

Article

A Multi-Factor Selection and Fusion Method through the CNN-LSTM Network for Dynamic Price Forecasting

Yishun Liu, Chunhua Yang, Keke Huang * and Weiping Liu

School of Automation, Central South University, Changsha 410083, China

* Correspondence: huangkeke@csu.edu.cn

Abstract: Commodity prices are important factors for investment management and policy-making, and price forecasting can help in making better business decisions. Due to the complex and volatile nature of the market, commodity prices tend to change frequently and fluctuate violently, often influenced by many potential factors with strong nonstationary and nonlinear characteristics. Thus, it is difficult to obtain satisfactory prediction effects by only using the historical data of prices individually. To address this problem, a novel dynamic price forecasting method based on multi-factor selection and fusion with CNN-LSTM is proposed. First, the factors related to commodity price are collected, and Granger causality inference is used to identify causal factors that affect the commodity price. Then, XGBoost is used to evaluate the importance of the remaining factors and screen out critical factors to reduce the interference of redundant information. Due to the high amount and complicated changes of the selected factors, a convolutional neural network is employed to fuse the selected factors and extract the hidden features. Finally, a long short-term memory network is adopted to establish a multi-input predictor to obtain the dynamic price. Compared with several advanced approaches, the evaluation results indicate that the proposed method has an excellent performance in dynamic price forecasting.



Citation: Liu, Y.; Yang, C.; Huang, K.; Liu, W. A Multi-Factor Selection and Fusion Method through the CNN-LSTM Network for Dynamic Price Forecasting. *Mathematics* **2023**, *11*, 1132. <https://doi.org/10.3390/math11051132>

Academic Editors: Huawei Liu, Chengyuan Zhang, Weiren Yu and Chunwei Tian

Received: 3 February 2023

Revised: 19 February 2023

Accepted: 21 February 2023

Published: 24 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: price forecasting; multi-factor selection; information fusion; long short-term memory network

MSC: 68T07

1. Introduction

Commodities are important basic raw materials for industrial and agricultural production, including crude oil, non-ferrous metals, steel, coal, etc. [1]. They have industrial attributes, as well as typical financial attributes. The commodity price is an important basis for investment management, business decision-making, and policy-making [2]. It is usually affected by multiple hidden variables, such as the global economy, supply-demand relationship, exchange rate, and so on [3,4]. With the development of economic globalization, the financial market, as a highly complex nonlinear dynamic system, has become more volatile [5]. Commodity prices usually fluctuate frequently, change widely, and exhibit strong nonstationary and nonlinear characteristics [6]. The inherent volatility and uncertainty of data changes bring many difficulties to high-precision price forecasting [7]. Therefore, accurate and robust price forecasting has become an important issue.

In the past decades, commodity price forecasting, which was focused on price analysis and prediction, has given rise to extensive attention. In general, these methods can be roughly divided into three categories: chaotic economics methods, statistical methods, and artificial intelligence methods. For chaotic economics methods, they use the nonlinear chaos theory to analyze and model price series. Rodríguez et al. adopted the multi-scroll Chua system to identify the Colombian coffee price dynamics, and employed artificial bee colony optimization to fine-tune the model [8]. Yuan et al. proposed an improved

multiplicative volatility approach to analyze the stock market price [9]. Wang et al. used the partial differential equation of the bitcoin trading network to analyze the changes in the bitcoin price [10]. Frezza assumed that the price followed a multi-fractional process with a random exponent to model the fluctuation of stock price [11]. Chaotic economics can take into account the impact of complex relationships, but it is too sensitive to parameters and initial conditions.

For statistical methods, Krzysztof employed various Bayesian models to predict the spot price of nickel, lead, and zinc together with dynamic model averaging (DMA) [12]. Zhu et al. extended the leverage heterogeneous autoregressive model with continuous volatility and jump (LHAR-CJ) with generalized autoregressive conditional heteroscedasticity (GARCH) to predict the Chinese nonferrous metals futures market volatility [13]. Thomas et al. combined wavelet-based multi-resolution analysis with autoregressive integrated moving average (ARIMA) models to forecast the monthly base metal price [14]. Sahinli adopted Holt–Winters multiplicative and additive methods to explore the future trend of potato prices in Turkey [15]. Hesam and Dejan chose the Brownian motion with mean reversion (BMMR) to estimate the copper price and used the bat algorithm to optimize the parameters [16]. Although the statistics-based methods can accomplish the general task of commodity price prediction, it is hard to deal with sequences with strong nonlinearity and time-varying characteristics.

With the prosperity of artificial intelligence and the advent of the big data era, many data-driven machine learning methods have emerged and are widely applied [17,18]. Astudillo et al. used the support vector regression (SVR) technique to make long-term predictions for copper prices [19]. Diego and Werner developed an adaptive hybrid forecasting model for copper price volatility together with GARCH and the fuzzy inference system (FIS) [20]. Zakaria et al. adopted the adaptive neuro-fuzzy inference system (ANFIS) to predict the volatility of the copper price and optimized the parameters in ANFIS through the genetic algorithm (GA), which effectively improved the prediction accuracy [21]. Machine learning approaches can effectively capture the nonlinearity and irregularity of price series, so they have good prediction performance.

In recent years, deep learning models have proven to be the most promising tools for time series forecasting. A neural network can learn from sample data and approximate any nonlinear function with arbitrary precision, so it usually has satisfactory results [22,23]. Chen et al. combined the residual with the extreme learning machine and proposed a deep residual compensation extreme learning machine model (DRC-ELM), which was used in the regression analysis of gold price [24]. Atsalakis et al. adopted the neuro-fuzzy controller to predict the change direction of daily Bitcoin price for investment trade [25]. Kamdem et al. adopted long short-term memory (LSTM) to predict commodity prices, such as for crude oil, and analyzed the correlation between COVID-19 and commodity price [26]. Wang and Li used the artificial neural network (ANN) to analyze the gold future in the New York Commodity Exchange COMEX [27]. Ugurlu et al. modified the traditional recurrent neural network and proposed a multi-layer gated recurrent unit to predict the Turkish electricity market price [28].

Furthermore, considering the unavoidable shortcomings of a single model in dealing with complex time series, many scholars combined multiple methods to generate synergistic effects and improve the overall forecasting performance. Werner and Esteban introduced ANN into GARCH with regressors to forecast the price of gold, silver, and copper, and the incorporation promoted the forecasting accuracy [29]. Ana et al. adopted a combined model based on the recurrent neural network (RNN) and graph convolutional network (GCN) to predict real-time oil prices [30]. Hu et al. proposed a hybrid deep learning method by integrating the LSTM-ANN network with the GARCH model for copper price volatility prediction [31]. Livieris et al. adopted convolutional layers and the LSTM network to analyze and forecast the daily gold price [32]. Marian employed discrete wavelet transform with support vector regression to predict gold-price dynamics [33]. Hu et al. proposed a hybrid carbon price forecasting method for multimodal carbon emission trading

market combining complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) and a windowed-based XGBoost approach [34]. Zhang and Liao employed the principal component analysis (PCA) and the hybrid fuzzy clustering algorithm to integrate technical indicators and adopted the radial basis function (RBF) neural network as the predictor for gold prices [35]. In general, hybrid models have better performance.

Although the existing methods have achieved good results, most of them only analyze their own historical data. They ignore the very important fact that commodity price is affected by many hidden factors, such as economic situations, transaction statuses, and so on. In the pattern of only using historical data to make forecasting, the information considered is relatively one-sided. In the multivariate forecasting framework, the auxiliary information of multiple factors is effectively used to make it possible to model prices more accurately, so it has a very large development space for improving the prediction performance [36,37]. However, as far as we know, there is relatively little research in the field of multivariate forecasting for commodity prices.

To this end, a novel commodity price forecasting method based on multi-factor selection and fusion together with the convolutional neural network (CNN) and long short-term memory (LSTM) network is proposed. Firstly, the factors that may be related to the change in commodity price were collected, and Granger causality inference was adopted to collect the causal factors. Next, extreme gradient boosting (XGBoost) was used to evaluate the importance of the remaining factors and screen out the most important factors. Then, in order to deeply explore the potential variation characteristics, the CNN was employed for factor fusion and feature extraction to reduce the burden of the predictor. Finally, considering the superiority of LSTM in sequence processing, it was adopted to build a multi-input long-term forecasting model to obtain the future price. Compared with several advanced methods, the proposed forecasting method takes into account the influence of external contributors, carries out screening and fusion processing, and has the best performance in general. In summary, the main contributions of this paper are as follows:

- In order to distinguish the core components of exogenous variables, a two-layer factor selection method based on the Granger causality inference and XGBoost is proposed.
- Utilizing the advantages of a CNN in hidden feature extraction and LSTM in time series processing, a multi-factor hybrid price forecasting model is proposed.
- Through the application of the proposed factor selection method in the SMM 0# zinc price, the conclusion further confirms the impact of the London Metal Exchange (LME) on Shanghai Metals Market (SMM). This provides a strong basis for the prediction and analysis of the zinc price.
- Compared with several advanced approaches, the realistic experiments show the superiority of the proposed method.

The rest of this paper is organized as follows. In Section 2, the methods related to this work are introduced. Section 3 describes the proposed method in detail. The experimental design and comparative analysis are given in Section 4. Finally, Section 5 presents the concluding remarks.

2. Preliminaries

2.1. Granger Causality Inference

The Granger causality inference (GCI) is a classical method that can measure the interaction between different time series [38]. It has been widely used in economics, neuroscience, and other fields in recent decades. For sequences $f(t)$ and $u(t)$, if the prediction effect of $f(t)$ with the past information of $f(t)$ and $u(t)$ is better than that alone with the past information of $f(t)$, sequence $u(t)$ helps to explain the future change of sequence $f(t)$. Therefore, $u(t)$ is considered to be the Granger cause of $f(t)$. The Granger causality is not the relationship between cause and effect as we usually understand it to be; it declares that the previous change of $u(t)$ can effectively explain the future change of $f(t)$. It only tests the chronological order of the variables in statistics.

A precondition of the Granger causality inference is that the time series must be stable, otherwise, there will be pseudo regression. Hence, the stationarity of each time series should be confirmed by the unit root test before the test. To test whether the variable $u(t)$ is a Granger cause of sequence $f(t)$, the original hypothesis “ H_0 : $f(t)$ is not the Granger cause of $u(t)$ changing” is put forward. The Granger causality inference model is established by estimating the following two regression models:

$$f(t) = \alpha_0 + \sum_{i=1}^p \alpha_i f(t-i) + \sum_{i=1}^q \beta_i u(t-i) + \varepsilon(t) \quad (1)$$

$$f(t) = \alpha_0 + \sum_{i=1}^p \alpha_i f(t-i) + \varepsilon(t) \quad (2)$$

where Equation (1) is an unconstrained regressive model of $f(t)$ and $u(t)$, noted as U . For Equation (2), it is an autoregressive model of $f(t)$, and is a constrained regression model, noted as R . α_0 is the constant term. α_i, β_i are the ratios of $f(t-i)$ and $u(t-i)$, they denote the contribution to $f(t)$. p and q represent the maximum time lag of variables $f(t)$ and $u(t)$, respectively, and $\varepsilon(t)$ is white noise. For p and q , the appropriate values can be determined by the Bayesian information criterion (BIC) or Akaike information criterion (AIC) [39].

The magnitude of Granger causality can be estimated by the logarithm of the corresponding F-statistic [40]. Then, the F-statistics can be constructed by the sum of residual squared RSS_R and RSS_U of the two regression models:

$$F = \frac{(RSS_R - RSS_U)/q}{RSS_U/(n-p-q-1)} \sim F(q, n-p-q-1) \quad (3)$$

where n is the sample size and RSS is calculated as follows:

$$RSS = \sum_{t=1}^n (f(t) - \hat{f}(t))^2 \quad (4)$$

Finally, the probability ρ of the original hypothesis can be obtained by looking up the table of F-distribution. If $\rho \leq \rho_{\max}, \beta_1, \beta_2, \dots, \beta_q$ significantly do not equal 0, so the original hypothesis should be rejected. In other words, $u(t)$ is the Granger cause of $f(t)$ changing. Otherwise, the original hypothesis should be accepted.

2.2. Extreme Gradient Boosting

Extreme gradient boosting (XGBoost) is an integrated promotion algorithm developed by Chen and Guestrin [41]. It evaluates the influence of different features by constructing the regression problem with gradient boosting. Unlike traditional gradient boosting, XGBoost does not add residuals to construct a stump every time, rather introduces a slightly larger tree with leaves and normalization to avoid high variance and overfitting. Therefore, the XGBoost algorithm can be regarded as an additive model consisting of multiple decision trees, expressed as Formula (5):

$$\hat{f}(t) = \sum_{k=1}^K RT_k(u(t)), RT_k \in G \quad (5)$$

Assuming that the dataset has n samples and m features $\mathfrak{S} = \{(u(t), f(t))\} (t = 1, 2, \dots, n)$, RT stands for a regression tree, the notation K is the number of trees, and $\hat{f}(t)$ is the regression result. $u(t)$ represents the input factor, and G is the space that contains the function of all decision trees.

$$G = \left\{ RT(u) = \omega_{s(u)} \right\} \quad (6)$$

where s denotes the structure of each tree which maps a sample to the corresponding leaf index and ω represents the leaf weights, namely the score of corresponding leaves.

$f_l(t)$ is defined as the regression result of t -th instance at l -th iteration, in order to train the tree structure, the objective function is minimized as Equation (7):

$$J_l = \sum_{t=1}^n L(f_l(t), \hat{f}_{l-1}(t) + RT_l(u(t))) + \Omega(f_l) \quad (7)$$

$$\Omega(f_l) = \gamma \cdot N_l + \frac{1}{2} \lambda \sum_{j=1}^{N_l} \omega_j^2 \quad (8)$$

where L denotes the loss function, N is the number of leaf nodes, and γ and λ are penalty factors. The second term $\Omega(f_l)$ represents the complexity of the tree model to avoid overfitting. Under the regularized objective function, complex models will be penalized, and the model with simple predictive functions will be selected as the best model.

Since the objective is difficult to deal with in Euclidean space by conventional methods, second-order Taylor expansion is employed to optimize the above problem [42]. Then, Equation (7) can be simplified as:

$$J_l = \sum_{t=1}^n \left[L\left(f_l(t), \hat{f}_{l-1}(t) + g_t RT_l(u(t)) + \frac{1}{2} g_t' RT_l^2(u(t))\right) \right] + \Omega(f_l) \quad (9)$$

$$g_t = \frac{\partial L(f_l(t), \hat{f}_{l-1}(t))}{\partial \hat{f}_{l-1}(t)} \quad (10)$$

$$g_t' = \frac{\partial^2 L(f_l(t), \hat{f}_{l-1}(t))}{\partial \hat{f}_{l-1}^2(t)} \quad (11)$$

where g_t and g_t' are the first and second-order gradients of loss functions, respectively.

The constant terms can be removed, and the objective is simplified as follow approximate formulation:

$$J_l = \sum_{t=1}^n \left[g_t \omega_{s(u(t))} + \frac{1}{2} g_t' \omega_{s(u(t))}^2 \right] + \gamma N_l + \frac{1}{2} \lambda \sum_{j=1}^{N_l} \omega_j^2 \quad (12)$$

For a fixed structure $s(u(t))$, the optimal weight ω_j^* of leaf j is obtained.

$$\omega_j^* = -\frac{\sum_{t \in I_j} g_t}{\sum_{t \in I_j} g_t' + \lambda} \quad (13)$$

The corresponding optimal solution is calculated by

$$J_l = -\frac{1}{2} \sum_{j=1}^T \frac{\left(\sum_{t \in I_j} g_t\right)^2}{\sum_{t \in I_j} h_t + \lambda} + \gamma N_l \quad (14)$$

where $I_j = \{t \mid s(u(t)) = j\}$ is the instance set of leaf j in the tree structure. The final value J_l is used to evaluate the quality of the tree. The smaller the value, the better the structure.

2.3. Convolutional Neural Network

The convolutional neural network (CNN) is a kind of feedforward neural network, which has a powerful ability on extracting features and has good performance in image vision, natural language processing, and so on [43]. The basic architecture of the CNN is mainly composed of two parts, i.e., the convolutional layer and pooling layer, as shown

in Figure 1. In essence, the CNN pursues constructing multiple filters to extract useful potential information through a layer-by-layer convolution and pooling of input data.

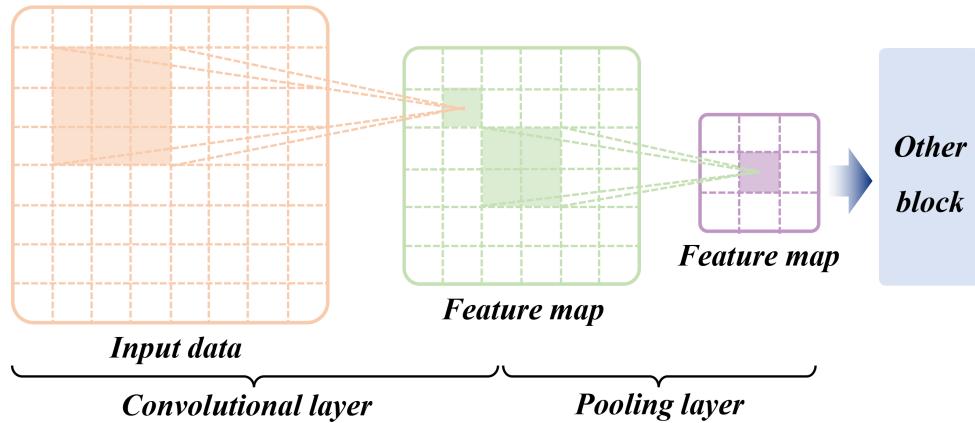


Figure 1. The basic architecture of a typical CNN.

In the convolutional layer, it contains a plurality of convolution kernels, which can be considered tiny windows. Feature maps of the previous layer are convolved with a convolution kernel and the output feature is generated by an activation function. The generated new features are usually more useful than the original features of the input data, which can promote the performance of the model. The operation of the convolutional layer can be described as follows:

$$m_j^l = a \left(\sum_{i \in M_j} m_i^{l-1} * k_{ij}^l + b_j^l \right) \quad (15)$$

where m_j^l represents the j th output feature map of the l th layer, M_j is the selection of input maps, k_{ij}^l denotes the weights between the i th input map and the j th output map, $*$ is the convolution operation, and b_j^l is the bias of the convolution kernel. $a(\cdot)$ represents the activation function such as rectified linear unit (ReLU), and it enables the nonlinear expression of the feature maps to enhance the feature expression capacity.

After the convolution operation, the features of the original data have been extracted, but the dimension of the extracted features is very high, and the application cost is very high in practice. In order to solve this problem, a pooling layer is usually added behind the convolution layer to reduce the dimension of the extracted features, so as to accelerate the convergence of the network. The pooling layer is a sub-sampling technique to extract certain values from convolution features and generate low-dimensional matrices. The pooling layer adopts a process similar to that of the convolution layer, using a small sliding window to take the convoluted features as the input and output a new value. Therefore, the output of the pooling layer can be regarded as a condensed version of the convolution layer's features. There are three pooling operations: maximum, minimum, and average pooling. The operation of the pooling layer can be formulated as Equation (16):

$$m_j^l = a \left(\zeta_j^l mp(m_i^{l-1}) + b_j^l \right) \quad (16)$$

where $mp(\cdot)$ represents the max pooling sub-sampling function and ζ_j^l is the bias. The pooling operation can ensure that the CNN can obtain a relatively robust feature representation, because small changes in input data will not change the output value of the pooling layer.

2.4. Long Short-Term Memory (LSTM) Network

The recurrent neural network (RNN) is a special kind of neural network; it can circulate the state in its own network and learn a lot of historical information, so it is very suitable for processing time series data [44]. However, with the growth of time, the RNN will not be able to complete the connection of information. LSTM is an improved recurrent neural network that uses cells to store long-term memory and introduces the gating mechanism to control cell states [45]. It avoids the problem that RNN cannot deal with long-distance information dependence, and can also solve the common problems of gradient explosion and gradient disappearance in neural networks. LSTM has been widely used in many fields, such as natural language processing, autonomous driving, weather forecasting, etc.

The infrastructure of LSTM is illustrated in Figure 2. The aforementioned gating mechanism includes an input gate, forget gate, and output gate. LSTM uses the historical data x to predict the output sequence $y = (y_1, y_2, \dots, y_d)$, where d is the prediction period. The maintenance and update of information follow several steps below. First, the input gate determines how much new information can be stored in the cell state, while calculating the candidate value \hat{C}_t that may be added to the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (17)$$

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (18)$$

Next, the forget gate determines how much information should be forgotten.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (19)$$

The cell state of this block C_t is calculated by discarding partial information of the previous cell state C_{t-1} and adding the cell state candidate of this block \hat{C}_t .

$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t \quad (20)$$

Finally, the output gate decides how much information in the cell state can be passed to the next memory block, and the final output results are as follows:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (21)$$

$$h_t = o_t \odot \tanh(C_t) \quad (22)$$

$$y_t = \Phi(W_y h_t + b_y) \quad (23)$$

where h_t is the hidden layer state. i_t , f_t , o_t are the state of the input gate, forget gate, and output gate, respectively. W_i , W_f , W_c , W_o , and W_y represent the appropriate weight matrices, b_i , b_f , b_c , b_o , and b_y denote the corresponding bias vectors. Moreover, $\sigma()$ and $\tanh()$ are sigmoid and hyperbolic tangent function, respectively, \odot is the element-wise product of the vectors and $\Phi()$ is the network output activation function.

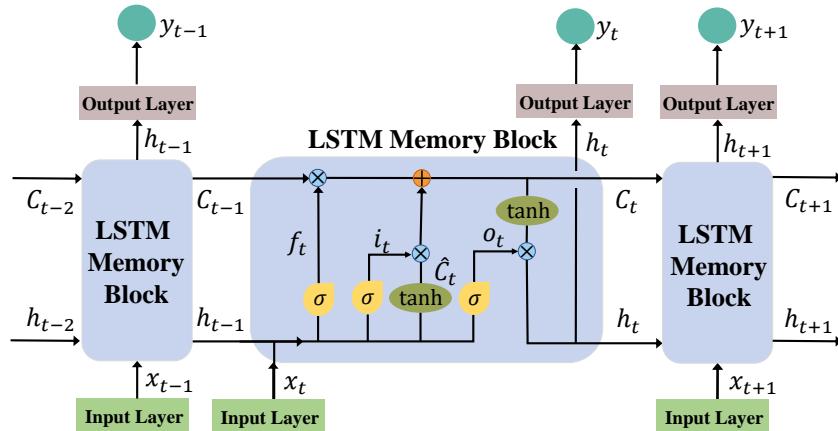


Figure 2. The structure of LSTM memory block.

3. The Proposed Method

This paper establishes a hybrid commodity price forecasting method based on multi-factor selection and fusion with the CNN-LSTM network (MFSFCL), which includes factors selection and price forecasting. Firstly, collect the factors that may affect the change in commodity price, use Granger causality inference to screen out the factors that have a causal relationship with the commodity price, and select the most important factors from the candidate factors by XGBoost. Then, the CNN is used to fuse the selected factors and extract the implicit features, and LSTM is employed to model the sequence to obtain the predicted value of the future price. The schematic diagram of the proposed MFSFCL method is illustrated in Figure 3.

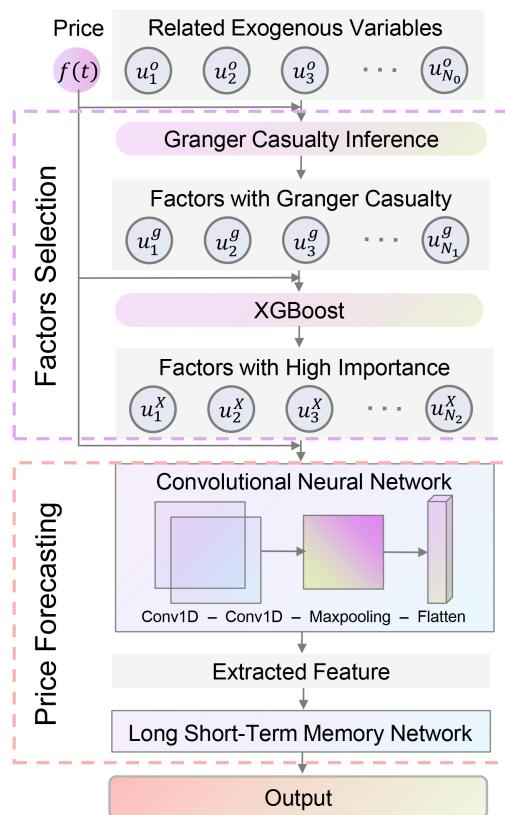


Figure 3. The schematic diagram of the proposed MFSFCL method.

3.1. Factor Selection

In detail, assuming that $f(t), t = 1, 2, \dots, n$ is the original price sequence, and n is the number of samples. $u_i^0(t), i = 1, 2, \dots, N_0$ are the sequences of collected factors, N_0 is the number of collected factors. For commodity price, the related factors perhaps include the economic situation, international currency exchange rate, the changes in mainstream exchanges, and so on.

First of all, not all of the collected factors are related to the change in commodity price, so we need to find out the factors that have an impact on price. The Granger causality inference is conducted between the collected exogenous factors and the target price sequence. If the probability ρ is less than ρ_{\max} , it indicates that the factor is one of the Granger causes of the price sequence $f(t)$. That is, adding this factor together to predict $f(t)$, the performance is better than using only $f(t)$. N_1 factors are screened out through the hypothesis test, defined as $u_j^g(t), j = 1, 2, \dots, N_1$.

Then, even though the above operation performed a preliminary screening of exogenous factors, there are still many remaining factors. Although the selected factors $u_j^g(t)$ are related to the price series $f(t)$ to a certain extent, different factors have different degrees of influence; that is, different degrees of importance. If all factors are employed to predict price in general, it will bring a great burden to the predictor. At the same time, factors with different influence degrees are mixed together and treated equally, and factors with low influence degrees may directly obscure those with high influence degrees. This overly complex and redundant information may not greatly improve the prediction effect or even have a negative effect. Hence, as the saying goes, “take the essence and discard the dross”, it is necessary to analyze and pick out the important factors.

XGBoost is a classic algorithm for feature engineering, which can automatically analyze the importance of each feature. Therefore, XGBoost is adopted to pick out the factors with high importance from $u_j^g(t)$. To assess the importance of each factor, information gain is adopted as Equation (24):

$$imp = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad (24)$$

where $G_L = \sum_{t \in I_L} |g_t|$, $G_R = \sum_{t \in I_R} |g_t|$, $H_L = \sum_{t \in I_L} g'_t$ and $H_R = \sum_{t \in I_R} g'_t$. I_L and I_R are the instances sets of left and right nodes.

After ranking the importance of factors imp , N_2 factors $u_k^X(t), k = 1, 2, \dots, N_2$ are finally obtained according to the change of importance, where N_2 is the number of factors ultimately selected. These factors also are the core exogenous variables that mainly affect the change in price.

3.2. Price Forecasting

After determining the factors affecting the changes in price, the next task is to make the prediction. Usually, there will be several factors selected by XGBoost, which still seems not so friendly for a single model to complete both feature extraction and prediction. In order to alleviate this problem, the CNN is employed to conduct factors fusion and feature extraction from a total of $N_2 + 1$ time series of the selected factors and price to obtain a higher-level feature representation (HLF) before establishing the predictor.

$$HLF = CNN(u_1^X(t), u_2^X(t), \dots, u_{N_2}^X(t), f(t)) \quad (25)$$

Of course, it is necessary to reconstruct the time series data before using the CNN to adapt to the input structure of the CNN [46]. The specific operations will not be repeated due to space limitations.

Finally, a prediction model is established based on the extracted features of the HLF . Due to the existence of memory cells, the LSTM network is very good at extracting sequence characteristics and has a good performance in time series prediction. Therefore, this paper

adopts the LSTM network as a predictor to establish a multi-input long-term prediction model to forecast the future price. In detail, the complete procedure of the proposed framework is conducted and given in Algorithm 1.

Algorithm 1: Price forecasting based on multi-factor selection and fusion with the CNN-LSTM network;

Input: The historical zinc price $f(t)$, the collected exogenous variables $u_i^o(t), i = 1, 2, \dots, N_0$;

Output: The forecasted future price $y(t)$;

- 1 **#Factors Selection**
- 2 **for** $i = 1, \dots, N_0$ **do**
- 3 Establish regression models according to Equations (1) and (2) for factor $u_i^o(t)$ and price series $f(t)$;
- 4 On the basis of Equations (3) and (4), the F-statistic is calculated according to the sum resident squared of the regression models;
- 5 Obtain the corresponding probability by looking up the table of F-distribution;
- 6 **if** $\rho < \rho_{\max}$ **then**
- 7 $u_i^o(t)$ is a causal factor and is selected;
- 8 **end**
- 9 **end**
- 10 The selected factors $u_j^g(t), j = 1, 2, \dots, N_1$ and $f(t)$ are combined to establish a decision tree according to XGBoost;
- 11 Calculate the importance of each factor by Equation (24);
- 12 Sort imp in descending order, and select the most critical factors $u_k^X(t), k = 1, 2, \dots, N_2$ based on the inflection point of imp change;
- 13 **#Price Forecasting**
- 14 Utilize the CNN to fuse selected factors $u_k^X(t)$ and price series $f(t)$, and extract hidden features $HLF = CNN(u_1^X(t), u_2^X(t), \dots, u_{N_2}^X(t), f(t))$;
- 15 Take the extracted features as input and establish a long-term forecasting model based on the LSTM network to obtain the future price $y(t)$.

4. Case Study

4.1. Experiment Settings

4.1.1. Data Description

This paper uses the classic commodity zinc as an example to verify the excellent performance of the proposed method. The price of 0# zinc in SMM is collected and set as the target. At the same time, 18 possible factors are collected which may affect the change in the zinc price, including the economic situation, international currency exchange rate, and stock, such as the S&P 500 Index, the price of LME zinc, and so on. For the convenience of expression, relevant abbreviations are shown in Table 1. The price series of SMM 0# zinc is illustrated in Figure 4. Due to the influence of many factors, the price fluctuates frequently and changes widely. Specifically, the time span of zinc price is from 3 January 2017 to 2 December 2020, excluding public holidays, with a total of 953 daily observations. The data from 3 January 2017 to 24 February 2020 (763 observations) is used for model training, meanwhile, the remaining data (190 observations) serves as a testing dataset to verify the ability of the forecasting model.

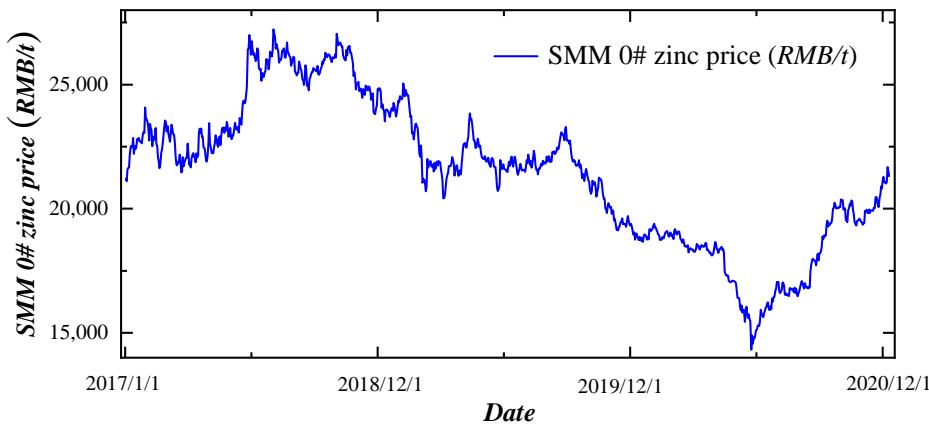


Figure 4. The 0# zinc price of Shanghai Metal Market from 3 January 2017 to 2 December 2020.

Table 1. Abbreviations of collected factors.

Number	Factor	Abbreviation	Number	Factor	Abbreviation
1	Closing price of S&P500	CPSP	10	Cash price of LME zinc	CAPLME
2	Opening price of S&P500	OPSP	11	Settlement price of LME zinc	SPLME
3	High price of S&P500	HPSP	12	Asian Stock of LME zinc	ASLME
4	Low price of S&P500	LPSP	13	Closing price of US dollar index	CPUS
5	Closing price of LME zinc	CPLME	14	Opening price of US dollar index	OPUS
6	Opening price of LME zinc	OPLME	15	High price of US dollar index	HPUS
7	High price of LME zinc	HPLME	16	Low price of US dollar index	LPUS
8	Low price of LME zinc	LPLME	17	Zinc index	ZI
9	Average price of LME zinc for three months	APLME	18	Nonferrous metals index fund	NMIF

4.1.2. Performance Evaluation Criteria

In order to evaluate the performance of the proposed MFSFCL more comprehensively, this paper uses some criteria from two different dimensions, numerical prediction accuracy, and direction prediction accuracy. For numerical prediction accuracy, several commonly used evaluation indexes are adopted, such as mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE). The smaller the value of these indexes, the smaller the prediction error. For price series, it is also very important to judge the direction of future change, so the direction prediction accuracy D_{stat} should also be considered. For D_{stat} , a higher value represents a more accurate prediction direction. The specific calculation process of the above indicators can be seen in [47].

4.1.3. Parameters Settings

For a more comprehensive analysis of market changes, long-term forecasts are usually more appropriate. Here, the historical price and the selected exogenous factors of the past 7 days are utilized to predict zinc price in the next 3 days, rather than just a single-step forecast. For Granger causality inference, the maximum lag is naturally set to 7 through the analysis of AIC and BIC, and the significance level ρ_{max} is set to 0.01. For XGBoost, according to experience, the maximum tree depth is set to 3, the learning rate is 1, and the number of decision trees is 100. For the CNN, it contains two convolution layers—a pooling layer and a ‘flatten’ layer. The number of convolution kernels is 256 and 512, respectively, and the time domain length of the convolution kernel is 3. The activation function is ReLU. Max pooling is used, followed by a flatten layer to facilitate connection with the LSTM predictor. For the LSTM network, the step size of the input layer and output layer are 7 and 3, respectively. The number of neurons in the two hidden layers is set to (512, 64) by trial and error. A reasonable number of neurons is helpful to learn the complex changes of

the sequence, while not making the model too complicated. As a rule of thumb, ReLU is used as the activation function, the learning rate is 0.01 and the batch size is 16.

4.2. Factor Selection

To remove the invalid information in the originally collected exogenous variables, the Granger causality inference is employed. Each variable carries out regression modeling with the zinc price, and the hypothesis test is used to judge whether this variable is helpful for price prediction. The concrete results are depicted in Table 2.

Table 2. The results of the Granger causality inference.

Factor	ρ	Factor	ρ
CSPS	0.0000	CAPLME	0.0000
OPSP	0.2378	SPLME	0.0000
HPSP	0.0022	ASLME	0.4320
LPSP	0.0004	CPUS	0.0042
CPLME	0.0000	OPUS	0.0319
OPLME	0.3726	HPUS	0.0070
HPLME	0.0000	LPUS	0.0043
LPLME	0.0000	ZI	0.7402
APLME	0.0000	NMIF	0.0344

In detail, if the hypothesis test probability of a factor is less than $\rho_{\max} = 0.01$, it will be considered to have a causal relationship with the change in the zinc price. In Table 2, there are 12 out of the original 18 factors that are causal to zinc price. They are CSPS, HPSP, LPSP, APLME, CAPLME, SPLME, CPUS, HPUS, LPUS, CPLME, HPLME, and LPLME. Generally speaking, S&P 500 index, the zinc price of the London Metal Exchange and the US dollar index, directly affect the change of the zinc price in the Shanghai Metal Market. In other words, the zinc price is mainly affected by international currency exchange rates and economic conditions. This is roughly consistent with our cognition. Generally, when the economic situation is good, the price of basic raw materials rises, and vice versa. The international price is marked and settled in US dollars. When the US dollar depreciates, the price of metals rises, and vice versa.

After removing the non-causal factors, it can be seen that there are still many factors. In reality, a few influencing factors often play a key role, and redundant information may reduce the prediction effect to a certain extent [48]. Therefore, XGBoost is utilized to evaluate the importance of the remaining factors to select the core elements. Set the zinc price series $f(t)$ as the regression target, and the information gain is adopted to measure the importance of each factor. The larger the information gain, the higher the importance of the factor. Figure 5 depicts the rank of factor importance.

It is clear that the importance of HPLME, CPLME, CAPLME, and APLME is much higher than other factors. Therefore, these four factors are selected as the key factors affecting the change in the zinc price in the Shanghai Metals Market. This does not mean that other factors are not helpful to the SMM zinc price forecasting, but that the selected four factors have a more direct and important influence. In general, the selected factors have a common characteristic. They are the zinc price of the London Metal Exchange, which means that LME has a great impact on SMM. It is gratifying that this conclusion is consistent with Yue's work. Yue et al. used the VAR-DCC-GARCH model to explore the relationship between Chinese nonferrous metals prices and the nonferrous metals prices from LME. The results showed that LME nonferrous metals prices have a great impact on Chinese nonferrous metals prices, and the co-movement of nonferrous metal prices between LME and Chinese markets presents hysteretic nature [49]. This finding directly reflects the effectiveness of the proposed factor selection method.

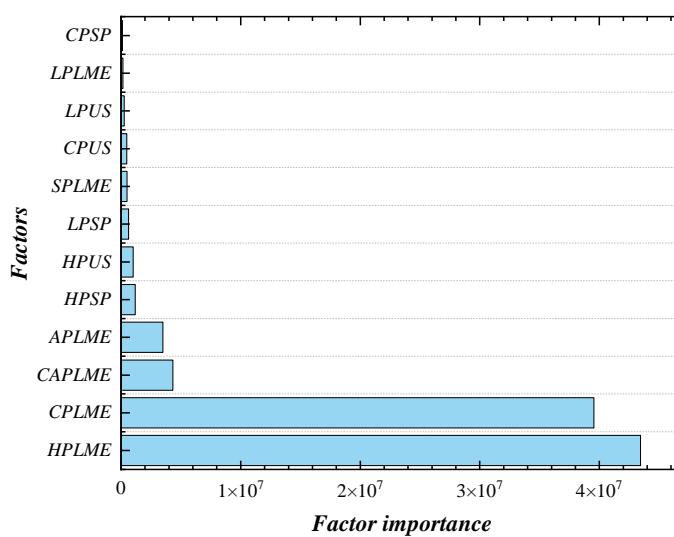


Figure 5. The rank of factor importance.

4.3. Comparative Analysis of Price Forecasting

4.3.1. Compared with Univariate Forecasting Method

In order to explore the performance of the proposed multi-factor forecasting method, MFSFCL is compared with several advanced univariate prediction methods such as ARIMA, multiple-output support vector regression (MSVR), ELM, feedforward neural network (FNN), and LSTM. The parameter (p, d, q) of ARIMA can be determined by AIC and BIC. For MSVR, the kernel function is linear, and the epsilon is set to 1. For the ELM network, the (neuron) number of hidden layers is 110. As for FNN (I-H-H-O), the number of hidden nodes is (64, 16), and the learning rate and batch size are set to 0.01 and 16. For LSTM, the parameter settings are the same as FNN.

The qualitative analysis results are shown in Figure 6. From the perspective of the numerical accuracy of the prediction, the proposed MFSFCL has the smallest MAE, RMSE, and MAPE, which means that it has the smallest prediction error. Compared with other univariate forecasting methods, the prediction error of MFSFCL is greatly reduced, which has obvious advantages. Interestingly, it can be found that except for the poor performance of ELM, the remaining univariate forecasting methods have similar numerical forecasting accuracy at different time steps. This is because the zinc price is affected by many factors. Although these univariate forecasting methods have good prediction capabilities, the single information also determines the upper limit of forecasting performance. The forecasting performance of ARIMA, MSVR, FNN, and LSTM reaches the upper limit for univariate forecasting methods. ELM often usually does not perform well in dealing with such complex prediction tasks due to their stochastic nature. In terms of directional accuracy, MFSFCL has obvious advantages in one-step forecasting and two-step forecasting compared with other univariate methods and also has competitive accuracy in three-step forecasting. In general, MFSFCL has the best overall performance on D_{stat} . At the same time, it can be found that the prediction effect gradually deteriorates with the passage of time, which is an inevitable phenomenon of all methods. After all, the future is full of uncertainty. The more uncertain it is in the future, the more difficult it is to predict.

From a quantitative point of view, MFSFCL significantly reduces the prediction errors compared to other univariate prediction methods, as shown in Table 3. The MAE, RMSE, and MAPE performances of the proposed MFSFCL are 58.66%, 54.22%, and 74.71% lower than those of the best performance of other methods, respectively, for the first day. Moreover, there is a prediction error decrease of 65.77%, 60.72%, and 79.51% for the second day and 70.35%, 65.79%, and 82.39% for the third day. Due to the characteristics of strong noise and violent fluctuation in financial time series, the univariate prediction method only considers one-sided information, which is difficult to obtain satisfactory results. Nevertheless, the

proposed MFSFCL reasonably takes into account the influence of exogenous variables on the change in the zinc price, and has strong predictive performance.

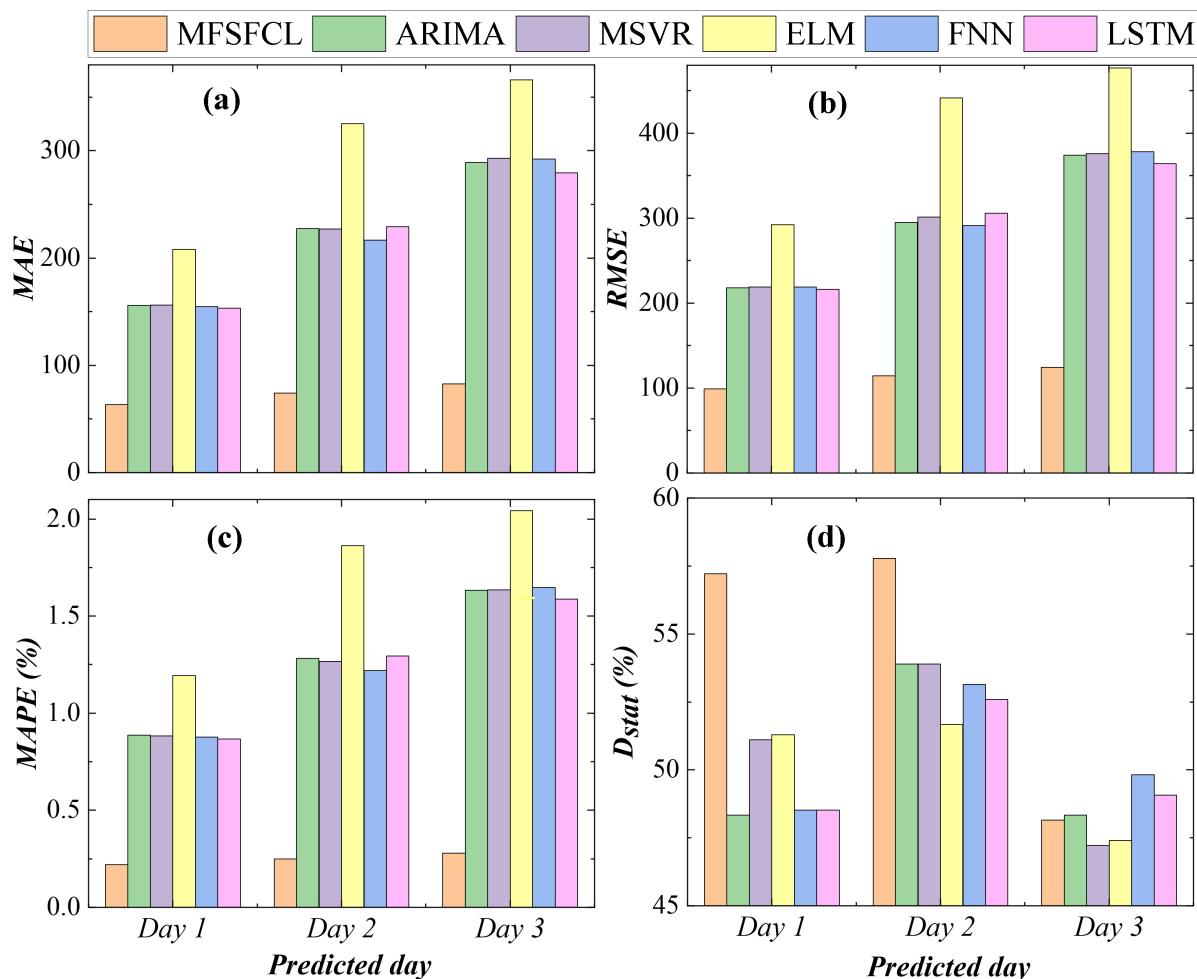


Figure 6. The comparison of different forecasting models for different days. (a–d) are the results for MAE, RMSE, MAPE, and D_{stat} , respectively.

Table 3. Forecasting performance evaluation metrics of the proposed MFSFCL and other univariate methods.

	MFSFCL	ARIMA	MSVR	ELM	FNN	LSTM
Day1	MAE	63.33	155.94	156.32	208.22	154.85
	RMSE	99.05	218.02	218.97	292.32	219.00
	MAPE (%)	0.22	0.89	0.88	1.19	0.88
	D_{stat}	57.22	48.33	51.11	51.30	48.52
Day2	MAE	74.13	227.54	226.96	325.18	216.59
	RMSE	114.37	295.18	301.08	441.41	291.20
	MAPE (%)	0.25	1.28	1.27	1.86	1.22
	D_{stat}	57.78	53.89	53.89	51.67	51.15
Day3	MAE	82.82	288.95	293.11	365.87	292.42
	RMSE	124.61	373.98	375.99	476.86	378.38
	MAPE (%)	0.28	1.63	1.64	2.04	1.65
	D_{stat}	48.15	48.33	47.22	47.41	49.81

Figure 7 explores the distribution of relative prediction errors for different methods on different days. In order to show the specific changes more clearly, a portion of the view is presented separately. In general, compared with the univariate prediction method, the relative forecasting error value of the proposed MFSFCL is the smallest, and basically fluctuates around 0, which means that MFSFCL has accurate forecasting accuracy and robust performance. This directly reflects that the rational use of exogenous variables can improve the forecasting effect.

Moreover, it also can be found that the performances of deep learning methods, such as FNN and LSTM, seem to be better than those of statistical prediction methods and machine learning methods. This is because the deep neural network has a strong modeling ability and can extract and analyze the variation rules of complex sequences. However, other methods may be difficult to deal with the time series data that fluctuates violently and varies dramatically. It is worth noting that LSTM performs the best among all univariate forecasting methods, which reflects the superiority of the LSTM network for time series processing. It builds the structure of the information cycle through memory cells, which can effectively associate the historical information with the current input, capture the dynamic change characteristics of the sequence, and obtain a better prediction effect.

4.3.2. Compared with the Case with Other Factors

To further clarify the effectiveness of the factors selected by the proposed MFSFCL, we randomly select 4 factors from the original 18 factors to predict the future zinc price. Due to the large number of combinations, 20 sets of non-repetitive factors are randomly selected. For each set of factors, they are sent to the CNN and LSTM for prediction, and the average of 10 interdependent experiments is used for analysis. Finally, the 3 sets with the best results among the 20 sets are selected for comparative analysis. The experimental results are shown in Table 4.

Table 4. Forecasting performance evaluation matrix under different factor combinations.

	The Selected Factors	OPSP-CAPLME-HPUS-SPLME	LPSP-NMIF-LPLME-SPLME	CPSP-ZI-ASLME-OPLME
Day1	MAE	63.33	88.53	99.84
	RMSE	99.05	117.05	126.11
	MAPE (%)	0.22	0.29	0.32
	D _{stat}	57.22	49.1	48.33
Day2	MAE	74.13	110.52	114.64
	RMSE	114.37	153.23	145.44
	MAPE (%)	0.25	0.37	0.37
	D _{stat}	57.78	56.94	53.78
Day3	MAE	82.82	135.18	124.68
	RMSE	124.61	189.14	159.55
	MAPE (%)	0.28	0.45	0.40
	D _{stat}	48.15	49.17	50.55

For the factors selected by the proposed method based on Granger causality inference and XGBoost, the prediction effect is significantly better than other factor combinations. Further, it can be found that the three best-performing combinations all contain several exogenous variables screened by Granger causality inference. This means that it is really helpful to use the screened factors for prediction. In the proposed MFSFCL, Granger causality inference determines whether the factor can promote the prediction effect of price series from the perspective of econometrics, and XGBoost analyzes and sorts the importance of each factor to the prediction from the perspective of machine learning. It is logical to

use valid, critical information to achieve great predictions. For the randomly selected factor combination, the addition of irrelevant factors brings redundant and inappropriate information, which affects the judgment of zinc price to a certain extent. The above results demonstrate the importance and validity of factor selection.

In addition, the multi-factor prediction results in Table 4 are all superior to the tested univariate prediction methods. This indicates that exogenous factors provide more reliable information for prediction and significantly improve the prediction accuracy.

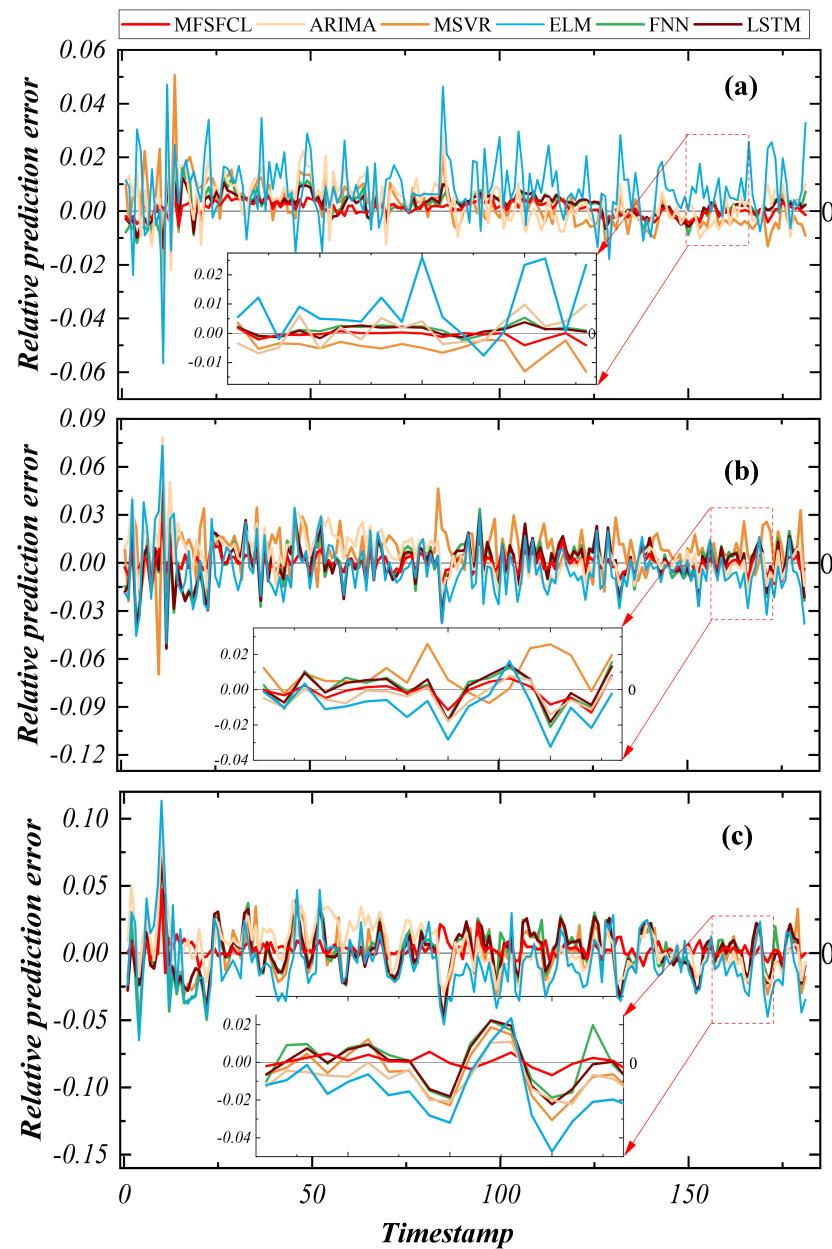


Figure 7. The relative prediction errors of different forecasting models for different days; (a–c) depict the relative prediction errors of day 1, day 2, and day 3, respectively.

4.3.3. Ablation Experiment

Finally, to verify the necessity and effectiveness of each module, an ablation experiment is performed on the proposed MFSFCL. The relevant results are shown in Table 5. For the convenience of demonstration, the combination of each module's abbreviations represents different scenarios under the ablation experiment. For instance, GCI-CNN-LSTM represents

the combination of Granger causality inference, convolutional neural network, and long short-term memory network.

Table 5. Forecasting performance evaluation matrix under different scenarios in the ablation experiment.

		MFSFCL	GCI-XGBoost-LSTM	GCI-CNN-LSTM	CNN-LSTM	LSTM
Day1	MAE	63.33	138.54	170.31	203.45	224.34
	RMSE	99.05	171.28	206.38	238.67	264.19
	MAPE (%)	0.22	0.45	0.57	0.66	0.71
	D_{stat}	57.22	47.47	48.54	46.54	46.98
Day2	MAE	74.13	160.25	194.05	210.67	239.32
	RMSE	114.37	196.92	233.30	249.45	282.50
	MAPE (%)	0.25	0.53	0.65	0.69	0.77
	D_{stat}	57.78	54.44	54.03	54.26	54.26
Day3	MAE	82.82	178.53	213.86	203.96	271.08
	RMSE	124.61	222.04	256.62	246.71	325.43
	MAPE (%)	0.28	0.58	0.71	0.67	0.88
	D_{stat}	48.35	48.15	50.22	50.06	48.46

Unsurprisingly, MFSFCL has the best performance in all scenarios of the ablation experiment. Compared with GCI-XGBoost-LSTM, the proposed method uses the CNN to fuse factors and price effectively, and learns the internal representation of time series data to extract higher-level features. Advanced knowledge representation can reduce the burden of the prediction model and improve the performance. For GCI-CNN-LSTM, it does not conduct further screening of Granger causality factors, but directly uses a large number of factors for prediction. MFSFCL uses XGBoost to pick out the factors with high importance and screen out the key components that have a great impact on price, avoiding the complex network structure in the predictor. Therefore, it has significant advantages in all kinds of prediction performance. For the CNN-LSTM, it does not perform any factor processing, and its forecasting performance is much worse than the previous methods. This is because some of the factors originally collected only have weak effects or even have no correlation with the change in the zinc price. The complex and redundant information not only brings great interference to the analysis of future changes for zinc price but also creates a complex network structure. With the model, it is difficult to obtain good forecasting performance in this case. For the single LSTM network, it is clear that various complex exogenous variables do not improve the forecasting effects and bring about very heavy burdens. Therefore, the overall performance of the LSTM network is the worst. Furthermore, the prediction performance in this case is far inferior to that of univariate LSTM, which directly reflects the importance of factor selection and processing.

Specifically, for the LSTM and CNN-LSTM, although their performances are not so ideal, it is obvious that the introduction of the CNN can extract hidden information from high-dimensional input and obtain high-level representation. This directly reduces the burden of the predictor LSTM, thus improving the prediction effect. Therefore, the CNN plays a key role in dealing with complex high-dimensional sequences. Compared with CNN-LSTM, GCI-CNN-LSTM adopts Granger causality inference to conduct preliminary processing on the originally collected exogenous variables and removes some invalid information, which directly improves the overall forecasting accuracy. Of course, on this basis, MFSFCL uses XGBoost to screen out the core components of the influencing factors, which greatly reduces the complexity of the prediction model and improves the overall forecasting performance.

In general, the proposed MFSFCL effectively makes use of the advantages of each module, takes into account the influence of external factors on zinc price, and conducts screening, fusion, and modeling for each factor. In general, it has a robust and accurate prediction performance.

5. Conclusions

The change in the commodity price is the key basis of market transactions and economic management, so it is necessary to make an accurate prediction of the dynamic commodity price. However, due to the influences of various hidden factors, the commodity price usually varies frequently and fluctuates violently with obvious nonlinear characteristics. Therefore, it is difficult to obtain accurate and robust prediction results only using the historical data of the price itself. To this end, this paper proposes a hybrid multi-factor price forecasting method based on factor selection and fusion with the CNN-LSTM network. Firstly, the Granger causality inference is used to remove the non-causal factors in the collected exogenous variables. Then, in order to screen out the factors which have a significant impact on the commodity price, XGBoost is adopted to evaluate and sort the importance of the remaining factors. Next, a CNN is employed to fuse the selected factors together and extract hidden features. Finally, a multi-input long-term prediction model is established by using the LSTM network to obtain the future price. The quantitative and qualitative results of comparative experiments indicate that the performance of the proposed MFSFCL outperforms other state-of-the-art methods. The analysis conclusion on the factors affecting the price of SMM also provides strong support for zinc price forecasting. It is a promising multi-factor forecasting method and can be widely used in other fields. Considering that different exogenous variables have different influences on prices, how to effectively use this characteristic to obtain more accurate prediction results is a direction that will be worthy of more research in the future.

Author Contributions: Conceptualization, Y.L.; methodology, Y.L. and C.Y.; software, W.L.; validation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, C.Y. and K.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (grant nos. 62073340, 61860206014), in part by the National Key R&D Program of China (2019YFB1705300), in part by the Shandong Key Laboratory of Industrial Control Technology (Qingdao University), in part by Fundamental Research Funds from the Central Universities of Central South University (2021zzts0199), and in part by the Science and Technology Innovation Program of Hunan Province (grant nos. 2021RC3018 and 2021RC4054).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bakas, D.; Triantafyllou, A. Commodity price volatility and the economic uncertainty of pandemics. *Econ. Lett.* **2020**, *193*, 109283. [[CrossRef](#)]
2. Liu, Y.; Yang, C.; Huang, K.; Gui, W.; Hu, S. A Systematic Procurement Supply Chain Optimization Technique Based on Industrial Internet of Thing and Application. *IEEE Internet Things J.* **2022**, *in press*. [[CrossRef](#)]
3. Ivanova, M.; Dospatliev, L. Effects of Diesel Price on Changes in Agricultural Commodity Prices in Bulgaria. *Mathematics* **2023**, *11*, 559. [[CrossRef](#)]
4. Giannerini, S.; Goracci, G. Entropy-Based Tests for Complex Dependence in Economic and Financial Time Series with the R Package tseriesEntropy. *Mathematics* **2023**, *11*, 757. [[CrossRef](#)]
5. Huang, K.; Wang, Z.; Jusup, M. Incorporating Latent Constraints to Enhance Inference of Network Structure. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 466–475. [[CrossRef](#)]
6. Fister, D.; Perc, M.; Jagrič, T. Two robust long short-term memory frameworks for trading stocks. *Appl. Intell.* **2021**, *51*, 7177–7195. [[CrossRef](#)] [[PubMed](#)]
7. Liu, W.; Wang, C.; Li, Y.; Liu, Y.; Huang, K. Ensemble forecasting for product futures prices using variational mode decomposition and artificial neural networks. *Chaos Solitons Fractals* **2021**, *146*, 110822. [[CrossRef](#)]

8. Rodríguez, A.; Melgarejo, M. Identification of Colombian coffee price dynamics. *Chaos Interdiscip. J. Nonlinear Sci.* **2020**, *30*, 013145. [[CrossRef](#)]
9. Yuan, Y.; Zhang, T. Forecasting stock market in high and low volatility periods: a modified multifractal volatility approach. *Chaos Solitons Fractals* **2020**, *140*, 110252. [[CrossRef](#)]
10. Wang, Y.; Wang, H. Using networks and partial differential equations to forecast bitcoin price movement. *Chaos Interdiscip. J. Nonlinear Sci.* **2020**, *30*, 073127. [[CrossRef](#)]
11. Frezza, M. A fractal-based approach for modeling stock price variations. *Chaos Interdiscip. J. Nonlinear Sci.* **2018**, *28*, 091102. [[CrossRef](#)] [[PubMed](#)]
12. Drachal, K. Forecasting prices of selected metals with Bayesian data-rich models. *Resour. Policy* **2019**, *64*, 101528. [[CrossRef](#)]
13. Zhu, X.H.; Zhang, H.W.; Zhong, M.R. Volatility forecasting in Chinese nonferrous metals futures market. *Trans. Nonferrous Met. Soc. China* **2017**, *27*, 1206–1214. [[CrossRef](#)]
14. Kriebelbaumer, T.; Angus, A.; Parsons, D.; Casado, M.R. An improved wavelet–ARIMA approach for forecasting metal prices. *Resour. Policy* **2014**, *39*, 32–41. [[CrossRef](#)]
15. Şahinli, M.A. Potato price forecasting with Holt-Winters and ARIMA methods: A case study. *Am. J. Potato Res.* **2020**, *97*, 336–346. [[CrossRef](#)]
16. Dehghani, H.; Bogdanovic, D. Copper price estimation using bat algorithm. *Resour. Policy* **2018**, *55*, 55–61. [[CrossRef](#)]
17. Wang, Z.; Li, Z.; Wang, R.; Nie, F.; Li, X. Large graph clustering with simultaneous spectral embedding and discretization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 4426–4440. [[CrossRef](#)]
18. Huang, K.; Tao, Z.; Liu, Y.; Sun, B.; Yang, C.; Gui, W.; Hu, S. Adaptive multimode process monitoring based on mode-matching and similarity-preserving dictionary learning. *IEEE Trans. Cybern.* **2022**, *in press*. [[CrossRef](#)]
19. Astudillo, G.; Carrasco, R.; Fernández-Campusano, C.; Chacón, M. Copper Price Prediction Using Support Vector Regression Technique. *Appl. Sci.* **2020**, *10*, 6648. [[CrossRef](#)]
20. García, D.; Kristjanpoller, W. An adaptive forecasting approach for copper price volatility through hybrid and non-hybrid models. *Appl. Soft Comput.* **2019**, *74*, 466–478. [[CrossRef](#)]
21. Alameer, Z.; Abd Elaziz, M.; Ewees, A.A.; Ye, H.; Jianhua, Z. Forecasting copper prices using hybrid adaptive neuro-fuzzy inference system and genetic algorithms. *Nat. Resour. Res.* **2019**, *28*, 1385–1401. [[CrossRef](#)]
22. Liu, C.; Wang, K.; Wang, Y.; Yuan, X. Learning deep multimanifold structure feature representation for quality prediction with an industrial application. *IEEE Trans. Ind. Inform.* **2021**, *18*, 5849–5858. [[CrossRef](#)]
23. Wang, S.H.; Nayak, D.R.; Guttery, D.S.; Zhang, X.; Zhang, Y.D. COVID-19 classification by CCSHNet with deep fusion using transfer learning and discriminant correlation analysis. *Inf. Fusion* **2021**, *68*, 131–148. [[CrossRef](#)] [[PubMed](#)]
24. Chen, Y.; Xie, X.; Zhang, T.; Bai, J.; Hou, M. A deep residual compensation extreme learning machine and applications. *J. Forecast.* **2020**, *39*, 986–999. [[CrossRef](#)]
25. Atsalakis, G.S.; Atsalaki, I.G.; Pasiouras, F.; Zopounidis, C. Bitcoin price forecasting with neuro-fuzzy techniques. *Eur. J. Oper. Res.* **2019**, *276*, 770–780. [[CrossRef](#)]
26. Kamdem, J.S.; Essomba, R.B.; Berinyuy, J.N. Deep learning models for forecasting and analyzing the implications of COVID-19 spread on some commodities markets volatilities. *Chaos Solitons Fractals* **2020**, *140*, 110215. [[CrossRef](#)]
27. Wang, J.; Li, X. A combined neural network model for commodity price forecasting with SSA. *Soft Comput.* **2018**, *22*, 5323–5333. [[CrossRef](#)]
28. Ugurlu, U.; Oksuz, I.; Tas, O. Electricity price forecasting using recurrent neural networks. *Energies* **2018**, *11*, 1255. [[CrossRef](#)]
29. Kristjanpoller, W.; Hernández, E. Volatility of main metals forecasted by a hybrid ANN-GARCH model with regressors. *Expert Syst. Appl.* **2017**, *84*, 290–300. [[CrossRef](#)]
30. Lazcano, A.; Herrera, P.J.; Monge, M. A Combined Model Based on Recurrent Neural Networks and Graph Convolutional Networks for Financial Time Series Forecasting. *Mathematics* **2023**, *11*, 224. [[CrossRef](#)]
31. Hu, Y.; Ni, J.; Wen, L. A hybrid deep learning approach by integrating LSTM-ANN networks with GARCH model for copper price volatility prediction. *Phys. A Stat. Mech. Its Appl.* **2020**, *557*, 124907. [[CrossRef](#)]
32. Livieris, I.E.; Pintelas, E.; Pintelas, P. A CNN–LSTM model for gold price time-series forecasting. *Neural Comput. Appl.* **2020**, *32*, 17351–17360. [[CrossRef](#)]
33. Risso, M. Combining wavelet decomposition with machine learning to forecast gold returns. *Int. J. Forecast.* **2019**, *35*, 601–615. [[CrossRef](#)]
34. Zhang, C.; Zhao, Y.; Zhao, H. A Novel Hybrid Price Prediction Model for Multimodal Carbon Emission Trading Market Based on CEEMDAN Algorithm and Window-Based XGBoost Approach. *Mathematics* **2022**, *10*, 4072. [[CrossRef](#)]
35. Zhang, F.; Liao, Z. Gold price forecasting based on RBF neural network and hybrid fuzzy clustering algorithm. In *Proceedings the Seventh International Conference on Management Science and Engineering Management*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 73–84.
36. Vakitbilir, N.; Hilal, A.; Direkoglu, C. Hybrid deep learning models for multivariate forecasting of global horizontal irradiation. *Neural Comput. Appl.* **2022**, *34*, 8005–8026. [[CrossRef](#)]
37. Zhang, Y.D.; Dong, Z.; Wang, S.H.; Yu, X.; Yao, X.; Zhou, Q.; Hu, H.; Li, M.; Jiménez-Mesa, C.; Ramirez, J.; et al. Advances in multimodal data fusion in neuroimaging: overview, challenges, and novel orientation. *Inf. Fusion* **2020**, *64*, 149–187. [[CrossRef](#)] [[PubMed](#)]

38. Bekun, F.V.; Alhassan, A.; Ozturk, I.; Gimba, O.J. Explosivity and Time-Varying Granger Causality: Evidence from the Bubble Contagion Effect of COVID-19-Induced Uncertainty on Manufacturing Job Postings in the United States. *Mathematics* **2022**, *10*, 4780. [[CrossRef](#)]
39. Gustavo, P.; O. Durão, F.; Bernardo, P.A.; Silva, M.C.E. Neural network approach based on a bilevel optimization for the prediction of underground blast-induced ground vibration amplitudes. *Neural Comput. Appl.* **2020**, *32*, 5975–5987.
40. Geweke, J. Measurement of linear dependence and feedback between multiple time series. *J. Am. Stat. Assoc.* **1982**, *77*, 304–313. [[CrossRef](#)]
41. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
42. Zhou, Y.; Li, T.; Shi, J.; Qian, Z. A CEEMDAN and XGBOOST-based approach to forecast crude oil prices. *Complexity* **2019**, *2019*, 4392785. [[CrossRef](#)]
43. Gao, R.; Xu, J.; Chen, Y.; Cho, K. Heterogeneous Feature Fusion Module Based on CNN and Transformer for Multiview Stereo Reconstruction. *Mathematics* **2023**, *11*, 112. [[CrossRef](#)]
44. Mikolov, T.; Karafiat, M.; Burget, L.; Cernocky, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the Interspeech, Makuhari, Chiba, Japan, 26–30 September 2010; Volume 2, pp. 1045–1048.
45. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [[CrossRef](#)]
46. Huang, K.; Wu, S.; Li, F.; Yang, C.; Gui, W. Fault Diagnosis of Hydraulic Systems Based on Deep Learning Model With Multirate Data Samples. *IEEE Trans. Neural Networks Learn. Syst.* **2021**, *33*, 6789–6801. [[CrossRef](#)]
47. Liu, Y.; Yang, C.; Huang, K.; Gui, W. Non-ferrous metals price forecasting based on variational mode decomposition and LSTM network. *Knowl.-Based Syst.* **2020**, *188*, 105006. [[CrossRef](#)]
48. Wang, S.; Celebi, M.E.; Zhang, Y.D.; Yu, X.; Lu, S.; Yao, X.; Zhou, Q.; Miguel, M.G.; Tian, Y.; Gorri, J.M.; et al. Advances in data preprocessing for biomedical data fusion: An overview of the methods, challenges, and prospects. *Inf. Fusion* **2021**, *76*, 376–421. [[CrossRef](#)]
49. Yue, Y.D.; Liu, D.C.; Shan, X. Price linkage between Chinese and international nonferrous metals commodity markets based on VAR-DCC-GARCH models. *Trans. Nonferrous Met. Soc. China* **2015**, *25*, 1020–1026. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.