法对模型进行超参数优化，对比贝叶斯优化对于以上四种机器学习算法预测精度的提升效果。研究结果表明，随机森林和极端梯度提升树因其模型自身的优点，可以实现对金融时序数据的准确预测，而贝叶斯优化利用高斯过程，不断更新先验，可以显著提高支持向量回归预测效果，均方误差(*MSE*)、平均绝对误差(*MAE*)、对称平均绝对百分比误差(*SMAPE*)和损失适应度(*LOSS*)分别降低了78.6%、94.7%、95.1%和97.0%。

关键词：机器学习；支持向量机；长短期记忆网络；随机森林；极端梯度提升树
中图分类号：TP312　**文献标识码**：A

Research on Stock Index Futures Price Prediction Model based on Machine Learning Algorithms

YANG Xuewei

(*School of Economics and Management*, *Qinghai University for Nationalities*, *Xining* 810007, *China*)
✉1661315325@qq.com

Abstract: With the combination of artificial intelligence technology and quantitative investment, various price prediction models based on machine learning algorithms have emerged. In order to study the effect of different machine learning algorithms on stock index futures price prediction, this paper proposes to use four commonly used machine learning algorithms, namely SVR (Support Vector Regression), LSTM (Long Short-Term Memory), RF (Random Forest) and XGBoost (Extreme Gradient Boosting), to construct a price prediction model, so as to predict the stock index futures price of Shanghai and Shenzhen 300. Bayesian algorithm is used to optimize the hyperparameters of the model, and the improvement effect of Bayesian optimization on the prediction accuracy of the four machine learning algorithms is compared. The research results show that RF and XGBoost can achieve accurate prediction of financial time series data due to their own advantages, while Bayesian optimization can significantly improve the prediction effect of support vector machines by using Gaussian process and constantly updating the prior. MSE, MAE, SMAPE and LOSS are reduced by 78.6%, 94.7%, 95.1% and 97.0% respectively.

Keywords: machine learning; SVR; LSTM; RF; XGBoost

1 引言(Introduction)

宏观经济背景、金融市场发展水平和投资者心理预期等多种复杂因素共同驱动金融工具价格变化，使得金融时序价格具有非平稳性、非线性和高噪声的复杂特性[1]。在国内金融市场高速发展的背景下，金融时序价格预测成为一个亟待解决的难题。伴随着人工智能技术的进步，机器学习算法为金融时序价格预测带来了新的研究思路，学界和业界也致力于运用机器学习算法预测各类金融工具短期趋势并构建量化择

基金项目：青海民族大学研究生创新项目基金(65M2022186)。

时策略，以期获取超额投资收益。

随着国内量化投资的兴起，众多金融机构已将机器学习广泛应用于产品定价、风险管理、量化选股、策略管理等领域。对于非线性、非平稳、更新频率快的金融市场数据，相较于传统统计分析方法，机器学习算法能够迅速挖掘出市场上更多潜在信息。本文选用支持向量回归(Support Vector Regression，SVR)、长短期记忆网络(Long Short-Term Memory，LSTM)、随机森林(Random Forest，RF)、极端梯度提升树(Extreme Gradient Boosting，XGBoost)四种常用的机器学习算法构建沪深300股指期货价格预测模型，并利用贝叶斯算法对模型进行超参数优化，比较其预测效果，为量化择时策略开发提供价格预测基础。

2 机器学习算法(Machine learning algorithms)

2.1 支持向量回归

SVR模型利用支持向量机分类的原理，通过在损失函数中加入松弛变量提高模型回归拟合性能[2]。SVR模型能有效处理多维度样本，能够摆脱神经网络预测模型的局部最优问题，达到唯一的全局最优解。SAPANKEVYCH等[3]系统梳理并总结了SVR模型在时间序列预测的相关研究文献。王洪平[4]运用SVR模型根据金融机构贷款余额预测货币供应量。肖阳等[5]基于三种不同的核函数建立了SVR多因子选股模型，并通过网格搜索和交叉验证法确定了模型参数的最优取值，回测结果表现优异，其中高斯核函数绩效表现最优，年化收益达到24.76%。

2.2 长短期记忆网络

LSTM模型属于循环神经网络(Recurrent Neural Network，RNN)的一种，其特殊之处在于RNN仅有记忆暂存的功能，而LSTM兼具长短期记忆功能，解决了RNN存在的长期依赖问题。自HOCHREITER等[6]提出LSTM之后，GERS等[7]又加入了遗忘门对LSTM结构进行完善，自此形成应用至今的LSTM完整结构。AHMED等[8]将损失函数与LSTM模型组合构建外汇损失函数长短期记忆模型(FLF-LSTM)，预测外汇市场欧元美元汇兑价格，并与其他模型进行对比分析，其研究表明，在外汇市场FLF-LSTM模型预测效果优于其他模型。GIANG等[9]提出了两种基于LSTM的股价预测模型，在美国、德国和越南三个股票数据集上的实验结果表明，此模型在预测股价波动趋势方面优于其他模型。LI等[10]利用差分整合移动平均自回归模型(ARIMA)和LSTM，选取三种股票市场指数的13 个技术指标构建价格预测模型，预测其收盘价并与其他模型进行对比分析，研究结果表明LSTM模型预测精度优于其他模型。

2.3 随机森林

RF作为近年新兴起的、高度灵活的集成算法，具有不易陷入过拟合、抗噪能力强、不用做特征选择、能够平衡误差和处理高维数据且数据集无须标准化及训练速度较快等优点。SIVAMANI等[11]基于社交媒体情感分析，利用包括RF在内的机器学习算法研究投资者情绪对公司股价的影响，研究表明：公众对公司的主观感知可以作为驱动其股价增长的因子。GHOSH等[12]采用RF和LSTM作为训练算法，证明了它们在预测标普成分股价格定向变动方面的有效性。

2.4 极端梯度提升树

XGBoost是陈天奇博士在梯度提升决策树算法的基础上，提出的一种改进算法。XGBoost利用并行化提高其运行速度，同时引入了损失函数的二阶偏导，预测效果更具一般性。衣静[13]通过将集合经验模态分解(EEMD)与XGBoost算法结合，构建了EEMD-XGBoost组合模型，利用模型预测深证综合指数的日收盘价，并对模型进行分析优化。LIU等[14]通过XGBoost筛选评价指标，利用遗传算法优化BP神经网络，构建上证50ETF期权价格预测模型。谷嘉炜等[15]提出XGBoost-ESN的股价预测组合模型，并使用网格搜索法对XGBoost模型和回声状态网络模型(ESN)进行参数优化。研究结果表明，改进的XGBoost-ESN组合模型能有效减少预测误差，对股票价格预测的精度更高。

3 模型基本原理(Model fundamentals)

3.1 SVR模型

SVR模型算法原理如下。

给定一个训练样本集$D=\{(x_i,y_i),i=1,2,\cdots,N\}$，SVR回归模型的目标是让训练集中的每个点$(x_i,y_i)$尽量拟合到一个线性模型：

$$y_i=w^T\phi(x_i)+b \tag{1}$$

其中，$\phi(\cdot)$为映射函数，$\phi(x_i)$是将x映射到高维特征空间的特征向量。

SVR的损失函数度量：

$$err(x_i,y_i)=\begin{cases}0 & |y_i-w\cdot\varphi(x_i)-b|\leqslant\varepsilon\\ |y_i-w\cdot\varphi(x_i)-b|-\varepsilon & |y_i-w\cdot\varphi(x_i)-b|>\varepsilon\end{cases} \tag{2}$$

损失函数度量在加入松弛变量后：

$$\min_{w,b}\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}(\check{\xi}_i+\hat{\xi}_i)$$
$$\text{s.t.}\begin{cases}-\varepsilon-\check{\xi}_i\leqslant y_i-w\cdot\varphi(x_i)-b\leqslant\varepsilon+\hat{\xi}_i\\ \hat{\xi}_i\geqslant 0,\check{\xi}_i\geqslant 0(i=1,\cdots,N)\end{cases} \tag{3}$$

其中，C为惩罚因子；N为训练模型的数目，$\check{\xi}_i$和$\hat{\xi}_i$分别用于度量目标值偏离ε的松弛变量。引入拉格朗日乘子$\{\alpha_i\geqslant 0,i=1,2,\cdots,N\}$，最优化问题(3)可转化为以下二次规划问题：

$$\max_{\alpha_i,\hat{\alpha}_i}\sum_{i=1}^{N}y_i(\hat{\alpha}_i,\alpha_i)-\varepsilon(\hat{\alpha}_i,\alpha_i)-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\hat{\alpha}_i,\alpha_i)(\hat{\alpha}_j,\alpha_j)K(x_i,x_j)$$
$$\text{s.t.}\begin{cases}\sum_{i=1}^{N}y_i(\hat{\alpha}_i,\alpha_i)=0\\ 0\leqslant\alpha_i,\hat{\alpha}_i\leqslant C\\ i=1,\cdots,N\end{cases} \tag{4}$$

其中，$K(\cdot)$为核函数，核函数的选择也是SVR算法的关键问题之一。表1中给出了SVR模型常用的几种核函数。本文选用

金融时序研究经常使用的高斯核径向基函数(*RBF*)，它具有出色的性能，被广泛应用于分类和回归问题。

表1 常用核函数

Tab.1 Commonly used kernel functions

核函数名	代数表达式	参数
多项式核	$K(x_i,x_j)=(x_iy_i+c)^d$	$d\geqslant 1$，d 为多项式次数
高斯核	$K(x_i,x_j)=e^{(-\frac{\|x_i-x_j\|^2}{2\sigma^2})}$	$\sigma>0$，σ 为高斯核的带宽
Sigmoid核	$K(x_i,x_j)=\tanh(\beta x_i^T x_j+\theta)$	$\beta>0$，$\theta<0$

3.2 LSTM模型

LSTM通过对RNN模型结构的优化，能有效避免RNN存在的梯度爆炸或梯度消失问题[16]。LSTM与RNN的区别在于，RNN结构简单，只有一个tanh层，而LSTM内部结构包含四个交互层：遗忘门、输入门、内部记忆单元、输出门。标准RNN结构如图1所示，LSTM结构如图2所示。

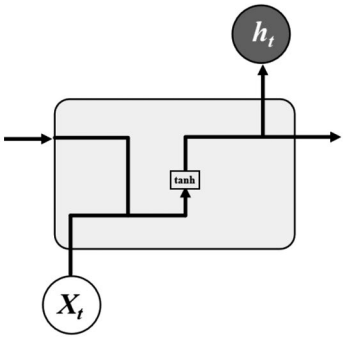

图1 标准RNN结构

Fig.1 Standard RNN structure

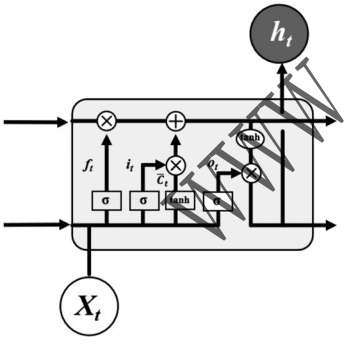

图2 LSTM结构

Fig.2 LSTM structure

LSTM中的第一步是通过遗忘门决定信息的丢弃和保留，其结构如图3所示，算法表达式(5)如下：

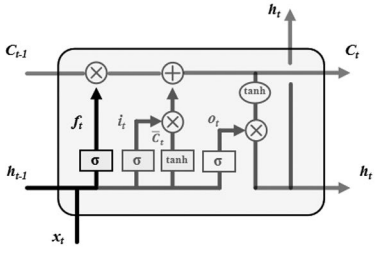

图3 遗忘门结构

Fig.3 Forget gate structure

$$f_t=\sigma(W_f\cdot[h_{t-1},x_t]+b_f) \tag{5}$$

LSTM中的第二步是确定被存放在细胞状态中的新信息，其结构如图4所示，算法表达式(6)如下：

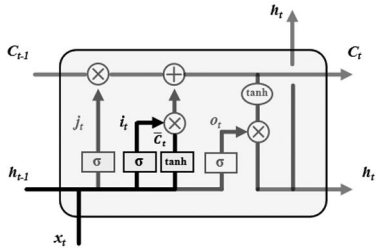

图4 输入门结构

Fig.4 Input gate structure

$$\begin{aligned} i_t&=\sigma(W_i\cdot[h_{t-1},x_t]+b_i)\\ \tilde{C}_t&=\tanh(W_C\cdot[h_{t-1},x_t]+b_C)\end{aligned} \tag{6}$$

LSTM中的第三步是更新细胞状态，将C_{t-1}更新为C_t，其结构如图5所示，算法表达式(7)如下：

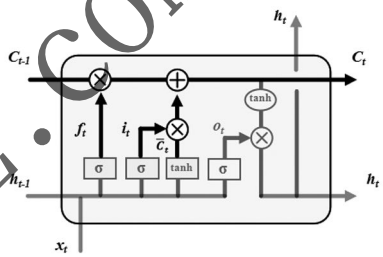

图5 更新状态结构

Fig.5 Update state structure

$$C_t=f_t\cdot C_{t-1}+i_t\cdot\tilde{C}_t \tag{7}$$

LSTM中的第四步是输出信息，输出前需先进行过滤，其结构如图6所示，算法表达式(8)如下：

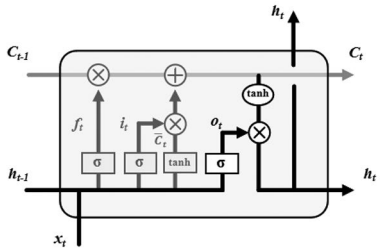

图6 输出门结构

Fig.6 Output gate structure

$$\begin{aligned} o_t&=\sigma(W_o[h_{t-1},x_t]+b_o)\\ h_t&=o_t\cdot\tanh(C_t)\end{aligned} \tag{8}$$

3.3 RF模型

随机森林是一种集成学习算法，即利用引导聚集算法(Bagging)和决策树算法(CART)成决策树的过程；它通过采集多个样本集，利用决策树对每个样本集建模，将所有决策树组合起来构建成随机森林，取所有决策树的结果平均值作为随机森林输出。由于各个决策树之间的具有明显的差异度，因此组合出的随机森林具有良好的泛化能力。随机森林的构造过程如图7所示。

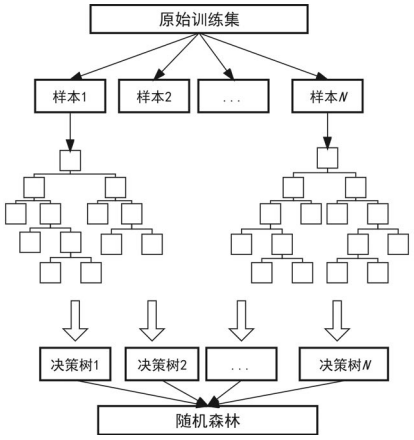

图7 随机森林构造过程

Fig.7 Random Forest construction process

3.4 XGBoost模型

XGBoost模型通过建立K个回归树，使用贪心算法、二次优化保证每个决策树叶子节点的预测值都是最优解，利用交叉验证选择最好的参数，加入正则化防止过拟合；具有效率高、效果好、能处理大规模数据、支持自定义损失函数等优点[17]。

XGBoost属于Boost集成学习方法，应用串行的基学习器，其中第k个学习器的学习目标是前$k-1$个学习器与目标输出的残差，最终的学习器表示如下：

$$\hat{y}^{(k)}=\sum_{k=1}^{t}f_k(x),k=1,2,\cdots,t \tag{9}$$

其中，$f_k(\cdot)$代表编号是k的基回归树，因此，对于输入$\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$，学习第$k$个基学习器时，我们学习的目标函数表示如下：

$$\min_{f_k(x)}(\sum_{i=1}^{n}l(y_i,\hat{y}_i^{(k-1)})+f_k(x_i)) \tag{10}$$

其中，y_i是第i条数据的真实输入，$\hat{y}_i^{(k-1)}$是已经学习的前$k-1$个学习器对i条数据的集成输出，$f_k(\cdot)$是待学习的第k个学习器。

XGBoost的目标函数会加入正则化项，决策树会在后期进行决策树剪枝防止过拟合，加入正则化项后的目标函数如下：

$$\min_{f_k(x),\,\Omega(f_j)}(\sum_{i=1}^{n}l(y_i,\hat{y}_i^{(k-1)})+f_k(x_i)+\sum_{j=1}^{k}\Omega(f_j)) \tag{11}$$

正则化项表示如下：

$$\Omega(f)=\gamma T+\frac{1}{2}\lambda\sum_{t=1}^{T}w_t^{\,2} \tag{12}$$

其中，T是基回归树的叶子节点总数，w_t是基回归树的第t个叶子节点的输出值，γ、λ是正则化项的系数，属于超参数。

当训练第k个基回归树时，前$k-1$个基回归树的正则化项是一个常数，我们单独把它们提取到C(常数)中，目标函数变成：

$$\begin{aligned}&\min_{f_k(x),\,\Omega(f_j)}(\sum_{i=1}^{n}l(y_i,\hat{y}_i^{(k-1)})+f_k(x_i)+\Omega(f_k))+\mathrm{C}=\\&\min_{f_k(x),\,w_t}(\sum_{i=1}^{n}l(y_i,\hat{y}_i^{(k-1)})+f_k(x_i)+\gamma T+\frac{1}{2}\lambda\sum_{t=1}^{T}w_t^{\,2})+\mathrm{C}\end{aligned} \tag{13}$$

其中，y_i与$\hat{y}_i^{(k-1)}$是常量，故损失函数是关于$f_k(x_i)$的函数，对$f_k(x_i)=0$求损失函数的二阶泰勒展开式如下：

$$\begin{aligned}l(y_i,\hat{y}_i^{(k-1)}+f_k(x_i))=&\ l(y_i,\hat{y}_i^{(k-1)})+\frac{\partial l}{\partial f_k(x_i)}\Big|_{f_k(x_i)=0}(f_k(x_i)-0)+\\&\frac{1}{2}\frac{\partial^2 l}{\partial f_k(x_i)^2}\Big|_{f_k(x_i)=0}(f_k(x_i)-0)^2=\\&l(y_i,\hat{y}_i^{(k-1)})+g_i f_k(x_i)+\frac{1}{2}h_i f_k(x_i)^2\end{aligned} \tag{14}$$

其中，$g_i=\dfrac{\partial l(y_i,\hat{y}_i^{(k-1)})}{\partial\hat{y}_i^{(k-1)}}=\partial_{\hat{y}_i^{(k-1)}}l(y_i,\hat{y}_i^{(k-1)})$，

$h_i=\dfrac{\partial^2 l(y_i,\hat{y}_i^{(k-1)})}{\partial(\hat{y}_i^{(k-1)})^2}=\partial^2_{\hat{y}_i^{(k-1)}}l(y_i,\hat{y}_i^{(k-1)})$。 (15)

将上面推出来的损失函数带回目标函数，化简得到公式(16)：

$$\begin{aligned}&\min_{f_k(x),\,w_t}(\sum_{i=1}^{n}l(y_i,\hat{y}_i^{(k-1)})+f_k(x_i)+\gamma T+\frac{1}{2}\lambda\sum_{t=1}^{T}w_t^{\,2})+\mathrm{C}=\\&\min_{f_k(x),\,w_t}(\sum_{i=1}^{n}l(y_i,\hat{y}_i^{(k-1)})+g_i f_k(x_i)+\frac{1}{2}h_i f_k(x_i)^2+\gamma T+\frac{1}{2}\lambda\sum_{t=1}^{T}w_t^{\,2})+\mathrm{C}\end{aligned} \tag{16}$$

其中，$l(y_i,\hat{y}_i^{(k-1)})$是常数，将其融入C后，删去常数C，得到公式(17)：

$$\min_{f_k(x),\,w_t}\sum_{i=1}^{n}(g_i f_k(x_i)+\frac{1}{2}h_i f_k(x_i)^2)+\gamma T+\frac{1}{2}\lambda\sum_{t=1}^{T}w_t^{\,2} \tag{17}$$

其中，g_i与h_i是常数，$f_k(x_i)=w$，改变求和的顺序，且合并$f_k(x_i)$与w得到新的目标函数如下：

$$\min_{w_t}\sum_{t=1}^{T}(\sum_{f_k(x_j)=w_t}g_j)w_t+\frac{1}{2}(\sum_{f_k(x_j)=w_t}h_j+\lambda)w_t^{\,2})+\gamma T \tag{18}$$

定义$G_t=\sum_{f_k(x_j)=w_t}g_j$，$H_t=\sum_{f_k(x_j)=w_t}h_j$，则目标函数进一步变成为公式(19)：

$$\min_{w_t}\sum_{t=1}^{T}(G_t w_t+\frac{1}{2}(H_t+\lambda)w_t^{\,2})+\gamma T \tag{19}$$

这是T个关于w_t的独立二次函数，让每一个二次函数取最小值，即

$$\begin{aligned}&w_t=-\frac{G_t}{H_t+\lambda}(H_t+\lambda>0)\\&\sum_{t=1}^{T}(-\frac{G_t^{\,2}}{2(H_t+\lambda)})+\gamma T\end{aligned} \tag{20}$$

确定基回归树结构的方法主要是递归地确定叶子节点是否适合被延伸。对于某个我们想要延伸的叶子节点t_x，计算其延伸前的目标函数值：

$$obj_1=\sum_{t=1}^{T}(-\frac{G_t^{\,2}}{2(H_t+\lambda)})+\gamma T \tag{21}$$

利用贪心算法遍历所有特征的所有可能取值，计算每个

取值延伸后的目标函数值，t_x分割出两个新的叶子节点t_1与t_2：

$$obj_2=\sum_{t=1}^{T+1}(-\frac{G_t^{\ 2}}{2(H_t+\lambda)})+\gamma(T+1) \tag{22}$$

两者求差，表示分割后的信息增益：

$$obj_1-obj_2=\frac{1}{2}(\frac{G_{t_1}^{\ 2}}{H_{t_1}+\lambda}+\frac{G_{t_2}^{\ 2}}{H_{t_2}+\lambda}-\frac{G_{t_x}^{\ 2}}{H_{t_x}+\lambda})-\gamma \tag{23}$$

取信息增益最大的分割为该叶子节点的最优解。可以设置信息增益取值下限，限制树生长过深，同时通过设置树的最大深度上限防止过拟合。

4 模型建立与超参数优化(Model building and hyperparameters optimization)

本文选取沪深300股指期货主力连续合约(IF9999)自2012 年1 月4 日至2022 年7 月29 日的开盘价、收盘价、最高价、最低价、成交量、成交额作为数据集，以约9∶1的比例将2,569 条交易数据划分为训练集与测试集。本文的训练数据的窗口长度选择收盘价序列ADF检验AIC最小准则计算出的默认滞后阶数25，即用过去25 天的开盘价、收盘价、最高价、最低价、成交量、成交额作为输入特征，未来1 天的收盘价作为标签，进行模型训练。

由于输入指标开盘价、收盘价等与成交量的量纲不同，数量级上的差异会对预测模型收敛带来不利影响，因此需要对原始数据中各项输入指标进行归一化处理，参照常用归一化处理方式如下：

$$x_i=\frac{X_i-\min(X_i)}{\max(X_i)-\min(X_i)} \tag{24}$$

其中，x_i是归一化之后的输入数据，$\max(X_i)$、$\min(X_i)$分别表示该指标中的最大值与最小值。

为了保持输入数据和输出数据在同一量纲，更真实客观地利用模型评价指标对模型预测能力进行评价，需要对输出数据进行反归一化处理，参照常用反归一化处理方式如下：

$$\hat{Y}_i=y_i\times(\max(X_i)-\min(X_i))+\min(X_i) \tag{25}$$

其中，$\hat{Y}_i$是反归一化之后的收盘价预测数据，y_i是归一化之前模型输出的收盘价预测数据。

模型评价指标参照前人时序数据回归算法预测研究经验，选取R^2、均方误差(*MSE*)、平均绝对误差(*MAE*)、对称平均绝对百分误差(*SMAPE*)、适应度函数(*LOSS*)作为模型评价指标：

$$R^2=\sum_{i=1}^{n}(\hat{Y}_i-\overline{Y}_i)^2\Big/\sum_{i=1}^{n}(Y_i-\overline{Y}_i)^2 \tag{26}$$

$$MSE=\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i-Y_i)^2 \tag{27}$$

$$MAE=\frac{1}{n}\sum_{i=1}^{n}|\hat{Y}_i-Y_i| \tag{28}$$

$$SMAPE=\frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{Y}_i-Y_i|}{(|\hat{Y}_i|+|Y_i|)/2} \tag{29}$$

$$LOSS=\frac{1}{4}(-R^2\times10^3+MSE\times10^{-2}+MAE+SMAPE\times10^6) \tag{30}$$

公式(26)—公式(29)中，$\hat{Y}_i$是收盘价预测数据，Y_i是收盘价真实数据；R^2表示预测值占真实值的比率，其值越大模型表现越好；*MSE*、*MAE*、*SMAPE*表示预测值与真实值的预测偏差，其值越小模型表现越好；公式(30)中的*LOSS*作为模型适应度函数，其值越小模型表现越好。

4.1 SVR预测模型构建

在构建SVR预测模型时，利用Python中Scikit-learn模块的SVR类实现回归预测，选择径向基函数(*RBF*)作为核函数，模型在训练集上的预测效果如图8所示。

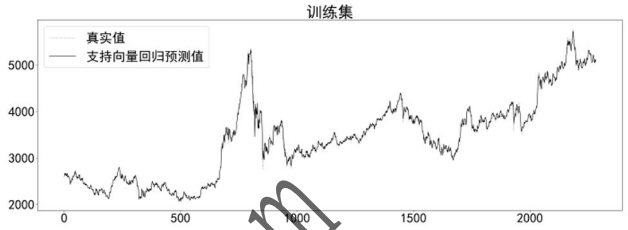

图8 SVR模型在训练集上预测效果

Fig.8 Prediction effect of SVR model on the training set

4.2 LSTM预测模型构建

在构建LSTM预测模型时，利用Python中Keras模块自带的LSTM函数实现回归预测，模型包含输入层、LSTM层、全连接层、输出层。*batch_size*设置为500，迭代100 次，设置全局随机种子，使用Adam优化器进行优化，模型在训练集上的预测效果如图9所示。

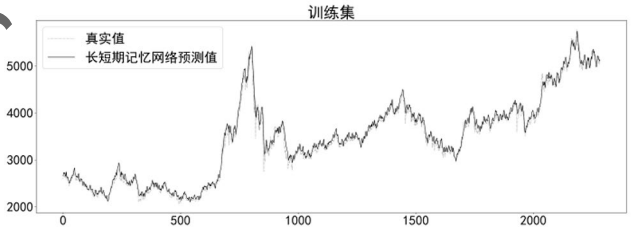

图9 LSTM模型在训练集上预测效果

Fig.9 Prediction effect of LSTM model on the training set

4.3 RF预测模型构建

在构建RF预测模型时，利用Python中Scikit-learn模块的RandomForestRegressor实现回归预测，设置子决策树最大树深*max_depth*为10，子决策树数量*n_estimators*为200，最小样本叶子数量*min_samples_leaf*为20，分割所需最小样本数*min_samples_split*为20，模型在训练集上的预测效果如图10所示。

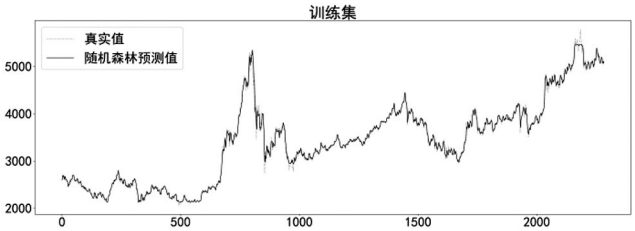

图10 RF模型在训练集上预测效果

Fig.10 Prediction effect of RF model on the training set

4.4 XGBoost预测模型构建

在构建XGBoost预测模型时，利用Python中XGBoost模块的XGBRegressor实现回归预测，设置惩罚项系数*gamma*为200，子决策树最大深度*max_depth*为20，子决策树数量*n_estimators*为100，随机采样比例*subsample*为0.6。模型在训练集上的预测效果如图11所示。

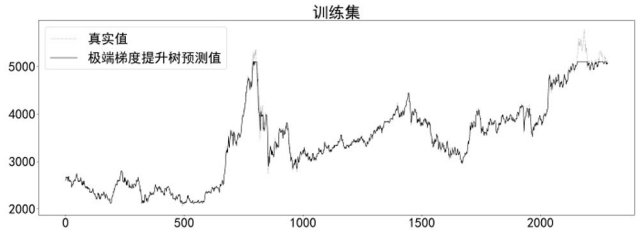

图11 XGBoost模型在训练集上预测效果

Fig.11 Prediction effect of XGBoost model on the training set

4.5 贝叶斯优化

贝叶斯优化算法是一种“黑箱”算法，不需要提前设置目标函数的表达式，即可寻找全局最优解，非常适合本文四种算法的超参数优化。

算法的主体框架不断迭代目标函数后验概率分布与极小值点的过程，使目标函数极小值不断减小，最终得到最优超参数。算法思路如下。

第一步：定义优化目标。

$$x_{\min}=\arg\min_{x\in X} f(x) \tag{31}$$

其中，$x_{\min}$是待优化的超参数，$f(x)$是待优化的目标函数。

第二步：对观测点进行高斯过程处理。

$$f(x_{1:t})\sim GP(\mu(x_{1:t}),\sum(x_{1:t},x_{1:t})) \tag{32}$$

其中，$\sum(x_{1:t},x_{1:t})$是超参数协方差矩阵。

根据贝叶斯定理，可以得到：

$$P(f(x_{t+1})\mid f_{1:t})\propto P(f(x_{1:t})\mid f_{t+1})P(f(x_{t+1})) \tag{33}$$

第三步：不断循环上述过程，最终实现$x_{\min}=x_{t+1}$，可得到最优化的超参数。

按照上述贝叶斯优化过程，构建优化流程如图12所示。

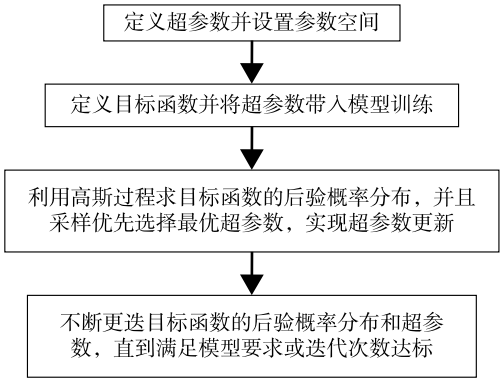

图12 贝叶斯优化过程流程图

Fig.12 Flowchart of Bayesian Optimization process

贝叶斯优化过程中，设置优化次数为200次。SVR模型优化参数为惩罚因子*C*和核函数参数*gamma*；LSTM模型优化参数为全连接层叶子节点数*dense_units*、学习率*learning_rate*、隐含层叶子节点数*lstm_units*_1；RF模型优化参数为子决策树最大深度*max_depth*、单个决策树使用特征比例*max_features*、最小样本叶子数量*min_samples_leaf*、分割所需最小样本数*min_samples_split*、子决策树数量*n_estimators*；XGBoost模型优化参数为惩罚项系数*gamma*、学习率*learning_rate*、子决策树最大深度*max_depth*、最小叶子节点样本权重和*min_child_weight*、子决策树数量*n_estimators*、L1正则化系数*reg_alpha*、L2正则化系数*reg_lambda*、随机采样比例*subsample*。各个模型优化后在测试集上的表现如图13至图16所示。

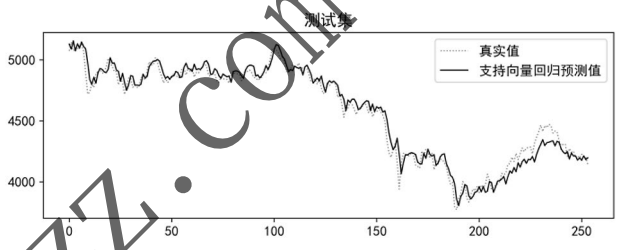

图13 优化后SVR模型在测试集上预测效果

Fig.13 Prediction effect of the optimized SVR model on the test set

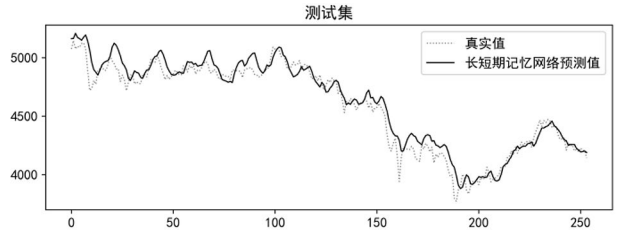

图14 优化后LSTM模型在测试集上预测效果

Fig.14 Prediction effect of the optimized LSTM model on the test set

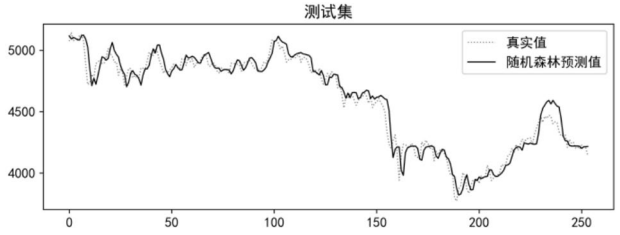

图15 优化后RF模型在测试集上预测效果

Fig.15 Prediction effect of the optimized RF model on the test set

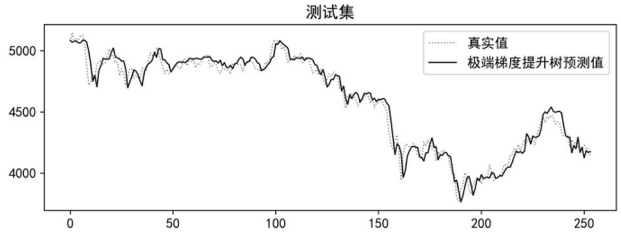

图16 优化后XGBoost模型在测试集上预测效果

Fig.16 The predicted effect of the optimized XGBoost model on the test set

5 优化后结果分析(Analysis of the results after optimization)

利用贝叶斯优化对各个算法模型进行优化训练后，其超参数选择如表2所示。

表2 贝叶斯优化后超参数选择

Tab.2 Hyperparameters selection after Bayesian optimization

算法模型	参数说明	符号表示	参数取值
SVR	惩罚因子	*C*	9,706
	核函数参数	*gamma*	0.0008
LSTM	全连接层叶子节点数	*dense_units*	121
	学习率	*learning_rate*	0.0013
	隐含层叶子节点数	*lstm_units_1*	350
RF	子决策树最大深度	*max_depth*	12
	单个决策树使用特征比例	*max_features*	0.9154
	最小样本叶子数量	*min_samples_leaf*	18
	分割所需最小样本数	*min_samples_split*	40
	子决策树数量	*n_estimators*	360
XGBoost	惩罚项系数	*gamma*	203
	学习率	*learning_rate*	0.0610
	子决策树最大深度	*max_depth*	5
	最小叶子节点样本权重和	*min_child_weight*	75
	子决策树数量	*n_estimators*	145
	L1正则化系数	*reg_alpha*	107
	L2正则化系数	*reg_lambda*	198
	随机采样比例	*subsample*	0.6090

根据表3可以看出，优化前RF和XGBoost预测效果显著优于SVR和LSTM，优化后各个算法模型预测效果较为均衡，而贝叶斯优化对于SVR算法预测效果提升最为明显。SVR算法经过贝叶斯优化后，*MSE*降低了99.54%，*MAE*降低了94.63%，*SMAPE*降低了95.06%。可以看出，优化后的SVR算法预测值最接近真实值，预测精度最高。

表3 优化前后评价指标对比

Tab.3 Comparison of evaluation indicators before and after optimization

算法模型	R^2		*MSE*		*MAE*		*SMAPE*		*LOSS*	
	优化前	优化后	优化前	优化后	优化前	优化后	优化前	优化后	优化前	优化后
SVR	−5.1004	0.9718	823,320	3,810	819	44	0.0668	0.0033	20,241	606
LSTM	0.9193	0.9437	10,881	7,592	**84**	64	0.0061	0.0046	1,366	973
RF	0.9553	0.9689	**6,031**	4,190	**57**	47	0.0042	0.0034	861	650
XGBoost	0.9410	0.9712	**7,950**	3,878	**67**	47	0.0049	0.0034	1,031	647

6 结论(Conclusion)

本文利用四种机器学习算法对沪深300股指期货主力连续合约收盘价进行预测研究，验证了机器学习算法对金融时序数据预测的可行性。通过对比贝叶斯优化前后预测效果，验证了贝叶斯优化对于机器学习算法预测效果提升的可得性。研究结果表明，RF和XGBoost可以实现对金融时序数据的准确预测，而贝叶斯优化可以显著提升SVR算法的预测精度。

本文具有选用输入指标过于简略、模型优化方法较为单一的局限性，可通过引入价值、技术、动量、反转、情绪等多种指标构建模型输入，引入遗传算法、粒子群算法、鲸鱼优化算法等多种超参数优化方法提高模型鲁棒性。

参考文献(References)

[1] 何光辉.处理金融时间序列的非平稳性和时变性[J].国际金融研究,2004(05):74−78.

[2] 杨向前,欧阳鹏.基于VMD和Attention−LSTM 的金融时间序列预测[J].软件,2020,41(12):142−149.

[3] SAPANKEVYCH N I, SANKAR R. Time series prediction using support vector machines: a survey[J]. IEEE Comptational Intelligence Magazine, 2009, 4(2):24−38.

[4] 王洪平.基于支持向量机的我国货币供应量预测[J].金融理论与教学,2020(05):12−15.

[5] 肖阳,丁琦.基于三种核函数的SVM选股模型的实证分析[J].中国商论,2020(15):56−58.

[6] HOCHREITER S, SCHMIDHUBER J. Long short−term memory[J]. Neural computation, 1997, 9(8):1735−1780.

[7] GERS F A, SCHMIDHUBER J, CUMMINS F. Learning to forget continual prediction with LSTM[J]. Neural computation, 2000, 12(10):2451−2471.

[8] AHMED S, HASSAN S U, ALJOHANI N R, et al. FLF−LSTM: A novel prediction system using forex loss function[J]. Applied Soft Computing Journal, 2020, 97:106780.

[9] GIANG T T H, NGUYEN T T, LE Q T. Dynamic sliding window and neighborhood LSTM-based model for stock price prediction[J]. SN Computer Science, 2022, 3(3):1-14.

[10] LI Z J, LIAO Y P, HU B, et al. A financial deep learning framework: predicting the values of financial time series with ARIMA and LSTM[J]. International Journal of Web Services Research (IJWSR), 2022,19(1):1-15.

[11] SIVAMANI B A, KARTHIKEYAN D, ARUMUGAM C, et al. Time series for forecasting stock market prices based on sentiment analysis of social media[J]. International Journal of Business Strategy and Automation (IJBSA), 2022, 2(2): 484-495.

[12] GHOSH P, NEUFELD A, SAHOO J K. Forecasting directional movements of stock prices for intraday trading using LSTM and random forests[J]. Finance Research Letters, 2022(46):102280.

[13] 衣静.基于EEMD和XGBoost算法的股票市场分析预测研究[D].济南:山东大学,2020.

[14] LIU X Z. High frequency price duration prediction of option based on XGBoost and GA-BP[J]. Frontiers in Economics and Management, 2021, 2(5):290-300.

[15] 谷嘉炜,韦慧.XGBoost-ESN组合模型股价预测方法[J].牡丹江师范学院学报(自然科学版),2022(01):1-5.

[16] 贺毅岳,李萍,韩进博.基于CEEMDAN-LSTM的股票市场指数预测建模研究[J].统计与信息论坛,2020,35(06):34-45.

[17] 林升,慕科,魏楷聪,等.机器学习在股价预测中的研究综述[J].经济师,2019(03):71-73,78.

作者简介：

杨学威(1996-)，男，硕士生.研究领域：量化投资.

(上接第12页)

[4] 方景芳,袁冲.基于车间道路约束的物料配送模型及算法研究[J].电子设计工程,2020,28(23):18-24.

[5] 赵炳巍,贾峰,曹岩,等.基于模拟退火算法的人工势场法路径规划研究[J].计算机工程与科学,2022,44(04):746-752.

[6] 孙鉴,刘淞佐,武晓晓,等.基于Spark的并行模拟退火算法求解TSP[J].电子测量技术,2022,45(04):53-58.

[7] 柳伍生,李旺,周清,等.“无人机-车辆”配送路径优化模型与算法[J].交通运输系统工程与信息,2021,21(06):176-186.

[8] 王星童,吴林鸿,赵启宇,等.粒子群-快速模拟退火算法在路径规划中的应用[J].信息技术与信息化,2021(06):13-16.

[9] 刘雪燕.沙漠光伏电池板清洗机器人的路径规划研究和设计[J].光源与照明,2021(06):67-68.

[10] WEN M, LARSEN J, CLAUSEN J, et al. Vehicle routing with cross-docking[J]. Journal of the Operational Research Society, 2008, 60(12):1708-1718.

作者简介：

王国娜(1999-)，女，硕士生.研究领域：农村区域发展.本文通信作者.

唐小平(1981-)，男，博士，副教授.研究领域：农业与农村经济发展研究.

(上接第29页)

[13] 黄健.基于深度学习与二维离散小波分解特征相融合的adaboost人脸识别模型[J].软件工程,2020,23(2):43-46.

[14] MALLAT S. A theory for multiresolution signal decomposition: the wavelet representation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989, 11(7):674-693.

[15] GAO S, CHENG M, ZHAO K, et al. Res2net: A new multi-scale backbone architecture[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 43(2):652-662.

[16] LIU Z, LUO P, WANG X, et al. Deep Learning Face Attributes in the Wild[C]// IEEE International Conference on Computer Vision. Santiago: IEEE, 2015:3730-3738.

[17] KARRAS T, LAINE S, AILA T. A style-based generator architecture for generative adversarial networks[C]// IEEE Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019:4401-4410.

[18] 夏皓,吕宏峰,罗军,等.图像超分辨率深度学习研究及应用进展[J].计算机工程与应用,2021,57(24):51-60.

[19] LIU S, XIONG C, SHI X, et al. Progressive face super-resolution with cascaded recurrent convolutional network[J]. Neurocomputing, 2021, 449:357-367.

[20] ZHANG K, ZHANG Z, CHENG C, et al. Super-identity convolutional neural network for face hallucination[C]// European Conference on Computer Vision. Munich: Springer, 2018:183-198.

[21] HUANG H, HE R, SUN Z, et al. Wavelet-srnet: A wavelet-based cnn for multi-scale face super resolution[C]// IEEE International Conference on Computer Vision. Venice: IEEE, 2017:1689-1697.

作者简介：

李洁沁(1994-)，女，硕士，讲师.研究领域：图像处理，计算机视觉.

谢丁峰(1978-)，男，硕士，副教授.研究领域：数据挖掘，大数据技术.