mathematics

Article

A Novel Variant of LSTM Stock Prediction Method Incorporating Attention Mechanism

Shuai Sang and Lu Li *

School of Mathematics, Physics and Statistics, Shanghai University of Engineering Science, Shanghai 201620, China; sangshuai2022@163.com
* Correspondence: lilu@sues.edu.cn

Abstract: Long Short-Term Memory (LSTM) is an effective method for stock price prediction. However, due to the nonlinear and highly random nature of stock price fluctuations over time, LSTM exhibits poor stability and is prone to overfitting, resulting in low prediction accuracy. To address this issue, this paper proposes a novel variant of LSTM that couples the forget gate and input gate in the LSTM structure, and adds a "simple" forget gate to the long-term cell state. In order to enhance the generalization ability and robustness of the variant LSTM, the paper introduces an attention mechanism and combines it with the variant LSTM, presenting the Attention Mechanism Variant LSTM (AMV-LSTM) model along with the corresponding backpropagation algorithm. The parameters in AMV-LSTM are updated using the Adam gradient descent method. Experimental results demonstrate that the variant LSTM alleviates the instability and overfitting issues of LSTM, effectively improving prediction accuracy. AMV-LSTM further enhances accuracy compared to the variant LSTM, and compared to AM-LSTM, it exhibits superior generalization ability, accuracy, and convergence capability.

Keywords: LSTM; attention mechanism; stock price prediction

MSC: 37N99

Citation: Sang, S.; Li, L. A Novel Variant of LSTM Stock Prediction Method Incorporating Attention Mechanism. *Mathematics* **2024**, *12*, 945. https://doi.org/10.3390/math12070945

Academic Editors: Heui Seok Lim, Sanghyuk Lee, Yeongwook Yang and Imatitikua Aiyanyo

Received: 28 February 2024
Revised: 20 March 2024
Accepted: 21 March 2024
Published: 22 March 2024

1. Introduction

Stock price prediction has always been a significant research topic in the field of finance. Researchers have proposed various algorithms to forecast the trends of stock prices, which are highly volatile, complex, and nonlinear [1–4]. Traditional forecasting methods include Multivariable Linear Regression, Exponential Smoothing, Autoregressive Integrated Moving Average (ARIMA), etc. However, these models are often relatively simple and may not achieve satisfactory prediction accuracy. In recent years, with the development of big data and artificial intelligence, machine learning and deep learning methods have been widely applied to stock price prediction research [5–7]. Examples include Random Forest, Extreme Gradient Boosting (XGBoost), and Support Vector Machine (SVM), among others. Compared to traditional methods, these approaches have effectively improved prediction accuracy.

Deep learning is a machine learning concept based on artificial neural networks. For many applications, deep learning models outperform shallow machine learning models and traditional data analysis methods [8]. There are various forms of neural networks, such as BP neural networks, RNN recurrent neural networks, etc. Long Short-Term Memory (LSTM), as an important research method in deep learning, has achieved significant success in the field of financial forecasting and has become one of the cutting-edge research methods. LSTM is an improvement on the structure of Recurrent Neural Networks (RNN), addressing the issues of RNN gradient vanishing, gradient exploding, and insufficient long-term memory by utilizing gate mechanisms when learning long-term dependencies [9]. It can more effectively utilize information previously present in time series and current input data to make more accurate predictions about the future.

Applying LSTM for prediction on S&P 500 component stocks from 1992 to 2015, it was found that the prediction performance was superior to Random Forest (RAF), Deep Neural Networks (DNN), and Logistic Regression Classifier (LOG) [10]. While LSTM can produce good predictions in some cases, it may still struggle with highly nonlinear time series [11]. One possible reason is that the time dependencies between elements in these time series are not well exploited [12]. This point is evidenced by [12], where the use of LSTM and other deep learning models for runoff prediction showed that the model's predictive accuracy can be demonstrated to some extent through temporal dependencies. In 2000, Felix Gers and his colleagues found that if the internal state of LSTM is not updated when processing input time series, the network would eventually collapse [13]. Therefore, they introduced the forget gate mechanism to update and reset its state [14]. The introduction of the forget gate mechanism somewhat effectively handles the time dependencies between elements, but the forget gate in the LSTM unit acts on the input data x_t and the hidden state h_t, without considering c_t. Additionally, experiments by [15] also demonstrated that the forget gate is the only crucial gate in LSTM. Meanwhile, Felix Gers and others [13] introduced peephole connections, giving rise to peephole LSTM. However, it was pointed out in [14] that, in most cases, peephole LSTM does not perform well compared to LSTM. Yet, experiments indicate that the input gate and output gate mechanisms are crucial gating mechanisms for finding better-performing LSTM units. The coupling of the forget gate and input gate has a comparable performance with LSTM. Therefore, considering that the forget gate is a key gating mechanism in LSTM, improving the input and output gates may lead to better results. This paper redesigns the internal structure of LSTM units and introduces a variant of the LSTM structure. Compared to existing LSTM and peephole LSTM, the variant LSTM captures the idea that the forget gate is the key gate, retaining the original forget gate while coupling the input gate with the forget gate for improvement. This enhances the information utilization efficiency of the variant LSTM unit relative to the LSTM unit. Additionally, although peephole LSTM performs poorly in many experiments, its idea is still insightful. Therefore, considering the long-term state c_t, but handling it through a "simple" forget gate. As a result, compared to peephole LSTM, the variant LSTM focuses structural improvements on the input gate and forget gate, making the structure simpler. In the backpropagation process, the number of parameters that need to be learned is also reduced, reducing the computational time of the model. Later, some scholars further modified LSTM. In [16], LSTM was integrated with CNN and an attention mechanism was introduced, discovering the feasibility of the model and improving its accuracy.

Additionally, when there is significant data noise, the predictive accuracy of LSTM for stock prices is greatly affected. Inspired by the human visual system, attention mechanisms in machine learning have been developed for a long time [17] and are now widely applied in various deep learning tasks, such as natural language processing, speech recognition, image recognition, and the processing of time series data. It is an information allocation mechanism that simulates the human brain's attention. The attention mechanism can assign different weights to the input data of the model, highlighting the importance of useful data and weakening the importance of less relevant data, reducing the impact of irrelevant parts. It was initially proposed by Bahdanau et al. [18] and applied to machine translation, giving a probability distribution of attention when seeking attention distribution. They later proposed the soft attention mechanism and hard attention mechanism [19]. And others constructed an LSTM with a soft attention mechanism and applied it to the prediction of flow in the Canadian basin, finding an improvement in accuracy; ref. [20] and others applied an LSTM with a soft attention mechanism to the prediction of photovoltaic generation, finding the new model to be more effective and robust compared to a multi-layer perceptron (MLP) and traditional LSTM models. We also chose the soft attention mechanism because it is a deterministic mechanism and fully differentiable. The hard attention mechanism is an uncertain mechanism; its process is a random process, and during decoding, the system does not use all hidden states but randomly selects some [21]. In this way, when computing gradients, it can effectively integrate with the algorithm to calculate directly.

In response to the issue of low prediction accuracy and poor model robustness in current stock price forecasting methods, considering the high volatility of stock data, the AMV-LSTM stock price prediction model is proposed. This paper improves upon the traditional LSTM neural network variant by seamlessly integrating an attention mechanism. The attention mechanism layer is placed in front of the variant LSTM, allowing data to pass through weight layers and normalization layers to obtain scores. These scores are then dot-producted with the original data vectors to assign weights, thereby forming new data vectors. Finally, these new data vectors are used as inputs to the variant LSTM, serving the purpose of noise reduction and forming a new neural network model. The model emphasizes the importance of different features in stock data, not only learning the temporal dependencies of input sequences but also effectively utilizing the interdependence features among data sequences. This will help improve the accuracy and stability of stock price prediction.

Based on the above description, the contributions of this paper are summarized as follows:

1. To better utilize the temporal dependencies between elements in time series, this paper improves the internal structure of LSTM. By coupling the forget gate with the input gate, simplifying LSTM, and enhancing its efficiency. Also, a simple forget gate is applied to forget the long-term state Ct, relieving the pressure on the LSTM to transmit long-term state information by filtering new inputs and historical state elements.
2. To enhance the model's robustness, an attention mechanism is integrated into the improved LSTM, proposing a variant LSTM model with integrated attention mechanism (AMV-LSTM). This improves the accuracy of stock price prediction by mitigating the impact of significant data noise. Compared to LSTM with integrated attention mechanism (AM-LSTM), it further improves the model's generalization ability, robustness, and convergence speed.
3. Applying AMV-LSTM to stock price prediction yields higher accuracy, providing valuable references for predicting stock prices in the financial market.

2. Related Work

Since the design idea of AMV-LSTM is based on classical LSTM, peephole LSTM, and attention mechanism, this section will first review the relevant contents of these three aspects.

2.1. Typical LSTM Model

The long short-term memory (LSTM) is a variant of the recurrent neural network (RNN) [22]. For time series data, it is more convenient for extracting various information features due to the addition of storage units capable of holding historical data. The LSTM network structure employs a special gate mechanism, consisting of a forget gate, an input gate, and an output gate in each LSTM neuron. The structure of the neuron is shown in Figure 1.

Where f_t is the remaining amount of data information of the previous LSTM neuron after passing the forgetting gate, i_t is the input amount of data information of the current LSTM neuron, $\tilde{c}_t$ is the candidate state value of the cell, and c_t is the output amount of data information of the LSTM neuron. The calculation formula for each gate of each LSTM cell and c_t and h_t is as follows:

$$\begin{aligned} f_t &= \sigma(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f) \\ i_t &= \sigma(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\ \tilde{c}_t &= tanh(W_{xc} \cdot x_t + W_{hc} \odot h_{t-1} + b_c) \\ o_t &= \sigma(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + b_o) \\ h_t &= o_t \odot tanh(c_t). \end{aligned} \tag{1}$$

where σ is the sigmoid activation function; $\odot$ is the Hadamard product, and x_t is the current input vector; W_{xf}, W_{xi}, W_{xc}, W_{xo} are related to input the corresponding weight vector; h_{t-1} is the state vector of the previous cell unit's hidden layer; W_{hf}, W_{hi}, W_{hc}, W_{ho}, respectively, are about the hidden layer weights of state vector; b_f, b_i,b_c, b_o are respective corresponding doors on the bias vector.

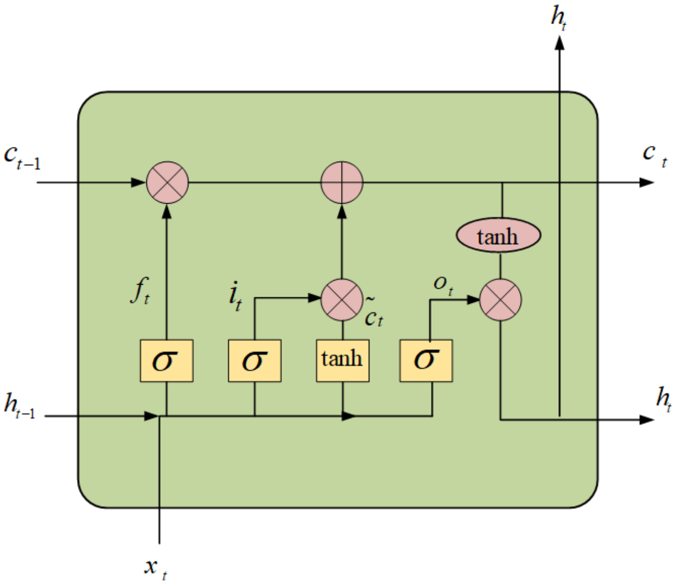


Figure 1. LSTM unit structure.

2.2. Peephole LSTM

In Graves et al.'s study in 2005 [23], they introduced precise time feedback and peephole connection counting. Peephole connections provide direct inputs to the three gates from the cell state to the LSTM, essentially containing information about the importance of past events. In Figure 2, peephole connections at time $t-1$ are represented by blue dashed lines, while solid lines represent peephole connections at time t.

The peephole connecting each LSTM door and c_t and h_t are calculated by the following formula:

$$\begin{aligned} f_t &= \sigma(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + W_{fc} \cdot c_{t-1} + b_f) \\ i_t &= \sigma(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + W_{ic} \cdot c_{t-1} + b_i) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\ \tilde{c}_t &= tanh(W_{xc} \cdot x_t + W_{hc} \odot h_{t-1} + b_c) \\ o_t &= \sigma(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + W_{oc} \cdot c_t + b_o) \\ h_t &= o_t \odot tanh(c_t). \end{aligned} \tag{2}$$

Compared with LSTM, the forward propagation formula of peephole connection LSTM has three more weight matrices: W_{fc}, W_{ic}, W_{oc}.

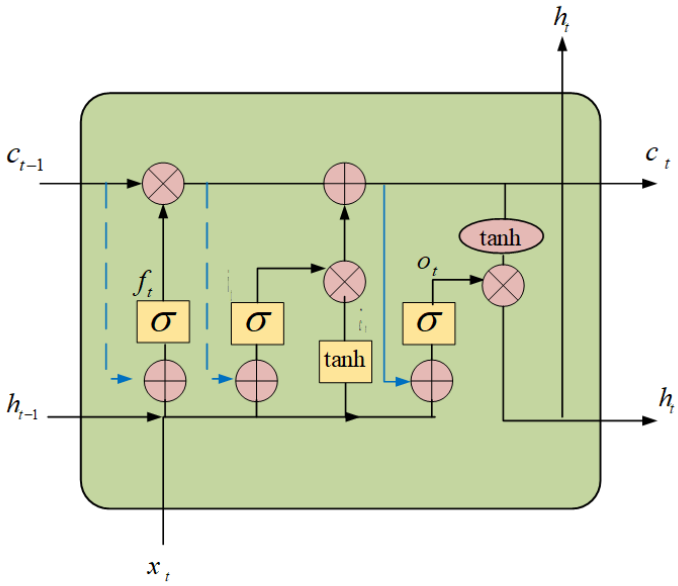


Figure 2. LSTM with peephole connection.

2.3. Attention Mechanism

The attention mechanism involves paying temporal attention to multiple inputs, and its core principle is that the values are weighted by a weighted average, with the weights determined by learnable functions. The basic function of this mechanism is to form a memory by focusing on multiple inputs at different points in time. Its main goal is to extract the characteristics of the input data and help the model comprehensively describe the traffic dynamics in the input data. The key point is that the model has the ability to focus on features based on specific aspects, which helps solve bottlenecks in the model's processing [24].

The Figure 3 illustrates the operational structure of the attention mechanism. Assuming the input data has m features, the dimension of each vector x_t (t = 1, 2 . . . , n) is m, where the time steps are represented by n. Consequently, at each step, n vectors x_t first undergo processing through the attention layer and then each vector x_t is input to each variant of the LSTM unit. Therefore, the initial step involves inputting the input vector x_t into the first attention layer, obtaining the weighted value vector a_t through Formula (3). Since Formula (3) has two learnable weight matrices W_{ax} and V_a, W_{ax} can project the vector x_t into an attention vector space, while V_a can calculate the weighted values. This process effectively provides different focuses on the input vector x_t, thereby reducing the noise in the data. Subsequently, vector a_t is normalized through the second layer to obtain the score of each element in each vector, i.e., vector p_t according to Formula (4). Finally, the score vector p_t is subject to the Hadamard product with the input vector x_t according to Formula (5), resulting in the output $\tilde{x}_t$, which serves as the input vector for the variant of the LSTM.

$$a_t = V_a \cdot tanh(W_{ax} \cdot x_t + b_a), \tag{3}$$

$$p_t = Softmax(a_t), \tag{4}$$

$$\tilde{x}_t = p_t \odot x_t. \tag{5}$$

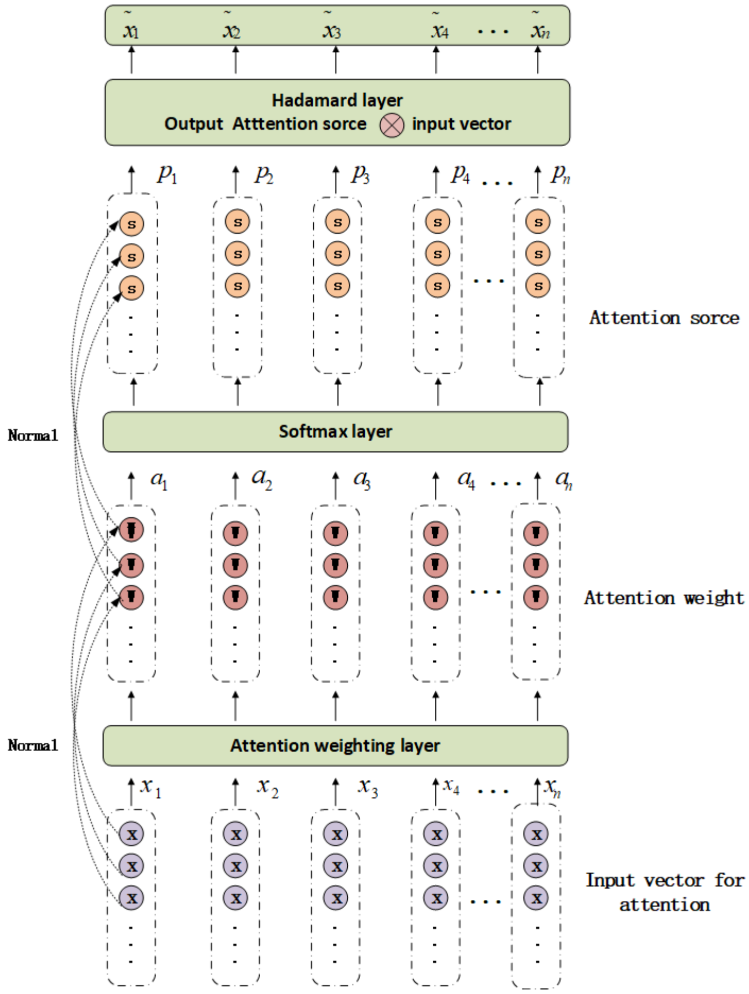

$\tilde{x}_1$ $\tilde{x}_2$ $\tilde{x}_3$ $\tilde{x}_4$ ... $\tilde{x}_n$
Hadamard layer
Output Atttention sorce ⊗ input vector
p_1 p_2 p_3 p_4 ... p_n
Attention sorce
Normal
Softmax layer
a_1 a_2 a_3 a_4 ... a_n
Attention weight
Normal
Attention weighting layer
x_1 x_2 x_3 x_4 ... x_n
Input vector for attention

Figure 3. Attention mechanism layer.

3. AMV-LSTM Model

In this section, the structure and algorithm of the AMV-LSTM model will be fully introduced.

3.1. Variational LSTM

The structure of the variant LSTM unit is shown in Figure 4.

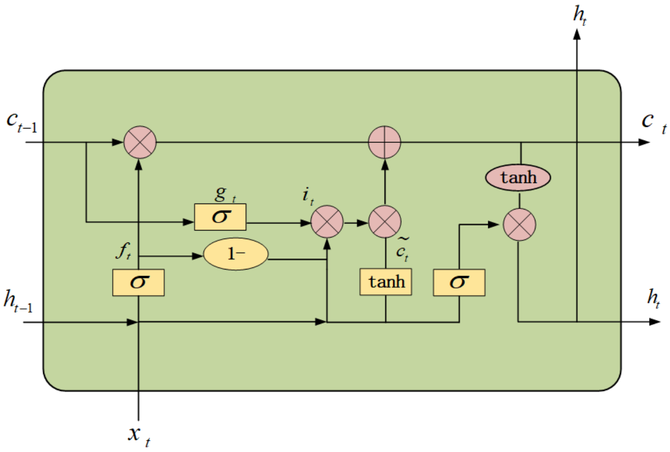


Figure 4. Variational LSTM unit.

The design of the variant LSTM was inspired by the classic LSTM and the peephole LSTM. Since the classic LSTM is relatively sensitive and has poor noise-handling capabilities, the variant LSTM initially couples the forget gate and the input gate on the basis of the classic LSTM model. This coupling establishes a fixed connection between the input gate and the forget gate, enabling the model to no longer make separate decisions when handling old and new information, but to consider both simultaneously. In other words, information is only forgotten when new input is available, and vice versa. This means that input and forget operations occur simultaneously until the information is transmitted to the cell state. When the input gate and the forget gate make independent decisions, the input gate needs to input and transmit all the information without the ability to filter the information. However, the structure after coupling reduces the input information for the input gate, thereby alleviating the processing pressure on the input gate. This reduction is achieved through the filtering by the forget gate and is not a blind random filtration of information. This approach helps to some extent in reducing noisy data and utilizing useful information more effectively, considering the substantial impact of noisy data on stock prices. Furthermore, compared to the LSTM, the coupled structure also reduces the number of parameters, which benefits model training, aids convergence, and ultimately helps improve the accuracy of stock price predictions.

Furthermore, a "simplified" forget gate is established for the long-term cell state c_t, enabling selective forgetting of long-term memory during the information transmission process, thereby enhancing the data noise handling capability. The reason it is referred to as a "simplified" forget gate is because the forget gate here is slightly different from the classic forget gate f_t, as it does not contain a bias term in the *sigmoid* activation function. This results in learning one less parameter when updating the model's parameters and speeds up the convergence rate of the model.

Due to the findings from previous researchers' experiments indicating that the performance of peephole LSTM is often not as effective as that of the classic LSTM, the design of the variant LSTM separates the forget gates for c_t and h_t, not sharing them. However, drawing on the concept of peephole LSTM, the c_t from the "simple" forget gate is input into the input gate. This design of the variant LSTM takes into account the information of c_t during the propagation process, thereby enhancing the efficiency of information transmission in each unit.

3.2. AMV-LSTM Structure

The structure of AMV-LSTM consists of an attention mechanism layer and a variant LSTM. In the model, the attention mechanism layer is located at the front of the variant LSTM, meaning that the input data first passes through the attention mechanism layer. After being processed by the attention mechanism layer, each element of the input data is effectively assigned weights. The processed vector $\tilde{x}_t$ is then input into the variant LSTM layer, entering each variant LSTM unit. Finally, the prediction values are obtained through a fully connected layer.

The structure of AMV-LSTM model is shown in Figure 5.

The combination of the variant LSTM with the attention mechanism is aimed at enhancing the robustness and generalization capability of the variant LSTM. Additionally, compared to AM-LSTM, the structure of AMV-LSTM becomes simpler, primarily due to the design of the variant LSTM. The information transmission is more efficient, and the number of parameters that need to be learned during the backpropagation process is reduced. This effectively addresses the shortcomings of AM-LSTM.

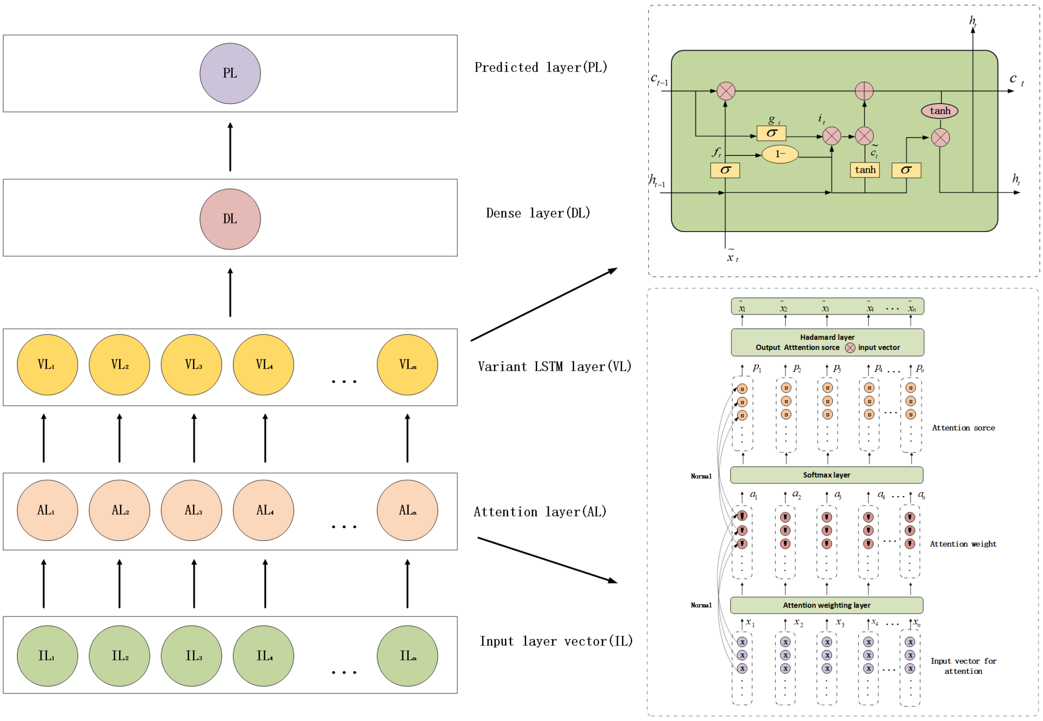


Figure 5. AMV-LSTM structure.

3.3. Forward Propagation Algorithm of AMV-LSTM

So far, the forward propagation formula of the AMV-LSTM model becomes

$$f_t = \sigma(W_{xf} \cdot \tilde{x}_t + W_{hf} \cdot h_{t-1} + b_f) \tag{6}$$

$$i_t = (1 - f_t) \odot g_t \tag{7}$$

$$g_t = \sigma(W_{cg} \odot c_{t-1}) \tag{8}$$

$$\tilde{c}_t = tanh(W_{xc} \cdot \tilde{x}_t + W_{hc} \odot h_{t-1} + b_c) \tag{9}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{10}$$

$$o_t = \sigma(W_{xo} \cdot \tilde{x}_t + W_{ho} \cdot h_{t-1} + b_o) \tag{11}$$

$$h_t = o_t \odot tanh(c_t) \tag{12}$$

$$a_t = V_a \cdot tanh(W_{ax} \cdot x_t + b_a) \tag{13}$$

$$p_t = Softmax(a_t) \tag{14}$$

$$\tilde{x}_t = p_t \odot x_t. \tag{15}$$

Using the Formulas (6)–(15) to pass the calculated h_τ through the fully connected layer, the predicted value is

$$\hat{y}_\tau = W_f \cdot h_\tau + b_y. \tag{16}$$

The loss function L is calculated using the mean square error (RMSE) as follows:

$$L = \frac{1}{2} \cdot (\hat{y}_\tau - y)^2. \tag{17}$$

The total error L_{all} is calculated as

$$L_{all} = \frac{1}{2} \cdot \sum_{i=1}^{n} (\hat{y}_i - y_i)^2. \tag{18}$$

3.4. Backpropagation Algorithm of AMV-LSTM

Backpropagation Through Time (BPTT) is a backpropagation algorithm based on time series, also known as gradient descent [25]. By iterating the parameters to be updated, it minimizes the error and finds the optimal parameter value. In the process of backpropagation to solve the gradient, the loss function L is required to calculate the gradient of each parameter, in which LSTM involves 14 parameters to be updated, including: Weight matrices W_{xf}, W_{xi}, W_{xc}, W_{xo}, W_{hf}, W_{hi}, W_{hc}, W_{ho}, W_y, and offset vectors b_f, b_i, b_c, b_o,b_y. AMV-LSTM needs to update parameters by adding three weight matrices, W_{ax}, W_{cg}, V_a; one parameter vector, b_a; and reducing one weight vector and one parameter vector, W_{xi} and b_i, respectively. AMV-LSTM needs to update a total of 16 parameters.

3.4.1. Variant Gradient Formula for LSTM Layer Parameters

The sensitivity of the loss function to the hidden variable h at the time of t is $\frac{\partial L}{\partial h_t}$, expressed by δ_t. Then, the backpropagation formula of the weight matrix and the parameter vector of the loss function L to the output gate o_t is (19)–(21)

$$\frac{\partial L}{\partial W_{oh,t}} = [\delta_t \odot tanh(c_t) \odot o_t \odot (1 - o_t)]h_{t-1}^T \tag{19}$$

$$\frac{\partial L}{\partial W_{ox,t}} = [\delta_t \odot tanh(c_t) \odot o_t \odot (1 - o_t)]\tilde{x}_t^T \tag{20}$$

$$\frac{\partial L}{\partial b_o} = \delta_t \odot tanh(c_t) \odot o_t \odot (1 - o_t). \tag{21}$$

The backpropagation formula of the loss function L to the weight matrix and the parameter vector of the forgetting gate f_t is (22)–(24).

$$\frac{\partial L}{\partial W_{fh,t}} = [\delta_t \odot o_t \odot (1 - tan^2(c_t)) \odot c_{t-1} \odot f_t \odot (1 - f_t)]h_{t-1}^T \tag{22}$$

$$\frac{\partial L}{\partial W_{fx,t}} = [\delta_t \odot o_t \odot (1 - tan^2(c_t)) \odot c_{t-1} \odot f_t \odot (1 - f_t)]\tilde{x}_t^T \tag{23}$$

$$\frac{\partial L}{\partial b_f} = \delta_t \odot o_t \odot (1 - tan^2(c_t)) \odot c_{t-1} \odot f_t \odot (1 - f_t). \tag{24}$$

The backpropagation formula of the loss function L to the weight matrix of the "simple" forgetting gate g_t is (25)

$$\frac{\partial L}{\partial W_{cg,t}} = [\delta_t \odot o_t \odot (1 - tan^2(c_t)) \odot \tilde{c}_t \odot (1 - f_t) \odot g_t \odot (1 - g_t)](c_{t-1})^T. \tag{25}$$

Loss function L to $\tilde{c}_t$ backpropagation formula of weight matrix and parameter vector for (26)–(28)

$$\frac{\partial L}{\partial W_{ch,t}} = [\delta_t \odot o_t \odot (1 - tan^2(c_t)) \odot i_t \odot (1 - \tilde{c}_t^2)]h_{t-1}^T \tag{26}$$

$$\frac{\partial L}{\partial W_{cx,t}} = [\delta_t \odot o_t \odot (1 - tan^2(c_t)) \odot i_t \odot (1 - \tilde{c}_t^2)]\tilde{x}_t^T \tag{27}$$

$$\frac{\partial L}{\partial b_c} = \delta_t \odot o_t \odot (1 - tan^2(c_t)) \odot i_t \odot (1 - \tilde{c}_t^2). \tag{28}$$

The partial derivative of the loss function L with respect to the weight matrix and bias vector of the output layer is the Formula (29) and (30)

$$\frac{\partial L}{\partial W_y} = (\hat{y}_\tau - y_\tau) \cdot h_\tau^T \tag{29}$$

$$\frac{\partial L}{\partial b_y} = \hat{y}_\tau - y_\tau. \tag{30}$$

3.4.2. Derivation of the Parameter Gradient Formula of the Attention Mechanism Layer

For the three parameters involved in the attention mechanism layer, according to the loss function L and the parameter relationship, the formula for obtaining the partial derivative of W_{ax} is as follows:

$$\begin{aligned}\frac{\partial L}{\partial W_{ax}} = \delta_t \cdot \{&[\xi_t \cdot W_{ox}^T(tanh(c_t) \odot o_t \odot (1 - o_t)) \odot x_t]^T \cdot \vartheta_t \\ &+ [\xi_t \cdot W_{fx}^T(o_t \odot (1 - tanh(c_t)^2) \odot c_{t-1} \odot f_t \odot (1 - f_t)) \odot x_t]^T \cdot \vartheta_t \\ &+ [\xi_t \cdot W_{fx}^T(o_t \odot (1 - tanh(c_t)^2) \odot \tilde{c}_t \odot (-g_t) \odot f_t \odot (1 - f_t)) \odot x_t]^T \cdot \vartheta_t \\ &+ [\xi_t \cdot W_{cx}^T(o_t \odot (1 - tanh(c_t)^2) \odot i_t \odot (1 - \tilde{c}_t^2)) \odot x_t)]^T \cdot \vartheta_t\}.\end{aligned} \tag{31}$$

The formula for finding the partial derivative of the loss function L with respect to V_a is

$$\begin{aligned}\frac{\partial L}{\partial V_a} = &[\xi_t \cdot W_{ox}^T(tanh(c_t) \odot o_t \odot (1 - o_t) \odot \delta_t) \odot x_t] \cdot [tanh(W_{ax} \cdot x_t + b_a)]^T \\ &+ [\xi_t \cdot W_{fx}^T(o_t \odot (1 - tanh(c_t)^2) \odot c_{t-1} \odot f_t \odot (1 - f_t) \odot \delta_t) \odot x_t] \cdot [tanh(W_{ax} \cdot x_t + b_a)]^T \\ &+ [\xi_t \cdot W_{fx}^T(o_t \odot (1 - tanh(c_t)^2) \odot \tilde{c}_t \odot (-g_t) \odot f_t \odot (1 - f_t)) \odot x_t] \cdot [tanh(W_{ax} \cdot x_t + b_a)]^T \\ &+ [\xi_t \cdot W_{cx}^T(o_t \odot (1 - tanh(c_t)^2) \odot i_t \odot (1 - \tilde{c}_t^2)) \odot x_t] \cdot [tanh(W_{ax} \cdot x_t + b_a)]^T.\end{aligned} \tag{32}$$

The formula for finding the partial derivative of the loss function L with respect to b_a is

$$\begin{aligned}\frac{\partial L}{\partial b_a} = &[diag(tanh(c_t)) \cdot diag(o_t \odot (1 - o_t)) \cdot W_{ox} \\ &+ diag(o_t \odot (1 - tanh(c_t)^2)) \cdot diag(c_{t-1}) \cdot diag((f_t \odot (1 - f_t)) \cdot W_{fx}) \\ &+ diag(o_t \odot (1 - tanh(c_t)^2)) \cdot diag(\tilde{c}_t) \cdot diag(-(g_t)) \cdot diag((f_t \odot (1 - f_t)) \cdot W_{fx}) \\ &+ diag(o_t \odot (1 - tanh(c_t)^2)) \cdot diag(i_t) \cdot diag((1 - \tilde{c}_t^2)W_{cx})] \cdot \zeta_t \cdot \delta_t.\end{aligned} \tag{33}$$

In the Formula (33), diag(a) represents the diagonal matrix with the elements of the vector a as diagonal elements, and ξ_t is the partial derivative of p_t with respect to the attention weight a_t obtained from the Softmax layer. The formula is as follows:

$$\begin{aligned}\xi_t = \frac{\partial p_t}{\partial a_t} &= \begin{bmatrix} \frac{\partial p_t^1}{\partial a_t^1} & \frac{\partial p_t^2}{\partial a_t^1} & \cdots & \frac{\partial p_t^{x_{dim}}}{\partial a_t^1} \\ \frac{\partial p_t^1}{\partial a_t^2} & \frac{\partial p_t^2}{\partial a_t^2} & \cdots & \frac{\partial p_t^{x_{dim}}}{\partial a_t^2} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial p_t^1}{\partial a_t^{x_{dim}}} & \frac{\partial p_t^2}{\partial a_t^{x_{dim}}} & \cdots & \frac{\partial p_t^{x_{dim}}}{\partial a_t^{x_{dim}}} \end{bmatrix} \\ &= \begin{bmatrix} p_t^1 * (1 - p_t^1) & -p_t^2 * p_t^1 & \cdots & -p_t^{x_{dim}} * p_t^1 \\ -p_t^1 * p_t^2 & p_t^2 * (1 - p_t^2) & \cdots & -p_t^{x_{dim}} * p_t^2 \\ \vdots & \vdots & \cdots & \vdots \\ -p_t^1 * p_t^{x_{dim}} & -p_t^2 * p_t^{x_{dim}} & \cdots & p_t^{x_{dim}} * (1 - p_t^{x_{dim}}) \end{bmatrix}\end{aligned} \tag{34}$$

The calculation formulas for ϑ_t in Equation (31) and ζ_t in Equation (33) are, respectively,

$$\vartheta_t = V_a \cdot (1 - tanh(W_{ax} \cdot x_t + b_a)^2) \cdot (x_t)^T \tag{35}$$

$$\zeta_t = diag(x_t) \cdot \xi_t \cdot [V_a(diag(1 - tanh(W_{ax} \cdot x_t + b_a)^2))]. \tag{36}$$

In Equation (34), p_t^i denotes the i-th element of vector p_t, a_t^i denotes the i-th element of vector a_t, and x_{dim} refers to the dimension of the input vector x.

At this point, all backpropagation formulas of AMV-LSTM are derived. After the corresponding gradient of each parameter is obtained, the Adam gradient descent method is used to update the parameters.

Taking W_{ax} as an example: Let $(\frac{\partial L}{\partial W_{ax,t}})Adam$ denote the gradient of $\frac{\partial L}{\partial Wax,t}$ after Adam updates. The calculation formula is as follows:

$$q_t = \delta \cdot q_{t-1} + (1 - \delta)\frac{\partial L}{\partial W_{ax,t}} \tag{37}$$

$$r_t = \gamma \cdot r_{t-1} + (1 - \gamma)(\frac{\partial L}{\partial W_{ax,t}})^2 \tag{38}$$

$$\hat{q}_t = \frac{q_t}{1 - \delta^t} \tag{39}$$

$$\hat{r}_t = \frac{r_t}{1 - \gamma^t} \tag{40}$$

$$(\frac{\partial L}{\partial W_{ax,t}})_{Adam} = \frac{\varpi \cdot \hat{q}_t}{\sqrt{\hat{r}_t} + \epsilon}. \tag{41}$$

In Equation (37), $\frac{\partial L}{\partial W_{ax,t}}$ is computed based on Equation (31). The initial values q_0 and r_0 are both set to 0. The parameters ϖ, δ, γ, and ϵ are experimentally set values. Usually, $\delta = 0.9$ and $\gamma = 0.999$ are the exponential decay rates for the matrices, while $\epsilon = 1 \times 10^{-8}$ is a very small number to prevent division by zero. $\varpi = 0.64$ is the learning rate for Adam, which is equal to the global learning rate in this paper [26].

The iterative formula for $W_{ax,t+1}$ at time step $t + 1$ is

$$W_{ax,t+1} = W_{ax,t} - (\eta * 0.9^i) \cdot (\frac{\partial L}{\partial W_{ax,t}})_{Adam}. \tag{42}$$

In Equation (42), i represents the number of iterations, η is a constant, set to 0.64 in this case. By utilizing Equations (37)–(42),the iterative update of the gradient $W_{ax,t}$ can be achieved, and similar methods are applied for updating other parameters.

3.5. Algorithm: AMV-LSTM Model

The algorithmic process of the AMV-LSTM model is as follows:

Step 1: Set the batch size, dimension of the hidden layer, time steps, Adam learning rate for AMV-LSTM, and determine the dimension of the input feature vectors. Partition the experimental data into training and testing samples and preprocess the data. Provide the training samples $X = x_1, x_2, x_3, \ldots, x_\tau$ along with their corresponding labels y.

Step 2: Involves feeding the training samples $X = x_1, x_2, x_3, \ldots, x_\tau$ into the attention mechanism layer, obtaining $\tilde{X} = \tilde{x}1, \tilde{x}2, \tilde{x}3, \ldots, \tilde{x}\tau$ according to Equations (3)–(5).

Step 3: Involves computing $\tilde{y}$ by applying the forward propagation algorithm to $\tilde{X}$ based on Equations (6)–(16), and obtaining Equation (17).

Step 4: Compute the gradients of the loss function with respect to the parameters of the LSTM layer using Equations (19)–(30), and obtain the gradients of the loss function with respect to the parameters of the attention mechanism layer using Equations (31)–(36).

Step 5: Employs the Adam gradient descent method to iteratively update each parameter based on Equations (37)–(42).

Step 6: According to the set number of iterations, determine the termination condition. If the number of iterations is less than or equal to the specified number, return to Step 4; otherwise, terminate the iteration.

Step 7: Train the pre-trained AMV-LSTM model using the training data, make predictions on the test set, and then denormalize the predicted data to obtain the final prediction results.

4. Experiments

In this section, to validate the superiority of AMV-LSTM, a simple comparison is first conducted by comparing the variant LSTM with the classical LSTM model. Subsequently, a comparison is made between AMV-LSTM and LSTM models incorporating attention mechanisms. The precision of AMV-LSTM in predicting the stock price for the next 1 day, the accuracy in predicting the next 2 days, and the convergence speed of AMV-LSTM are studied separately. Ultimately, it is concluded that the performance of AMV-LSTM is more outstanding.

The data used in the experiment are from China Ping An Bank, sourced from the Uminer platform. It is a representative stock in the financial industry. The time span of the dataset is from 19 January 2007, to 12 April 2022, comprising a total of 3590 data points. Fourteen dimensions of stock features were selected, including the opening price, highest price, lowest price, closing price, as well as 5-day, 10-day, 20-day, 60-day, and 120-day exponential moving average indicators, and 5-day, 10-day, 20-day, 60-day, and 120-day simple moving average lines. The time step is set to 10, predicting the stock price for the 11th day based on the stock prices of the preceding 10 days.

During preprocessing, we first checked whether there were missing values or anomalies among the 3590 data points. Since the opening price of stocks is zero during non-trading hours, we removed data points with opening prices of zero to ensure data integrity. Subsequently, considering the practical scenario where the actual opening price of a stock on a given day may not be exactly the same as the previous day's closing price due to factors such as stock dividends, we adjusted the stock data. Two methods, namely forward adjustment and backward adjustment, are commonly used for stock price adjustments. Forward adjustment maintains the stock price while lowering the pre-adjustment closing price below the stock price; backward adjustment increases the adjusted stock price. Both methods effectively correct stock prices. In this study, we applied backward adjustment to correct the stock prices.

The specific steps are as follows:

(1) Determine the ex-dividend date of the stock, which is the date on which the company announces the ex-dividend date.

(2) Calculate the ex-dividend factor for each day prior to the ex-dividend date. The ex-dividend factor is the ratio of the stock price before the ex-dividend date to the price after the ex-dividend date. The calculation formula is

$$\mathrm{EDF} = 1 - \frac{\mathrm{CD}}{\mathrm{BTED}}. \tag{43}$$

In Equation (43), EDF refers to the ex-dividend factor, CD refers to cash dividends, and BTED refers to the stock price on the day before the ex-dividend date.

(3) Use ex-dividend factors to adjust all prices after the ex-dividend date. For each day's stock price, multiply it by the corresponding ex-dividend factor to obtain the adjusted price.

(4) Based on the stock's ex-dividend status, repeat steps 2 and 3 until all ex-dividend factors have been applied to their respective dates.

Afterwards, the dataset was split into a training set and a test set. In this study, 3000 data points were selected as the training set, and 500 data points were chosen from the remaining data as the test set. After the dataset division, due to the impact of measurement units and variance among stock features, it is necessary to normalize the data. The normalization formula is as follows:

$$x_n^* = \frac{x_n^{'} - min(x_n)}{max(x_n) - min(x_n)}. \tag{44}$$

The evaluation criteria for the model include Mean Squared Error (MSE), Mean Absolute Error (MAE), and the Coefficient of Determination (R^2). MSE, MAE, and R^2 are metrics used to gauge the fitting degree and prediction accuracy of the model. For MSE and MAE, smaller values are better, while for the Coefficient of Determination (R^2), a value closer to 1 is preferable. Market volatility affects the model's predictive performance, as reflected in these metrics. When the model exhibits stronger robustness, even in the presence of significant market volatility, it can still capture the nonlinear relationship in stock price data. Consequently, MSE and MAE decrease, while R^2 increases. Comparing models, the more significant the reductions in MSE and MAE, the stronger the model's robustness and its ability to capture nonlinear data relationships, resulting in higher predictive accuracy of stock prices and greater increases in R^2. The formulas for calculating these three evaluation metrics are as follows:

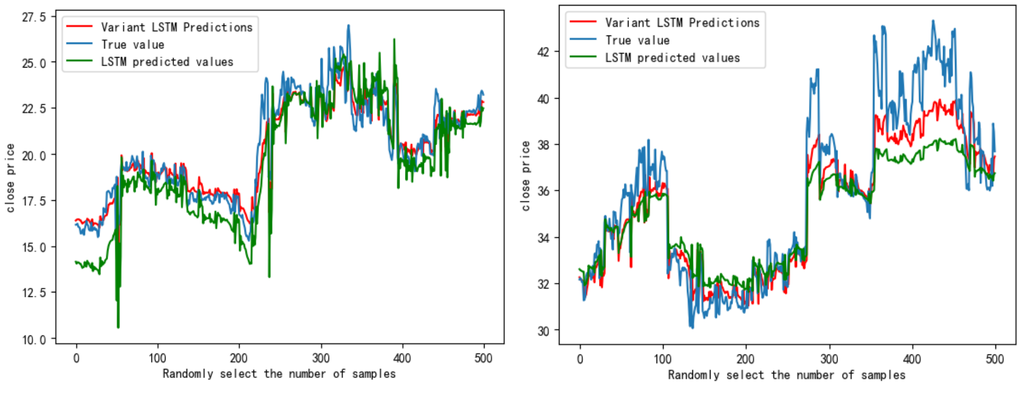
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{45}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{46}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}. \tag{47}$$

4.1. Comparison between Variant LSTM and Classic LSTM

The classic LSTM is highly unstable when predicting stock prices based on 14-dimensional stock features. It is prone to generating ineffective prediction results due to the influence of data noise. Even when capable of handling data, it tends to suffer from overfitting issues. In contrast, modified versions of LSTM demonstrate more stability in predicting stock prices and offer some relief in situations of overfitting (Figure 6).



(**a**) (**b**)

Figure 6. The comparative analysis between Variant LSTM and LSTM in predicting. (**a**) Train set; (**b**) Test set.

In the experiment, Adam's learning rate is 0.64. And the model performs best when the learning rate of Adam is at 0.64, with stable predictive performance observed between 0.6 and 0.75. Additionally, the optimal batch size is 500. The iterations are conducted for a total of 100 rounds each. From the performance chart, it is evident that, compared to the classic LSTM, the variant LSTM yields more accurate predictions on the test set. To confirm that the "simple" forget gate indeed selectively forgets information, a heatmap of the weights of the "simple" forget gate W_{gc} is plotted (Figure 7). This visualization provides a more intuitive understanding of the distribution of numerical values.

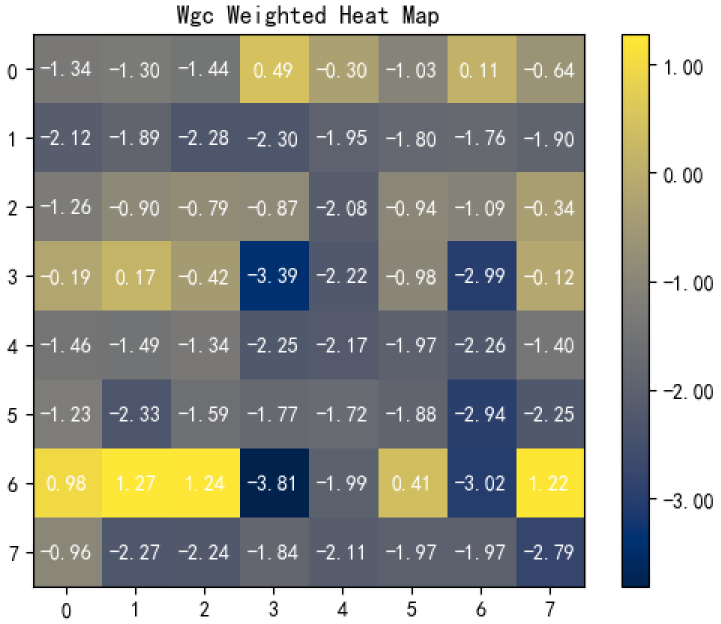


Figure 7. W_{gc} Weighted Heat Map.

From the graph, it is evident that the "simple" gate plays a role in the information within c_t; most of the information can be selectively forgotten, thus alleviating the burden on the LSTM caused by an excess of information during transmission.

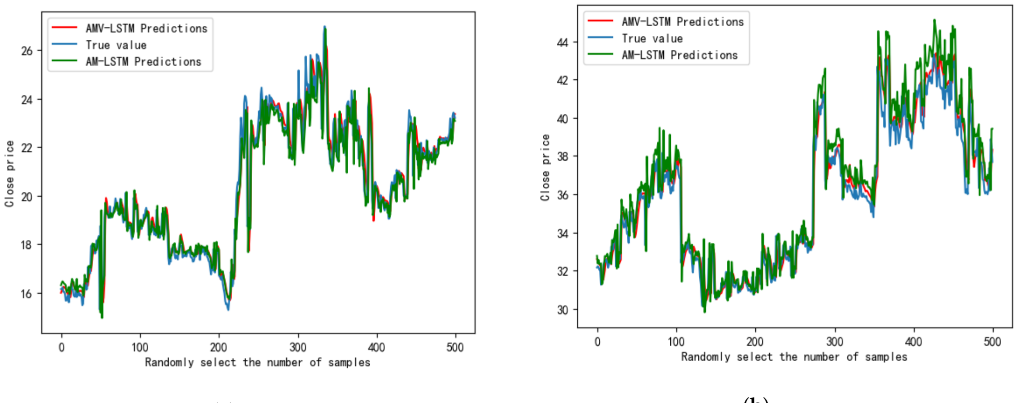
Furthermore, although the classic LSTM performs reasonably well in predicting stable stock prices, it struggles to capture sudden spikes or drops in stock prices in a timely manner. While the variant LSTM shows significant improvement in both stable and unstable stock prices compared to the classic LSTM, its accuracy still does not reach a very high level.

4.2. Comparison Experiment of AMV-LSTM and AM-LSTM

4.2.1. Predicting Future Stock Prices for 1 Day

Due to the variant LSTM's relatively low accuracy in predicting stock prices, an attention mechanism layer is added in front of the variant LSTM. This forms the AMV-LSTM, which first predicts the stock price for the next day.

In comparison with the variant LSTM shown in the predicted performance Figure 8, it is evident that AMV-LSTM has significantly improved accuracy in both the training and test sets.



(**a**) (**b**)

Figure 8. The comparative analysis between AMV-LSTM and AM-LSTM in predicting. (**a**) Train set; (**b**) Test set.

Comparing with Tables 1 and 2, it can be observed that, after 20 rounds of iteration, the accuracy of the AMV-LSTM has increased by 6.7727% on the training set and by 18.8125% on the testing set, compared to the variant LSTM. Additionally, when compared to the AM-LSTM, the performance of AMV-LSTM remains superior, especially on the testing set, where the accuracy has further increased by approximately 3%.

Table 1. Comparison of evaluation indicators.

Evaluation Indicators	MSE (Train)	MSE (Test)	MAE (Train)	MAE (Test)	R^2 $(Train)$	R^2 $(Test)$
LSTM	0.001616	0.002551	0.031581	0.037447	0.776809	0.163586
Variant LSTM	0.000606	0.001328	0.017157	0.026679	0.845297	0.743773

Table 2. Evaluation index data comparison.

Evaluation Indicators	MSE (Train)	MSE (Test)	MAE (Train)	MAE (Test)	R^2 $(Train)$	R^2 $(Test)$
AM-LSTM	0.000429	0.001060	0.012837	0.023940	0.898990	0.904071
AMV-LSTM	0.000412	0.000638	0.013507	0.017364	0.913024	0.931898

From Figures 9 and 10, it can be seen that when facing significant fluctuations in stock prices, AMV-LSTM is more adaptable to such abrupt changes compared to AM-LSTM. This demonstrates that the information utilization capability of AMV-LSTM is stronger.

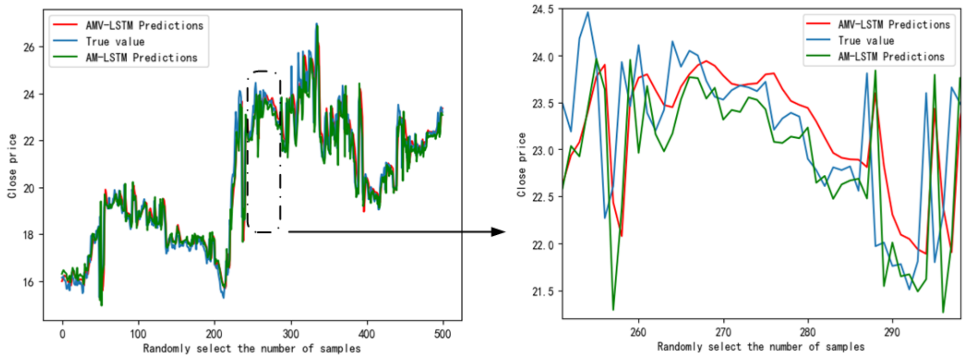


Figure 9. AMV-LSTM and AM-LSTM train set partial prediction magnification results.

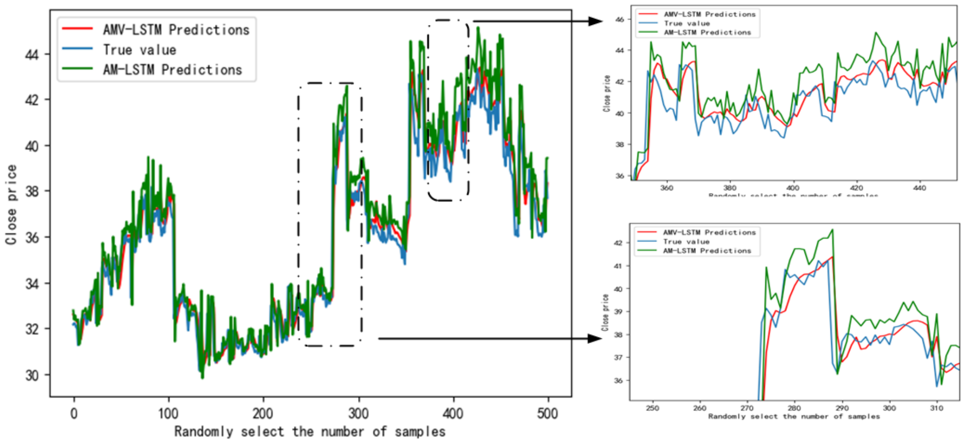
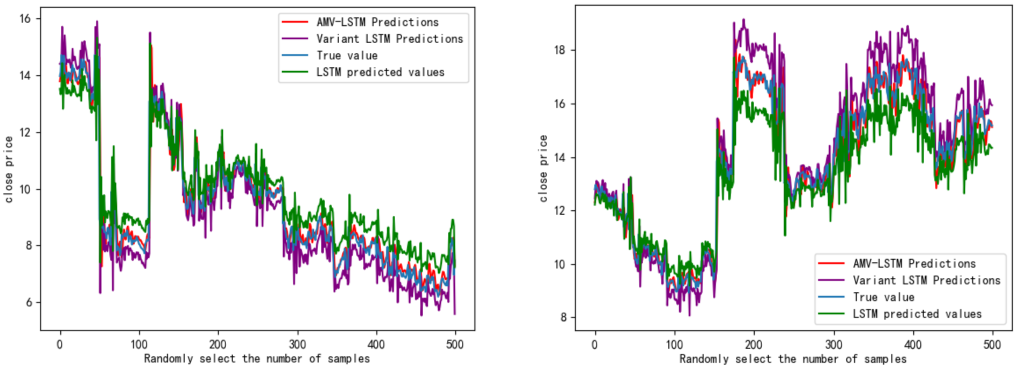


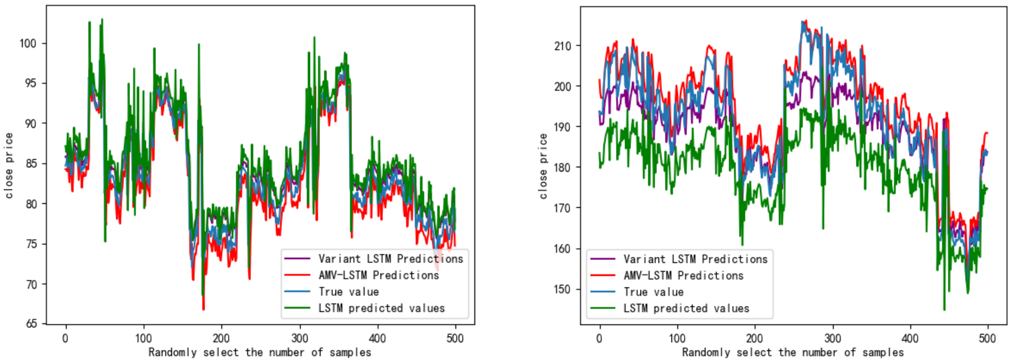
Figure 10. AMV-LSTM and AM-LSTM test set partial prediction magnification results.

To verify the applicability of the model to data from different industries and highlight the effect of integrating attention mechanisms into variant LSTMs, we used AMV-LSTM to predict the stock prices of American companies such as IBM and Ford. We compared these results with those obtained using LSTM and variant LSTMs and found that the performance of the AMV-LSTM model remained superior. The comparative plot of the model's predictive performance is shown below (Figures 11 and 12):



(**a**) (**b**)

Figure 11. Comparison of FORD stock price prediction models. (**a**) Train set; (**b**) Test set.



(**a**) (**b**)

Figure 12. Comparison of IBM stock price prediction models. (**a**) Train set; (**b**) Test set.

Furthermore, the comparison between AMV-LSTM and the more advanced model GRU is presented in the table below.

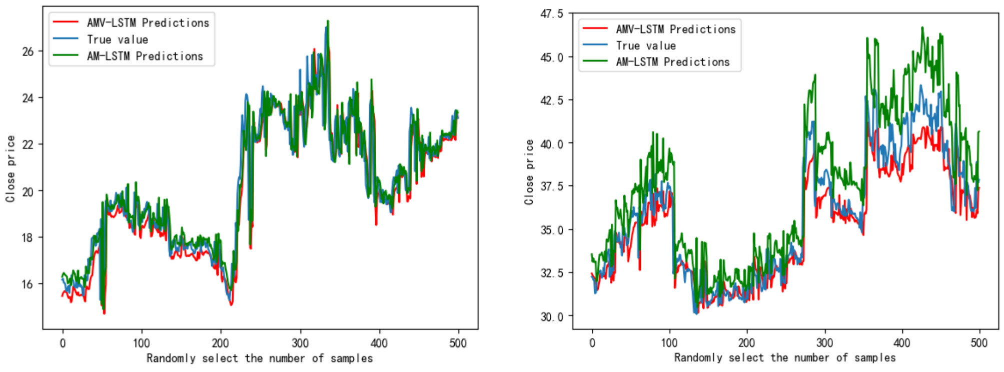
Table 3 shows that, compared to GRU, the overall performance metrics of AMV-LSTM, including MAE, MSE, and R^2, are still superior, both on the training and testing datasets.

Table 3. Evaluation index data comparison.

Evaluation Indicators	MSE (Train)	MSE (Test)	MAE (Train)	MAE (Test)	R^2 $(Train)$	R^2 $(Test)$
AMV-LSTM	0.000412	0.000638	0.013507	0.017364	0.904071	0.931898
GRU	0.00111805	0.00579771	0.0257391	0.0582083	0.750927	−1.60025

4.2.2. Predicting Stock Prices for the Next 2 Days

To assess the generalization ability of AMV-LSTM, stock price predictions for the next 2 days were conducted using both AMV-LSTM and AM-LSTM (Figure 13). This involves forecasting stock prices for the 12th days using the stock prices of the preceding 10 days.



(**a**) (**b**)

Figure 13. The AMV-LSTM and AM-LSTM predict future stock prices for the next 2 days. (**a**) Train set; (**b**) Test set.

Based on the effectiveness charts of stock price predictions for the next 2 days, both AMV-LSTM and AM-LSTM show a certain decline in predictive capabilities (Table 4). However, the predictive performance of AMV-LSTM is still superior to that of AM-LSTM.

Table 4. Predicting stock prices for the next 2 days.

Forecast the Next 2 Days	**MSE (Train)**	**MSE (Test)**	**MAE (Train)**	**MAE (Test)**	R^2 $(Train)$	R^2 $(Test)$
AM-LSTM	0.000652	0.003171	0.016602	0.046714	0.857474	0.745920
AMV-LSTM	0.000728	0.001219	0.018275	0.024840	0.864855	0.805888

The various evaluation indicators of AMV-LSTM, although showing a slight decrease in predicting the next 2 days, still maintain an accuracy of over 80%. However, the impact on the prediction accuracy of AM-LSTM is comparatively larger, especially in the performance on the test set. This validates that the generalization ability of AMV-LSTM is stronger compared to AM-LSTM. Overall, considering the comprehensive performance, AMV-LSTM still outperforms AM-LSTM.

4.2.3. Comparison of Convergence Speed

In order to compare the convergence speed of AMV-LSTM and AM-LSTM, the Mean Squared Error (MSE) in the testing set of both models is set to 0.0015. When the model's MSE reaches 0.0015, the iteration is terminated, and the number of iterations as well as the time taken for both models are recorded. Experimental results show that AMV-LSTM requires only 11 iterations, taking 321.496 s, while AM-LSTM requires 13 iterations, taking 379.821 s.

Iterative error is shown in Figure 14:

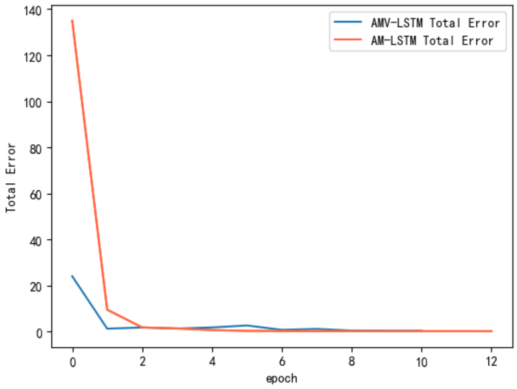


Figure 14. Iterative error.

To compare the computational complexity of the three models, we iterated LSTM, AM-LSTM, and AMV-LSTM 10 times, respectively, and recorded the time taken by each model for each iteration.

Table 5 shows that LSTM took a total of 47.2695 s for 10 iterations, with an average time of 4.72695 s per iteration. AM-LSTM took 293.034 s for 10 iterations, averaging 29.3034 s per iteration. AMV-LSTM took 291.576 s for 10 iterations, averaging 29.1576 s per iteration. Compared to LSTM, AMV-LSTM took approximately 25 s more per iteration, but achieved a nearly 77% improvement in prediction accuracy on the test set. Despite the increase in time, there was a significant improvement in accuracy. Compared to AM-LSTM, AMV-LSTM reduced the time per iteration by 0.2 s, showing a slight decrease in time, while still achieving an approximately 3% increase in accuracy on the test set. This reflects that the computational complexity of AMV-LSTM is higher than LSTM but lower than AM-LSTM.

Table 5. The time taken to iterate the number of rounds.

Running Time	LSTM	AM-LSTM	AMV-LSTM
1	4.75585	31.0443	29.3056
2	9.46196	60.1404	58.4812
3	14.2038	89.2226	87.6172
4	18.9221	118.206	116.747
5	23.6535	147.254	145.9
6	28.3667	177.226	175.061
7	33.0821	206.187	204.266
8	37.8014	235.121	233.36
9	42.5604	264.067	262.461
10	47.2695	293.034	291.576

5. Conclusions

The prediction of stock prices has always been a complex and highly challenging issue. In order to enhance the accuracy of stock price prediction as much as possible, the AMV-LSTM model was developed in this study. We first improved the structure of LSTM and designed a variant of LSTM, drawing inspiration from both classic LSTM and peephole LSTM. The variant LSTM integrates the forget gate and input gate, and introduces a "simplified" forget gate. Experimental results demonstrate that the robustness of the variant LSTM has been enhanced. The added "simplified" gate effectively filters data information, improving the efficiency of information utilization, and effectively addressing LSTM overfitting. This will directly contribute to the improvement of stock price prediction accuracy in real-world scenarios.

Additionally, an attention mechanism layer was added to the variant LSTM, resulting in the design of AMV-LSTM, for which the backpropagation formula was derived. Compared to the variant LSTM, AMV-LSTM shows overall performance improvement. Compared to AM-LSTM, AMV-LSTM demonstrates stronger robustness, excelling not only in predicting the stock price for the next day but also maintaining superior overall performance in predicting the stock price for the next two days, indicating enhanced generalization ability. This implies a longer-term reliability and predictive capability in forecasting market trends. From a decision support perspective, providing forecasts for the next two days' stock prices helps investors reduce decision-making blindness and randomness, thereby enhancing the accuracy and efficiency of investment decisions and facilitating the formulation of investment strategies. In terms of managing trading risks, it enables better avoidance of market risks, timely adjustment of investment portfolios, and higher investment returns. Compared to other benchmark models and variant models, AMV-LSTM exhibits lower Mean Squared Error (MSE) for stock price prediction than the models LSTM-CNN and LSTM+CNN+CBAM presented in [16].

Finally, in terms of convergence speed, experimental results show that AMV-LSTM requires fewer iterations than AM-LSTM to achieve the same level of accuracy, indicating that

AMV-LSTM converges faster. A faster convergence speed implies that less computational resources and time are required for real-time predictions, making investors' decisions more timely and enhancing the practicality of forecasting.

In the future, integrating the model with specific investment strategies could potentially yield higher profits for investors and enhance risk management capabilities by providing timely alerts. Additionally, applying this model to other similar fields such as runoff prediction and short-term electricity forecasting is also a viable option worth exploring. These avenues can be considered in future work.

Author Contributions: The experiments of the article were designed and conducted by S.S., who also completed the writing of the article. L.L. led and supervised the research. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the National Natural Science Foundation of China (62173222) and Shanghai University of Engineering Science Horizontal Research Project (SJ20230195).

Data Availability Statement: The data can be made available by the authors on request.

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

References

1. Bathla, G.; Rani, R.; Aggarwal, H. Stocks of year 2020: prediction of high variations in stock prices using LSTM. *Multimed. Tools Appl.* **2023**, *82*, 9727–9743. [CrossRef]
2. Huang, W.; Nakamori, Y.; Wang, S.Y. Forecasting stock market movement direction with support vector machine. *Comput. Oper. Res.* **2005**, *32*, 2513–2522. [CrossRef]
3. Shen, S.; Jiang, H.; Zhang, T. *Stock Market Forecasting Using Machine Learning Algorithms*; Department of Electrical Engineering, Stanford University: Stanford, CA, USA, 2012; pp. 1–5.
4. Zhu, C.; Yin, J.; Li, Q. A stock decision support system based on DBNs. *J. Comput. Inform. Syst.* **2014**, *10*, 883–893.
5. Guresen, E.; Kayakutlu, G.; Daim, T.U. Using artificial neural network models in stock market index prediction. *Expert Syst. Appl.* **2011**, *38*, 10389–10397. [CrossRef]
6. Jiang, W. Applications of deep learning in stock market prediction: recent progress. *Expert Syst. Appl.* **2021**, *184*, 115537. [CrossRef]
7. Nikou, M.; Mansourfar, G.; Bagherzadeh, J. Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. *Intell. Syst. Account. Financ. Manag.* **2019**, *26*, 164–174. [CrossRef]
8. Janiesch, C.; Zschech, P.; Heinrich, K. Machine learning and deep learning. *Electron. Markets* **2021**, *31*, 685–695. [CrossRef]
9. Landi, F.; Baraldi, L.; Cornia, M.; Cucchiara, R. Working Memory Connections for LSTM. *Neural Netw.* **2021**, *144*, 334–341. [CrossRef]
10. Fischer, T.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669. [CrossRef]
11. Liu, F.; Cai, M.; Wang, L.; Lu, Y. An ensemble model based on adaptive noise reducer and over-fitting prevention lstm for multivariate time series forecasting. *IEEE Access* **2019**, *7*, 26102–26115. [CrossRef]
12. Chen, X.; Huang, J.; Han, Z.; Gao, H.; Liu, M.; Li, Z.; Liu, X.; Li, Q.; Qi, H.; Huang, Y. The importance of short lag-time in the runoff forecasting model based on long hort-term memory. *J. Hydrol.* **2020**, *589*, 125359. [CrossRef]
13. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: continual prediction with lstm. *Neural Comput.* **2000**, *12*, 2451–2471. [CrossRef]
14. Yang, H.; Hu, J.; Cai, J.; Wang, Y.; Chen, X.; Zhao, X.; Wang, L. A New MC-LSTM Network Structure Designed for Regression Prediction of Time Series. *Neural Process Lett.* **2023**, *55*, 8957–8979. [CrossRef]
15. Wu, Z.; King, S. Investigating gated recurrent networks for speech synthesis. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 5140–5144.
16. Hongrui, Z.; Lei, X. Research on Stock Prediction Based on LSTM-CNN-CBAM Model. *Comput. Eng. Appl.* **2021**, *57*, 203–207.
17. Soydaner, D. Attention mechanism in neural networks: where it comes and where it goes. *Neural Comput. Appl.* **2022**, *34*, 13371–13385. [CrossRef]
18. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
19. Girihagama, L.; Naveed Khaliq, M.; Lamontagne, P.; Perdikaris, J.; Roy, R.; Sushama, L.; Elshorbagy, A. Streamflow modelling and forecasting for Canadian watersheds using LSTM networks with attention mechanism. *Neural Comput. Appl.* **2022**, *34*, 19995–20015. [CrossRef]
20. Zhou, H.; Zhang, Y.; Yang, L.; Liu, Q.; Yan, K.; Du, Y. Short-Term Photovoltaic Power Forecasting Based on Long Short Term Memory Neural Network and Attention Mechanism. *IEEE Access* **2019**, *7*, 78063–78074. [CrossRef]
21. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. *Int. Conf. Mach. Learn.* **2015**, *37*, 2048–2057.

22. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017,** *28*, 2222–2232. [CrossRef]
23. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM networks. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005; Volume 4, pp. 2047–2052. [CrossRef]
24. Kumar, I.; Tripathi, B.K.; Singh, A. Attention-based LSTM network-assisted time series forecasting models for petroleum production. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106440. [CrossRef]
25. Lin, Z.; Cheng, L.; Huang, G. Electricity consumption prediction based on LSTM with attentionmechanism. *IEE J. Trans. Electr. Electron. Eng.* **2020**, *15*, 556–562. [CrossRef]
26. Huang, J.; Niu, G.; Guan, H.; Song, S. Ultra-Short-Term Wind Power Prediction Based on LSTM with Loss Shrinkage Adam. *Energies* **2023**, *16*, 3789. [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.