mathematics

MDPI

Article

Research on Stock Index Prediction Based on the Spatiotemporal Attention BiLSTM Model

Shengdong Mu [1,2,3,†], Boyu Liu [4,*,†], Jijian Gu [2], Chaolung Lien [3] and Nedjah Nadia [5]

[1] Collaborative Innovation Center of Green Development in the Wuling Shan Region, Yangtze Normal University, Chongqing 408100, China; 20180038@yznu.edu.cn
[2] Chongqing Vocational College of Transportation, Chongqing 402200, China; xiaozhuang@cqwu.edu.cn
[3] Department of Business Administration, International College, Krirk University, Bangkok 10220, Thailand; chaolung@staff.krirk.ac.th
[4] School of Innovation and Entrepreneurship, Hubei University of Economics, Wuhan 430000, China
[5] Department of Electronics Engineering and Telecommunications, State University of Rio de Janeiro, Rio de Janeiro 205513, Brazil; nadia@eng.uerj.br
* Correspondence: liuboyu@hbue.edu.cn
† These authors contributed equally to this work.

Abstract: Stock index fluctuations are characterized by high noise and their accurate prediction is extremely challenging. To address this challenge, this study proposes a spatial–temporal–bidirectional long short-term memory (STBL) model, incorporating spatiotemporal attention mechanisms. The model enhances the analysis of temporal dependencies between data by introducing graph attention networks with multi-hop neighbor nodes while incorporating the temporal attention mechanism of long short-term memory (LSTM) to effectively address the potential interdependencies in the data structure. In addition, by assigning different learning weights to different neighbor nodes, the model can better integrate the correlation between node features. To verify the accuracy of the proposed model, this study utilized the closing prices of the Hong Kong Hang Seng Index (HSI) from 31 December 1986 to 31 December 2023 for analysis. By comparing it with nine other forecasting models, the experimental results show that the STBL model achieves more accurate predictions of the closing prices for short-term, medium-term, and long-term forecasts of the stock index.

Keywords: spatiotemporal attention; bidirectional long short-term memory network (BiLSTM); stock index prediction; graph attention network

MSC: 68V30

check for updates

Citation: Mu, S.; Liu, B.; Gu, J.; Lien, C.; Nadia, N. Research on Stock Index Prediction Based on the Spatiotemporal Attention BiLSTM Model. *Mathematics* **2024**, *12*, 2812. https://doi.org/10.3390/math12182812

Academic Editors: Jie Wen, Cai Xu and Jinrong Cui

Received: 3 July 2024
Revised: 8 September 2024
Accepted: 9 September 2024
Published: 11 September 2024

Copyright: © 2024 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/).

1. Introduction

The stock market is a complex, nonlinear, noisy, and dynamic system [1]. Most research methods have achieved good results in the short-term forecasting of stock indices. For example, Devi et al. [2] verified the superiority of the autoregressive integral sliding average (ARIMA) model in forecasting accuracy by analyzing the stock data of four companies. However, due to the highly noisy and nonlinear nature of financial time series data, it is difficult for traditional parametric equations to effectively describe these data, thus limiting the scope of application of traditional time series models. Against this background, machine learning models offer new ways of modeling and forecasting stock indices. Although shallow machine learning models can learn statistical laws from a large number of training samples, these models are limited in their ability to express complex functions with limited samples and computational resources. In contrast, deep learning can better approximate complex functions by constructing multiple hidden layers and large-scale training datasets, thus improving prediction accuracy. For example, Rather et al. [3] used a recurrent neural network (RNN) and combined it with a genetic algorithm to optimize the model and successfully improved the prediction accuracy of six stock returns. Hoseinzade

et al. [4] constructed a convolutional neural network (CNN) that significantly outperforms traditional baseline algorithms in predicting indices such as the S&P 500 and NASDAQ. Zulqarnain et al. [5] captured long-term signal correlation by using a gated recurrent unit (GRU) layer and evaluated it on the datasets of Hang Seng Index HSI, German DAX, and S&P 500 and found that the prediction accuracy of the GRU-CNN based method is extremely high. In recent years, the long short-term memory neural network (LSTM), proposed by Hochriter et al. [6], has become one of the most popular models due to its effective mitigation of the gradient vanishing problem caused by RNNs through a gate structure. A study by Fischer and Krauss [7] also confirmed that the LSTM has a better performance in predicting stock indices as compared to the Random Forest Model and the deep neural network (DNN) with higher accuracy. In addition, the bidirectional long short-term memory network (Bi-LSTM), which consists of two LSTMs operating in reverse, shows significant advantages in handling large-scale time series data. Shah et al. [8] noted that Bi-LSTM achieves predictive performance beyond that of standard LSTMs through an additional data training process. Attention mechanisms (AMs), especially spatial attention and temporal attention, are increasingly used in time series prediction. These mechanisms not only learn spatiotemporal relationships dynamically but also assign different weights to the prediction results based on the different effects of the attributes. Chaudhari et al. [9] argued that the attention mechanisms can be used as a way of resource reallocation to extract more effective feature information in the design of deep neural networks. Zhang et al. [10] proposed a hybrid LSTM that combines the spatial attention mechanism and the Bi-LSTM hybrid model, which significantly improved the prediction accuracy by assigning higher weights to key features. In addition, the temporal pattern-based attention mechanism proposed by Shih et al. [11] can learn the interdependencies between variables over multiple time steps, effectively alleviating the long-term dependency memory challenges due to training instability and gradient vanishing problems.

However, due to the difficulty in capturing the complex relationships between data structures in the stock market, most of the existing methods are unable to explicitly influence each other when modeling the potential interdependencies in data structures. Meanwhile, since graph attention networks can address asymmetric data, features between data can be better incorporated into the model if different learning weights are assigned to different neighboring nodes using attention coefficients. Based on this, we propose an unsupervised stock index prediction method based on a temporal attention mechanism, which captures complex relationships between sequences by using a multi-hop graph attention network and introduces a temporal attention mechanism to model the temporal dependence of the sequences, which effectively improves the prediction of the time series.

2. Related Work

2.1. Long Short-Term Memory (LSTM)

Hochreiter et al. [4] proposed the long short-term memory neural network LSTM, which introduces a memory store (cell) and gate structure (gate) to the traditional recurrent neural network. Cell (c_t) is used to record neuron states, which is achieved by a sigmoid neural layer and a point-by-point multiplication operation. This implementation selectively allows information to pass through the gates. Figure 1 shows the operation of the LSTM model. The LSTM is designed with two gates to control the amount of information in the memory cell state (Cell): One is the forget gate, which can be interpreted as the "memory gap", which determines how much of the "memory" of the previous cell state can be retained until the current moment. The other is the input gate, which decides how much of the new input information should be added to the cell state at the current moment, alongside the candidate gate. The candidate gate controls the fusion ratio between the "historical" information and the "current" stimulus, thus determining the amount of the new cell state to be updated. Finally, the LSTM also has an output gate that controls how much information is output from the cell state, ensuring that only the most critical information is delivered. Each gate is described in detail below. With such a gating structure, LSTM can

capture long-dependent information in the input through the gate structure, which greatly alleviates the problems of gradient vanishing and gradient explosion.

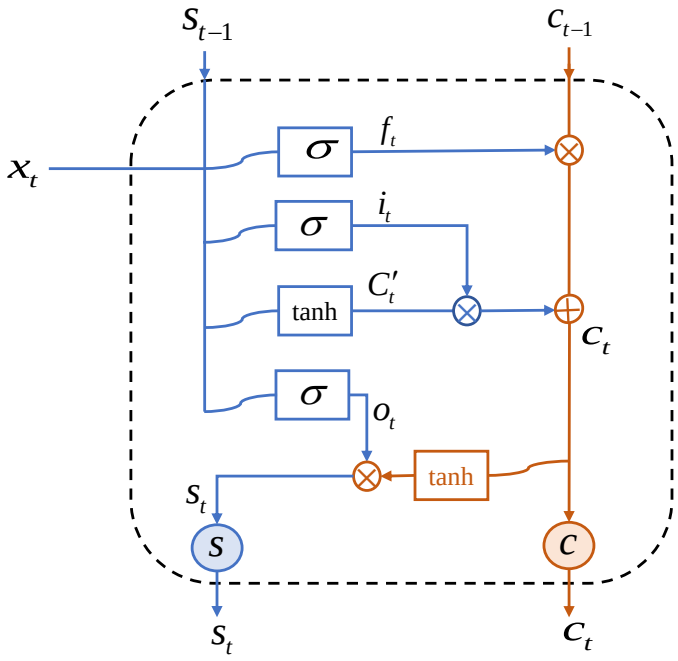


Figure 1. Long short-term memory neural network.

The forgetting gate is modeled in Equation (1), as follows:

$$f_i = \sigma\left(W_f^T \times s_{t-1} + U_f^T \times x_t + b_f\right) \tag{1}$$

The input gate is modeled in Equation (2), as follows:

$$i_r = \sigma\left(W_i^T \times s_{t-1} + U_i^T \times x_t + b_i\right) \tag{2}$$

The candidate gate is modeled in Equation (3), as follows:

$$C'_t = \tanh\left(W_c^T \times s_{t-1} + U_c^T \times x_t + b_c\right) \tag{3}$$

The model function of the memory cell is shown in Equation (4), as follows:

$$C_t = f_t \times C_{t-1} + i_t \times C'_t \tag{4}$$

The output gate is modeled in Equation (5), as follows:

$$O_t = \sigma\left(W_o^T \times s_{t-1} + U_o^T \times x_t + b_o\right) \tag{5}$$

The final output on the time series is shown in Equation (6), as follows:

$$s_t = O_t \times \tanh(C_t) \tag{6}$$

The loss function is shown in Equation (7), as follows:

$$\min J(\theta) = \sum_{t=1}^{T} \text{loss}\left(\hat{y^{(t)}}, y^{(t)}\right) \tag{7}$$

2.2. Bidirectional Long Short-Term Memory Neural Network (Bi-LSTM)

In traditional time series processing, long short-term memory networks (LSTMs) usually only utilize historical information and ignore future data. The bidirectional long short-term memory neural network (Bi-LSTM) [12,13] uses two separate hidden layers based on LSTM to process the sequence data in both forward and backward directions, connects the two hidden layers to the same output layer, and stores both previous and latter information as the current time base of the time series data, so theoretically, the prediction performance will be better than the unidirectional LSTM. Bi-LSTM's hidden layer output includes the activation output of the forward hidden layer and the activation output of the backward hidden layer. Figure 2 illustrates the operation mechanism of the Bi-LSTM model, showing how information from both directions is integrated and jointly affects the output layer, providing a deeper understanding and enhanced prediction capability for time series data.

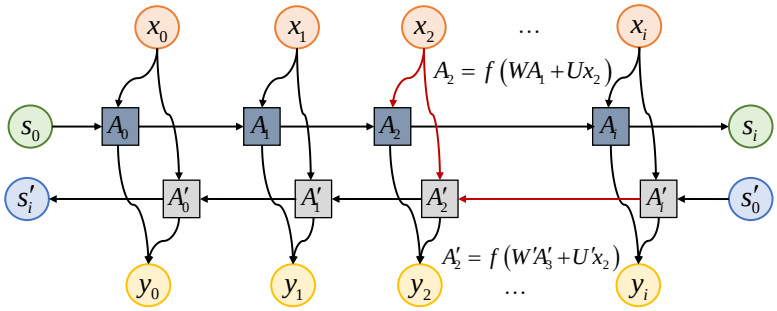
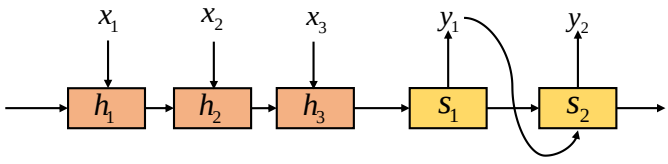


Figure 2. Structure of the bidirectional Bi-LSTM.

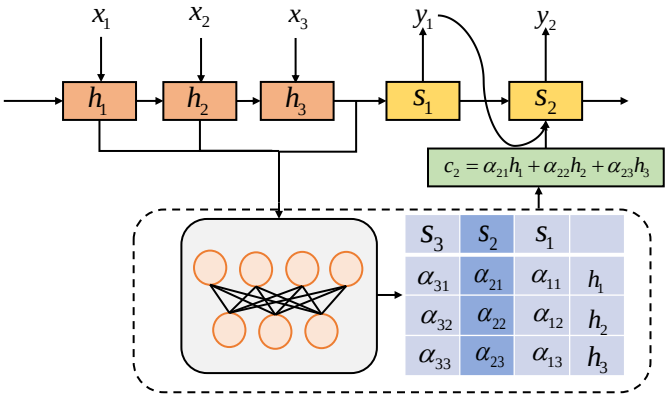
2.3. Attention Mechanisms

Bi-LSTM cannot capture the different contributions of different time points and different input features to the closing price. The attention mechanism [14] is a model that simulates the attention mechanism of the human brain, which can be well-optimized for traditional models by calculating the probability distribution of attention, highlighting the role of a critical input on the output, and allocating the computational resources to more important tasks with limited computational power. In this context, this paper proposes an innovative spatiotemporal attention mechanism specifically designed to capture the dynamic spatiotemporal correlations in the stock market, which combines both spatial and temporal attention to more accurately identify and utilize the information that is most critical for prediction. Figure 3 illustrates a comparison between the traditional model and the model after the introduction of the attention mechanism. Where Figure 3a shows the traditional model and Figure 3b shows the model after the introduction of the attention mechanism.



(**a**)

Figure 3. *Cont.*



(**b**)

Figure 3. Comparison between the traditional model and the model after the introduction of the attention mechanism. (**a**) Traditional model; (**b**) model after introducing the attention mechanism.

2.3.1. Spatial Attention Mechanisms

In the spatial dimension, the interactions between different input features are highly dynamic. For this reason, this paper uses a spatial attention mechanism to adaptively capture the dynamic correlations between nodes in the spatial dimension. This mechanism measures the impact of each external feature on the closing price by representing the attention weights that sum to one. For each time step, t, the weights are used to measure how much attention a node should focus on the state of another node in order to predict its state to compute the future closing price.

2.3.2. Temporal Attention Mechanisms

Although Bi-LSTM cells can maintain long-term dependencies by storing temporal information through their cell structure and controlling the increase or decrease in information through a gate mechanism, the temporal information controlled through gates causes the Bi-LSTM to adjust its cell state in each time window, T, a final state that tends to retain information about the most recent inputs compared to the historical inputs. In a short time series of predictions, this property does not cause serious bias. However, over longer prediction horizons, the standard model may underestimate the impact of earlier states. Temporal attention mechanisms are used to understand the influence of these hidden states in each time window. In this paper, Bi-LSTM cells store temporal information while the attention mechanism is used to assess the importance of these different cell states on the closing price prediction.

2.3.3. Spatiotemporal Attention Mechanisms

Spatial correlations are represented through the spatial attention mechanism, which assigns attention weights to primitive attributes. Temporal relevance is then realized through the temporal attention mechanism by assigning attention weights to the hidden states in spatial attention. The spatiotemporal attention mechanism combines spatial and temporal correlations, allowing the neural network to automatically focus more attention on valuable information.

3. Description of the Problem and Construction of the Model

3.1. Description of the Problem

The architecture of the STBL model proposed in this paper is shown in Figure 4. The training data in the figure are denoted as $x_{\text{train}} = [x^1, \cdots, x^T_{\text{train}}]$, where T_{train} denotes the length of the training set, and $x^t \in R^N$ represents the value of the N-dimensional time series at any given moment. The model learns from the multivariate time series of the training data and then forecasts the stock index using the test data. In addition, $y(t) \in \{0, 1\}$ is

used to indicate the presence or absence of anomalies at each test timescale, where $y(t) = 1$ indicates that noisy data occurred at moment t.

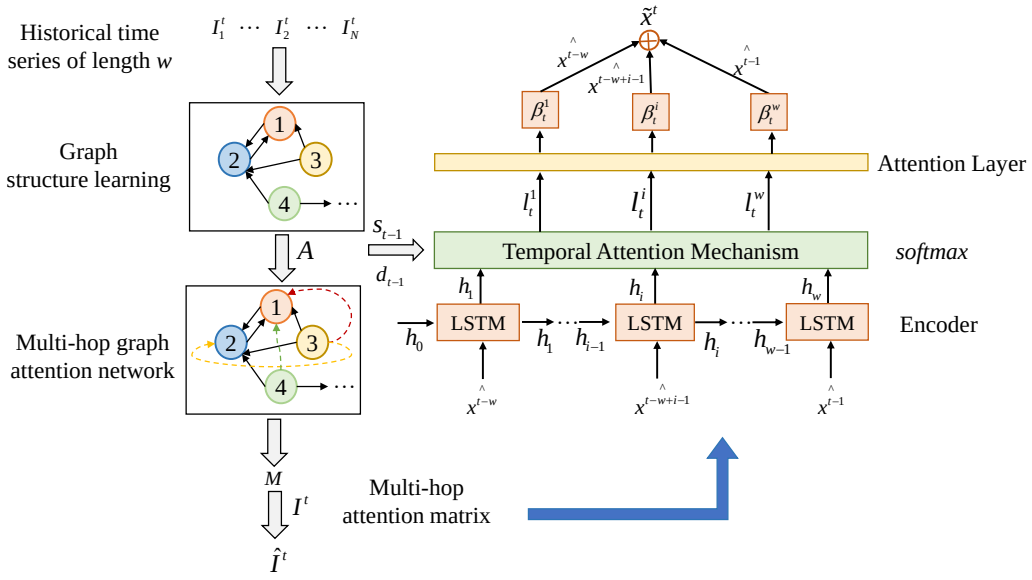


Figure 4. Visualization of the STBL architecture.

The model uses a prediction-based approach to define its inputs, and the inputs to the model are defined at moment t based on historical time series data with a specific window size, s, as shown in Equations (8) and (9):

$$I^t := \left[x^{t-s}, x^{t-s+1}, \cdots, x^{t-1}\right] \tag{8}$$

$$I^t \in R^{N \times s} \tag{9}$$

The target output of the model is to predict the expected value of each data point at moment t, i.e., $\tilde{x}^t$.

3.2. Methodological Architecture

The STBL architecture consists of three core components: graph structure learning, multi-hop graph attention network, and the LSTM-based temporal attention mechanism. The input at moment t consists of historical time series data I^t of size s. The graph structure learning component learns and constructs a directed graph structure between data nodes by calculating the similarity among the data structure nodes, represented by the adjacency matrix A. This component is responsible for learning the dependency graph between the data structures. Building on this, the multi-hop graph attention network learns the dependency graph between data structures and, through this learning process, generates a multi-hop attention matrix M for updating the node features [15], which enables the network to efficiently capture and reinforce important feature relationships and enhances the model's ability to understand the data dynamics. For the updated node data, STBL uses an LSTM-based temporal attention mechanism to capture the temporal relevance of the sequence. This mechanism optimizes the model's ability to process information in the time dimension, making it more sensitive to changing trends and patterns in the time series. Ultimately, STBL performs stock index forecasting on the multivariate time series by predicting the data value $\tilde{x}^t$ at the next time point.

3.3. Graphical Structure Learning

STBL requires graphs of relationships between data structures. In stock market analyses, the relationships between data structures are often not predefined, and this study learns the relationships between these data structures through cosine similarity. Given

the often asymmetric nature of these relationships, this paper uses a directed graph, G, to characterize the relationships between data structures, where each node represents a data structure, and directed edges pointing from one node to the other denote the effect of the former on the latter. In order to describe this relationship graphically, we use the adjacency matrix A to represent the directed graph, G. $A_{ij} = 1$ indicates that a directed edge from node i to node j exists, and conversely, $A_{ij} = 0$ indicates that such an edge does not exist.

In the absence of a priori knowledge, the neighborhood candidate set of each data structure i is defined to include all other data structures except itself, as illustrated by Equation (10), as follows:

$$C_i \subseteq \{1, 2, \cdots, N\} \backslash \{i\} \tag{10}$$

where the symbol $\backslash$ indicates that the set $\{1, 2, \cdots, N\}$ does not contain element i.

Further, introducing an embedding vector for each data structure node not only represents the characteristics of that node but also characterizes the dependency between data structures by calculating the similarity between the embedding vectors. We define $v_i \in R^d$ to denote the embedding vector of the ith data structure node, where $i \in \{1, 2, \cdots, \mathrm{N}\}, \mathrm{N}$ denotes the number of data structure nodes, and d denotes the dimension of the embedding vector. The similarity between a data structure node i and its candidate node j is the cosine similarity between the two nodes, denoted as Equation (11), as follows:

$$e_{ij} = \cos\left[v_i, v_j\right] = \frac{v_i \cdot v_j}{\|v_i\| \times \|v_j\|} \tag{11}$$

The similar nodes in the top k of the ranking are selected from the candidate set of node i to build a directed edge, as shown in Equation (12):

$$A_{ij} = 1\{j \in \mathrm{Top}\, K(\{e_{ik} \mid k \in \mathbb{C}_i\})\} \tag{12}$$

This approach not only constructs directed graphs characterizing the relationships between data structures, but also provides an efficient analytical framework for stock index forecasting based on complex relational graphs.

3.4. Multi-Hop Graph Attention Networks

In order to deeply explore the spatial correlation between multivariate time series, the STBL model employs a multi-hop graph attention network to capture the complex dependencies between data structures, which is especially critical for stock index forecasting. Based on the established directed graph structure (as defined in Equation (12)), STBL first uses a graph attention network (GAT) to learn the attention weights between neighboring data structure nodes to generate a single-hop attention matrix $M^1 \in R^{N \times N}$. In the traditional assumption of graph neural networks, the representation of a node is significantly influenced by its neighboring nodes in the graph structure, and the node features are usually updated by the features of the single-hop neighbor nodes. Graph attention networks add an attention mechanism to graph neural networks, enabling the model to assign unique attention weights to each single-hop neighbor node based on the dependencies between data structure nodes [16]. In stock market analysis, it can fine-tune the model's perception of the relationship between different market factors, thus improving the accuracy and sensitivity of the prediction. The specific calculation process is shown in Figure 5. At the moment t, the graph attention network can calculate the attention coefficient $\alpha_{i,j}$ between node i and its neighboring nodes, and this calculation is shown by the following Equations (13)–(15), which enable the STBL model to accurately understand and predict the dynamics of the stock index:

$$g_i^t = v_i \oplus WI_i^t \tag{13}$$

$$\pi(i, j) = \mathrm{LeakyReLU}\left(a^T\left(g_i^t \oplus g_j^t\right)\right) \tag{14}$$

$$\alpha_{i,j} = \frac{\exp(\pi(i,j))}{\sum_{k \in N_i \cup (i\}} \exp(\pi(i,k))} \tag{15}$$

where $j \in N_i$. During the computation, $I_i^t \in \mathbb{R}^w$ is the input feature of data structure node i at moment t. $N_i = \{j | A_{ij} > 0\}$ is the set of neighboring nodes of node i learned from the adjacency matrix A. The shared parameter $W \in R^{d \times w}$ is used in the model for the linear transformation of the features of each node, which enables the dimensionality enhancement of the features. v_i is the embedding vector of the data structure node i, which characterizes the different features of different data structure nodes. $\oplus$ denotes splicing, and g_i^t connects the data structure embedding v_i with the corresponding transformed feature WI_i^t, where a is used to map the spliced high-dimensional features to a real number, which is further computed to obtain the attention coefficients. In order to compute these coefficients efficiently, the model employs LeakyReLU as a nonlinear activation function and normalizes the attention coefficients using the softmax function. Further, the single-hop attention matrix M^1 can be obtained by $\alpha_{i,j}$.

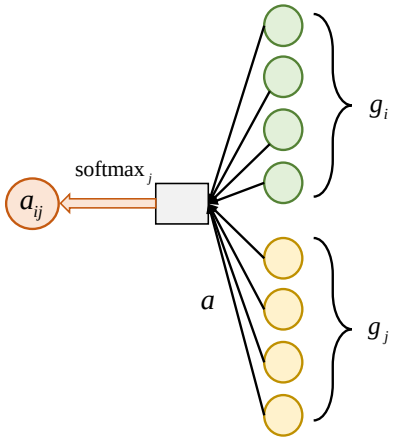


Figure 5. Calculation process of the attention coefficient.

In order to capture the important information between nodes that are not directly connected in the graph structure, this paper introduces the concept of multi-hop neighbor nodes to compute the multi-hop attention weight matrix M through an attention diffusion process, which is implemented based on the power expansion of the single-hop attention matrix M^1. For the depth of the multi-hop graph attention network, this study considers up to three-hop neighbor nodes, and the calculation processes are shown in Equations (16) and (17), as follows:

$$M = \sum_{i=0}^{3} \theta_i M^i \tag{16}$$

$$\sum_{i=0}^{3} \theta_i = 1 \tag{17}$$

where $1 \geq \theta_i > 0, M^0 = I$, and I is the unit matrix. Here, θ_i is the attention attenuation factor, and $\theta_i > \theta_{i+1}$, reflecting the fact that the weight corresponding to more distant nodes should be smaller in the message aggregation process; this is because the information transfer between distant nodes may not be as important or direct as that of the near nodes [17]. Due to the different dependencies between data structures in different datasets, θ_i is taken after comparing the experimental results.

The input feature vectors can be mapped to the aggregated representation $\hat{I}_i^t$ through the multi-hop attention matrix M. Using these attention weights, the model can adaptively

aggregate the spatial features of the data structure nodes as shown in Figure 6. The aggregated representation of a specific data structure, node i, is shown in Equation (18):

$$\hat{I}_i^t = \sum_{j \in [1,N]} M_{i,j} I_j^t \tag{18}$$

where $M_{i,j}$ is the attentional weight of node j on node i. To ensure the weights are normalized, we set the condition that the sum of the weights equals 1, as shown in Equation (19):

$$\sum_{j=1}^{N} M_{ij} = 1 \tag{19}$$

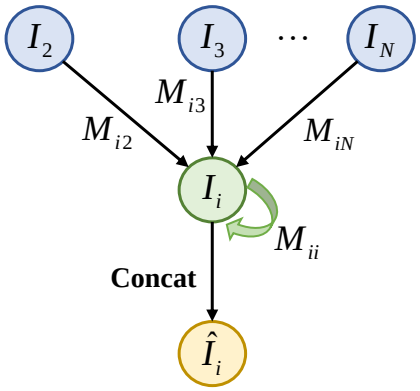


Figure 6. Aggregated representation of data node i.

3.5. Spatiotemporal Attention Mechanisms Based on Long Short-term Memory

In the STBL model, the multi-hop graph attention network focuses on learning the inter-dimensional dependencies between multivariate time series. In addition, STBL employs an LSTM-based temporal attention mechanism to accurately capture the temporal information of the sequences, which is crucial for stock index prediction. Unlike traditional methods that use only LSTM for encoding and forecasting [18], the STBL model only uses LSTM units as encoders and selects the temporal component that is the most important for the prediction of the next time step by learning the hidden states of the input sequence, thus calculating the attentional weights of each state. Specifically, the input sequence at moment t is represented as shown in Equation (20):

$$\hat{I}_i^t := \left[x^{\hat{t-s}}, x^{t-s+1}, \ldots, x^{\hat{t-1}} \right] \tag{20}$$

These sequences are used to predict $\tilde{x}^t$. Firstly, the input sequence is encoded as a series of hidden states, defining the w hidden states as $\{h_1, h_2, \cdots, h_w\}$, and then the ith hidden state is updated as shown in Equation (21):

$$h_i = f_1\left(h_{i-1}, x^{t-w+i-1}\right) \tag{21}$$

where f_1 is an LSTM cell. Next, with reference to the previous hidden state $d_{t-1} \in R^P$ and the unit state $s_{t-1} \in R^P$ in the encoder LSTM cell, the attentional weight of each encoder hidden state is computed as shown in Equations (22) and (23):

$$l_t^i = v_d^{\mathrm{T}} \tanh(W_d[d_{t-1}; s_{t-1}] + U_d h_i) \tag{22}$$

$$\beta_t^i = \frac{\exp\left(l_t^i\right)}{\sum_{j=1}^{w} \exp\left(l_t^j\right)} \tag{23}$$

where $1 \leq i \leq w$ and $v_d \in R^m, W_d \in \mathbb{R}^{m \times 2p}, U_d \in \mathbb{R}^{m \times n}$ are the parameters to learn, and β_t^i denotes the importance of the ith encoder hidden state at time t for prediction. Since each encoder's hidden state is mapped to a time component of the input, this attention weight can be used on the corresponding time component to adaptively select the corresponding input sequence across all time steps. Using the attention weights β_t^i, the weighted average and predictable $\tilde{x}^t$ of $\hat{I}_i^t$ is computed as shown in Figure 7. The specific calculation is shown in Equation (24):

$$\tilde{x}^t = \sum_{i=1}^{w} \beta_i^i x^{t-w+i-1} \tag{24}$$

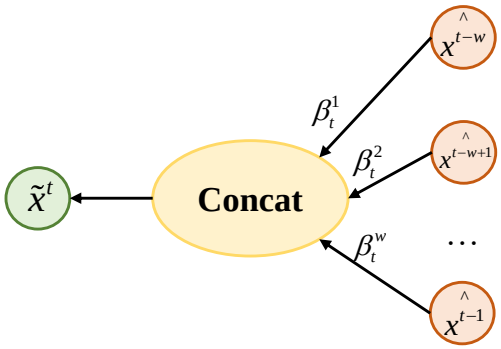


Figure 7. Calculation process of $\tilde{x}^t$.

In addition, in order to optimize the model performance, a mean square error loss function is used to minimize the difference between the actual and predicted stock indices as shown in Equation (25):

$$L_{\text{MSE}} = \frac{1}{T_{\text{train}} - w} \sum_{t=w+1}^{T_{\text{train}}} \left\| \tilde{x}^t - x^t \right\|_2^2 \tag{25}$$

Our proposed method not only strengthens the model's ability to capture temporal information but also greatly improves the accuracy of stock index prediction by precisely adjusting the importance of the input sequence at each time step. The training process of the STBL model is shown in Algorithm 1, which describes in detail the execution steps of the whole algorithm.

Algorithm 1 STBL Training Flow

Input: Training data $x_{\text{train}} = \{(x^1, \ldots, x_{\text{train}}^T)\}$; sliding window size s
for each epoch **do**
  for $t = w+1, \ldots, T$ in x_{train} **do**
    Construct input data $I^t = [x^{(t-s)}, x^{(t-s+1)}, \ldots, x^{(t-1)}]$
    Build directed graph G using embedding vectors v_i, $i \in \{1, 2, \ldots, N\}$ and adjacency matrix A
    Compute multi-hop attention weights matrix M using Equations (13) to (17)
    Compute the aggregated representation $\hat{I}_i^t$ using Equation (18)
    Update each time step's prediction $\tilde{x}^t$ using Equations (21) to (24)
  end for
  Optimize using Adam by minimizing the loss function given in Equation (25)
end for

4. Model Validation

4.1. Selection of Data Sets

In order to investigate the applicability and validity of the STBL model for the forecasting of real stock indices, this part applies the STBL to the forecasting of the daily closing

price of the Hong Kong Hang Seng Stock Index (HIS), with the data sample selected from the interval of 31 December 1986 to 31 December 2023 [19]. The response variable used in the study is the daily closing price of the Hang Seng Index, whereas the influencing factors include the market factor and technical factor, totaling 29 different indicators, the details of which are listed in Table 1. All data are sourced from the RESET Financial Database.

Table 1. Hong Kong Hang Seng Index characteristic vectors.

Eigenvectors	Indicator	Eigenvectors	Indicator	Eigenvectors	Indicator
x_1	Opening price	x_{11}	5-day average volume	x_{21}	10 days offline
x_2	Highest price	x_{12}	10-day average volume	x_{22}	20 days offline
x_3	Lowest price	x_{13}	20-day average volume	x_{23}	Momentum 5
x_4	Closing Price	x_{14}	5-day Exponential Moving Average	x_{24}	Momentum 10
x_5	Volume	x_{15}	10-day Exponential Moving Average	x_{25}	Momentum 15
x_6	Turnover	x_{16}	12-day Exponential Moving Average	x_{26}	MB
x_7	Up and Down	x_{17}	20-day Exponential Moving Average	x_{27}	UP
x_8	5-day SMA	x_{18}	26-day Exponential Moving Average	x_{28}	DN
x_9	10-day SMA	x_{19}	MACD	x_{29}	RSI
x_{10}	20-Day SMA	x_{20}	5 days offline		

In this paper, the following criteria are used to select the dataset: firstly, the initial data with zero daily turnover are excluded, and all the trading data from 31 December 1986 to 31 December 2023 are selected. In order to fully assess the efficacy of the prediction model on different time scales, this paper conducts prediction analysis on the closing prices of the stock index on the 1st day (the next day), the 7th day, the 30th day, the 60th day, and the 120th day, respectively. In this study, 25% of the data are set as part of the test set, from the remaining data, 15% are classified as part of the validation set, and the remaining 60% of the data are used as part of the training set to train the model. Additionally, to optimize the training efficiency and computational accuracy of the model, this study normalized the data before model fitting by mapping all the raw data to the [0,1] interval through a linear transformation. The aim of this study is to propose the STBL model for predicting the dynamics of stock indices.

4.2. Selection of Assessment Indicators

In order to comprehensively assess the effectiveness of the STBL model for stock index closing price predictions, this paper uses the following evaluation metrics to measure the proposed model: the mean absolute percentage error (MAPE), as shown in Equation (26), root mean square error (RMSE), as shown in Equation (27), and the mean absolute error (MAE), as shown in Equation (28). These metrics are used for quantitatively evaluating the model's prediction accuracy. Smaller values of MAPE, RMSE, and MAE indicate higher accuracy [20].

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100\% \tag{26}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{27}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{28}$$

In addition to the quantitative error measures, to further validate the model's ability to capture market dynamics, this paper will also predict the upward and downward trends of stock indices. This section will use the prediction classification accuracy as a measure to assess the accuracy of the model in predicting stock market trend movements.

4.3. Setting of Parameter Operating Environment

The running environment for training the model was as follows: 2.4 GHz quad-core Intel Core i7 processor with 8 GB 2133 MHz LPDDR4 RAM. The development and training of the model were carried out in a Python language environment, using TensorFlow 2.2.0 as the deep learning framework.

Firstly, Adam was chosen as the optimizer in this study. Adam combines the advantages of both the adaptive gradient algorithm and root mean square propagation strategies. For example, (1) the adaptive gradient algorithm retains a learning rate for each parameter to enhance performance over sparse gradients. (2) The root-mean-square propagation adaptively reserves a learning rate for each parameter based on the mean of the nearest magnitude of the weight gradient, making it perform well when dealing with non-stationary and non-linear problems. Moreover, Adam's algorithm is easy to implement and has high computational efficiency and low memory requirements.

When constructing the Bi-LSTM model, parameters need to be set, including the number of time steps in the window, denoted as T. This paper evaluates the short-, medium-, and long-term prediction effects of the model by choosing $T \in \{1, 7, 30, 60, 120\}$ to calculate the model performance, respectively. The number of hidden layers in each attention module, the number of hidden layers in the encoder m, and the number of hidden layers in the decoder p are set. In this paper, we set $m = p \in \{16, 32, 64, 128, 256\}$, where the best performance is obtained on the validation set when $m = p = 128$ for evaluation purposes.

The main structure of spatial–temporal-Bi-LSTM constructed in this paper consists of a fully connected layer, a spatial attention mechanism, a Bi-LSTM recursive layer, a temporal attention mechanism, and another Bi-LSTM layer. To prevent overfitting, a dropout strategy is applied during each training batch to randomly deactivate some hidden layer nodes. In this paper, the dropout rate is set to 0.2. To eliminate the effect of randomness in the training process, each model is trained 10 times, and the results are recorded and averaged to evaluate the overall performance of the model.

4.4. Analysis of Results

This paper presents a comprehensive comparison of ten models to evaluate their performance in stock index prediction. These include six baseline models: the support vector regression machine (SVR), convolutional neural network (CNN), gated recurrent unit (GRU), standard long short-term memory network (LSTM), bidirectional long short-term memory network (Bi-LSTM), and bidirectional long short-term memory network incorporating the principal component analysis (PCA-Bi-LSTM). In addition, four models that introduce attention mechanisms are examined: a bidirectional long short-term memory network incorporating a spatial attention mechanism (Spatial-Bi-LSTM), a bidirectional long short-term memory network incorporating a temporal attention mechanism (Temporal-Bi-LSTM), a standard LSTM model incorporating a spatiotemporal attention mechanism (spatial–temporal-LSTM), and a Bi-LSTM model incorporating a temporal attention mechanism (spatial–temporal-Bi-LSTM). When the step size T = 7, the RMSE and MAE values of the STBL model are 10.0881 and 260.559, respectively. Compared to the worst-performing CNN model, the RMSE and MAE of the STBL model improved by 71.19% and 72.08%, respectively. Additionally, the MAPE value of the STBL model is 1.0312%, which is a 2.5% improvement over the underperforming SVR model. When the step size is increased to 60, the performance of the STBL model remains outstanding. Regarding the number of iterations, when T = 60, the model's performance gradually stabilizes and begins to decline after 50 iterations.

To ensure the accuracy of the evaluation results and comparability with the original data, the prediction results of all models were back-normalized. Table 2 demonstrates the prediction performance of these models at different time steps (T = 7 and T = 60). Overall, shallow learning models such as SVR perform poorly in all time steps; in contrast, deep learning models such as CNN, GRU, and LSTM outperform the shallow learning models. In particular, the Bi-LSTM model outperforms the unidirectional LSTM in terms

of accuracy due to the addition of an inverse structure to the standard LSTM, enabling it to capture spatial features and bidirectional temporal dependencies from historical data more efficiently. Similarly, the spatial–temporal-Bi-LSTM outperforms the spatial–temporal-LSTM. Moreover, models that incorporate the attention mechanism generally outperform those without it in terms of prediction accuracy, demonstrating its advantage in handling complex data features.

Table 2. Comparison of the predictive effectiveness of 10 models (T = 7 and T = 60).

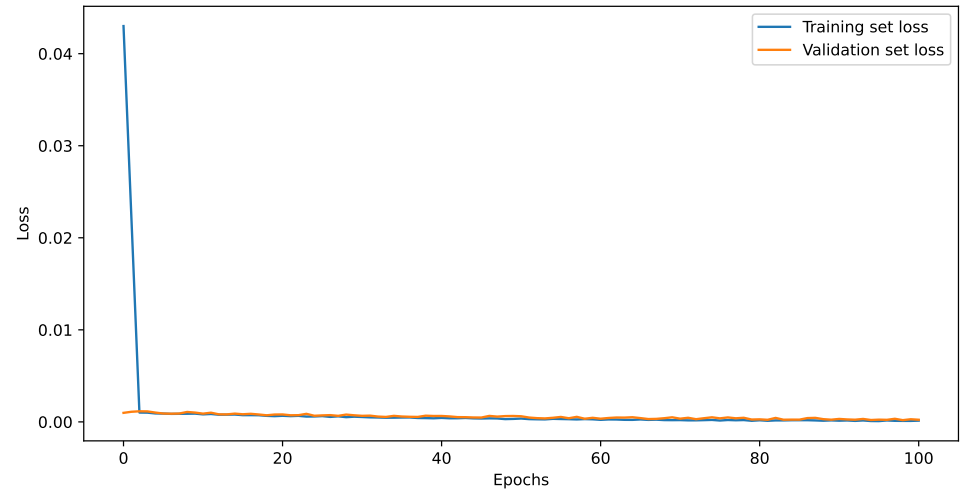
Modeling	T = 7				T = 60			
	RMSE	**MAE**	**MAPE (%)**	**Dacc**	**RMSE**	**MAE**	**MAPE (%)**	**Dacc**
SVR	34.8216	927.5214	3.4802	0.5131	54.1835	1270.6599	6.0231	0.5100
CNN	35.0903	933.6563	3.4517	0.5151	50.3149	1134.7480	5.6600	0.5151
GRU	32.0672	826.5044	3.0804	0.5365	36.1998	1024.3891	4.6063	0.5151
LSTM	29.4284	732.2569	2.7479	0.5345	35.3955	830.3973	3.3313	0.5192
Bi-LSTM	22.3612	460.5986	1.7585	0.5477	30.5073	581.3288	2.6449	0.5222
PCA-Bi-LSTM	20.8834	428.8445	1.7320	0.5630	29.6777	530.8796	2.2103	0.5396
Spatial-Bi-LSTM	19.4792	397.9046	1.5626	0.5794	26.3059	559.4067	1.8166	0.5467
Temporal-Bi-LSTM	14.1680	371.2324	1.4729	0.5702	15.6761	413.1557	1.5790	0.5600
Spatial–Temporal-LSTM	11.5562	304.6775	1.5596	0.5773	16.4698	335.4199	1.5290	0.5600
Spatial–Temporal-Bi-LSTM	10.0881	260.5592	1.0312	0.5804	13.5692	400.3019	1.3066	0.5630

To comprehensively assess the effectiveness of the STBL model in stock index prediction, Figures 8 and 9 are used in this study to show the trend of the training and validation set losses with the number of iterations at different time steps (T = 7 and T = 60). At the short time step, T = 7, the effect of the temporal attention mechanism is not significant, and the training and validation set losses show a gradual decrease regardless of whether the temporal attention mechanism is applied or not. However, the situation is different at the longer time step, T = 60. If the temporal attention mechanism is not used, the validation set loss instead shows a rising trend as the number of iterations increases; while the model using the temporal attention mechanism shows a continuous decreasing trend in the validation set loss even at longer time steps, and stabilizes after 50 iterations. This suggests that in multi-step prediction, the temporal attention mechanism can effectively capture the long-term dependency between the data, and thus the performance of the model on the training and validation sets gradually improves as the number of iterations increases.

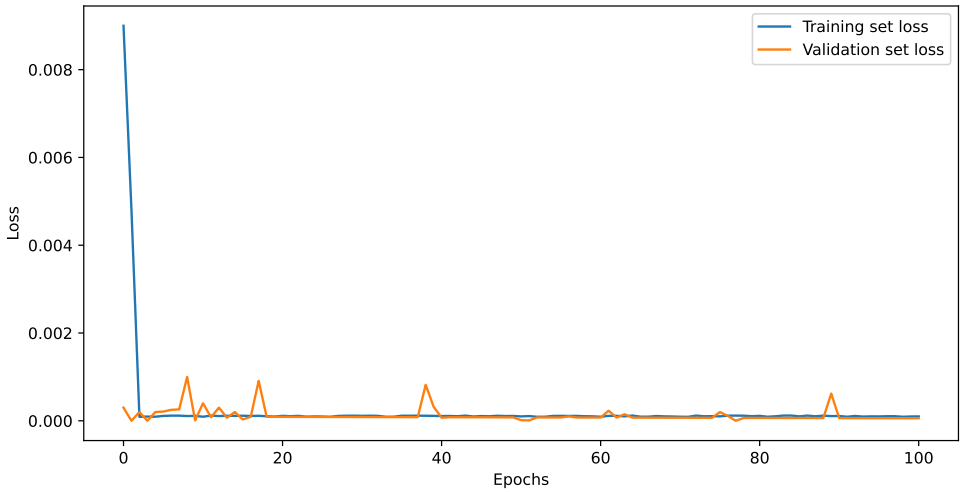
Model complexity is primarily defined by the computational load during the forward pass and the number of parameters in the model. FLOPS (floating-point operations) serve as a crucial metric for assessing computational complexity. The STBL model has a FLOPS value of 76.9 million, indicating that it requires approximately 76.9 million floating-point operations to perform specific tasks or during forward propagation. A higher FLOPS value generally means that the model requires more computational resources, which can lead to increased running times.

The parameter count (Params) reflects the total number of learnable parameters in the model. The STBL model has 29.4 million parameters, including weights and biases. These parameters are essential elements in the learning process of the model, determining not only its learning capacity and expressive power but also its ability to generalize. While a larger number of parameters can enhance the model's learning capabilities, it also increases the demand for memory and may lead to overfitting, especially in cases with limited data.

By comparing the complexities of different models in Table 3, it is evident that the STBL model has a relatively low FLOPS value of 76.9 M. This typically indicates that the model operates faster and requires less computational demand. Additionally, the STBL model's access disc value of 29.4 M is comparatively low, suggesting that it requires fewer memory resources. This not only helps reduce the computation time but also alleviates overfitting issues, making the model more suitable for use in resource-constrained environments.



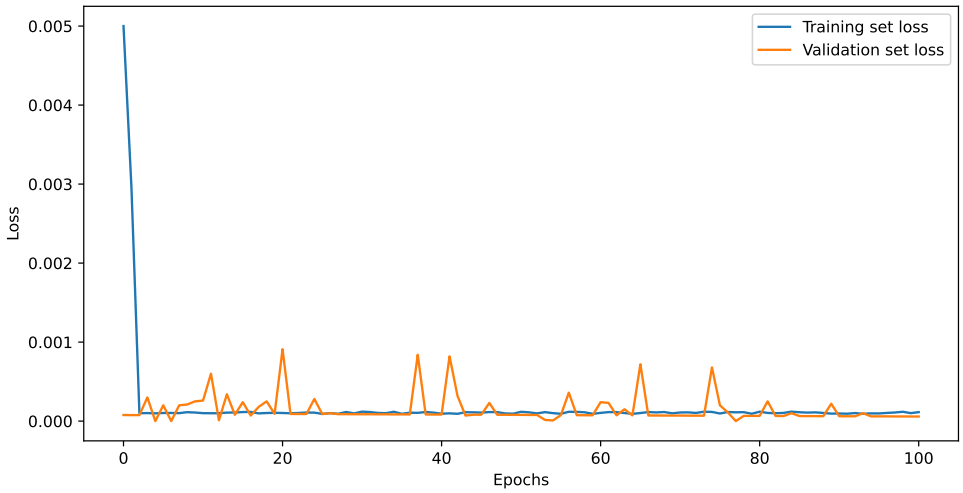
(**a**)



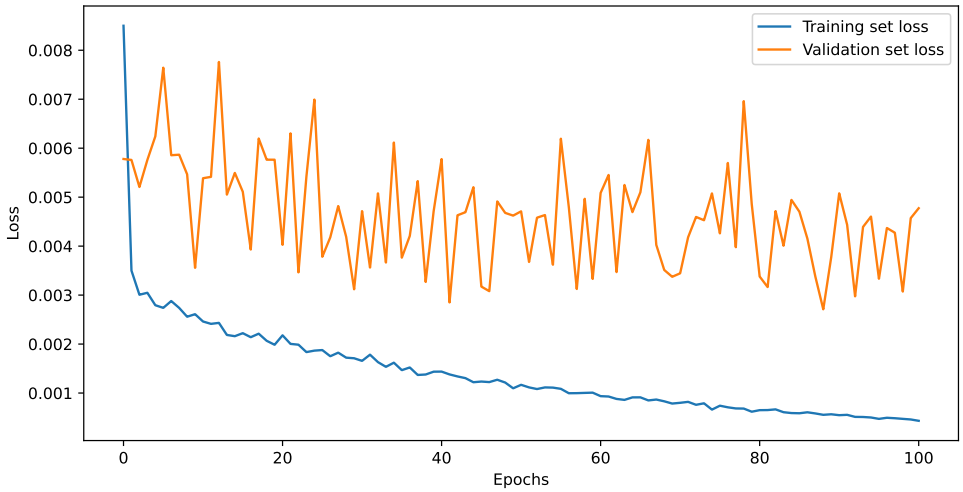
(**b**)

Figure 8. Training set loss function and validation set loss function for T = 7. (**a**) Includes temporal attention mechanisms; (**b**) does not include attention mechanisms.

Figure 10a,b shows the variations of RMSE and MAE for different models when increasing the time step. Models including BiLSTM, Spatial-BiLSTM, Temporal-BiLSTM, STBL, and PCA-BiLSTM are compared. By comparing with principal component analysis (PCA) methods, this study assesses the effectiveness of feature screening methods. The visualization of attention weights not only shows how much attention each feature receives but also helps to identify the most critical features. Comparing the PCA-BiLSTM and Spatial-BiLSTM models shows that feature screening using the spatial attention mechanism achieves lower prediction errors at any time step. In particular, the prediction accuracy of PCA-BiLSTM is lower than that of the original BiLSTM at T = 1. This may be because the PCA may cause the input data to lose some of the valid information when predicting at a single step, which in turn reduces the prediction accuracy of the model. On the contrary, through the spatial attention mechanism, the model can effectively screen and analyze the input features, extract the data structure of the input indicators more efficiently, and improve the attention to the key variables, while reducing unnecessary computation of the non-informative features during the training process of the model, which in turn reduces the risk of overfitting and high computational costs, and achieves higher prediction accuracies.



(**a**)

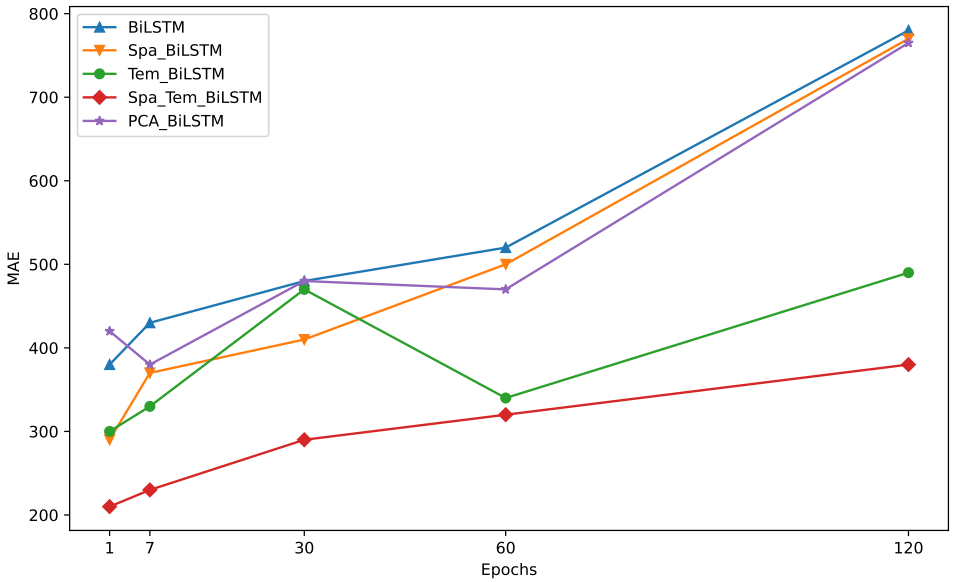


(**b**)

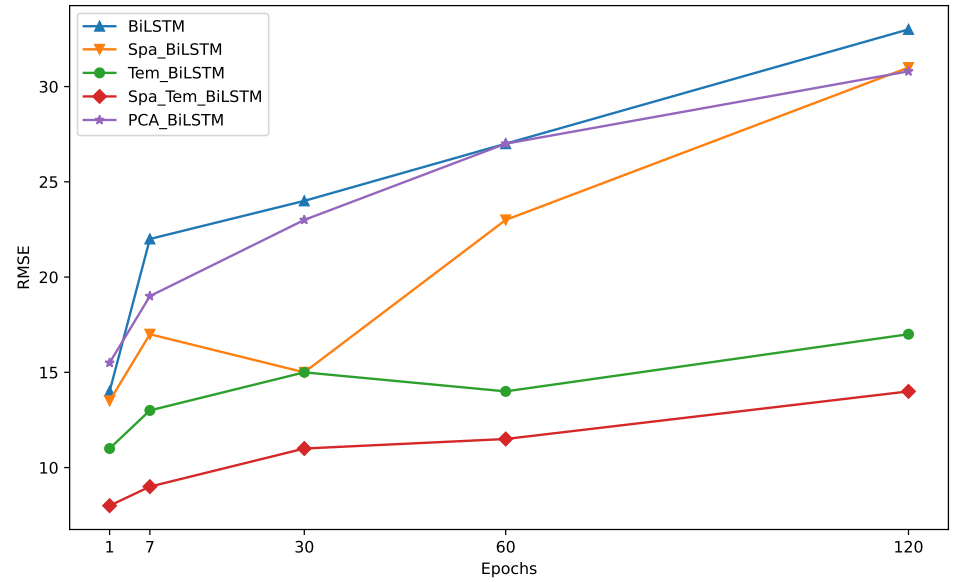
Figure 9. Training set loss function and validation set loss function at T = 60. (**a**) Includes temporal attention mechanisms; (**b**) does not include attention mechanisms.

Table 3. Comparison of the complexities of different models.

Models	Computation/FLOPS	Access Discs/Byte	Compute Density (FLOPS/Byte)
SVR	31.0G	675M	45.9
CNN	22.6G	472M	47.9
GRU	7.72G	211M	36.3
LSTM	3.63G	77.5M	50.1
Bi-LSTM	4.07G	100M	40.7
PCA-Bi-LSTM	1.15G	57.8M	19.9
Spatial-Bi-LSTM	876M	47.6M	18.4
Temporal-Bi-LSTM	545M	36.8M	14.81
Spatial–Temporal-LSTM	68.5M	19M	3.61
Spatial–Temporal-Bi-LSTM	76.9M	29.4M	2.91



(**a**)



(**b**)

Figure 10. Comparison of the performances of BiLSTM models with and without the attention mechanism when the time step is changed. (**a**) Change in MAE for different models; (**b**) change in RMSE for different models.

As can be seen in Figure 10, although the prediction accuracy of all models generally decreases as the time step increases, the increase in the prediction error of Temporal-BiLSTM is smaller and decreases at T = 60 compared to T = 30. This indicates that the model with the temporal attention module has a significant advantage in performing long-term series forecasting. Models without the temporal attention mechanism have substantially higher prediction errors and significantly lower prediction accuracies when the time step increases, emphasizing the important role of the temporal attention mechanism in maintaining the long-term dependencies of the sequences.

In summary, the Bi-LSTM model based on the attention mechanism achieves better performance in stock index sequence prediction by deeply learning the spatiotemporal relationships between different attributes and sequences, providing strong interpretability for this neural network model.

Figures 11 and 12 show the prediction results of each model against the actual stock closing prices at time steps T = 7 and T = 60, respectively. It can be observed from the graphs that the fitting errors of all models increase as the time step increases. This phenomenon suggests that although the models can capture the short-term movements of stock prices better, the accuracy and reliability of the models are still challenged in terms of long-term forecasting.

By analyzing the data in Table 4, it is evident that the STBL model achieves significant improvements in prediction accuracy compared to the Bi-LSTM and LSTM models. Specifically, the mean absolute percentage error (MAPE) of the STBL model decreases by 21.03% and 14.72% relative to the Bi-LSTM and LSTM models, respectively. Additionally, the relative performance improvements are 43.93% and 33.08%. In terms of standard deviation (SD), the STBL model is closer to the mean error compared to the BP and LSTM models, indicating that the STBL model not only has higher accuracy in predicting exchange rate fluctuations but also ensures that most predictions deviate less from the actual positions.

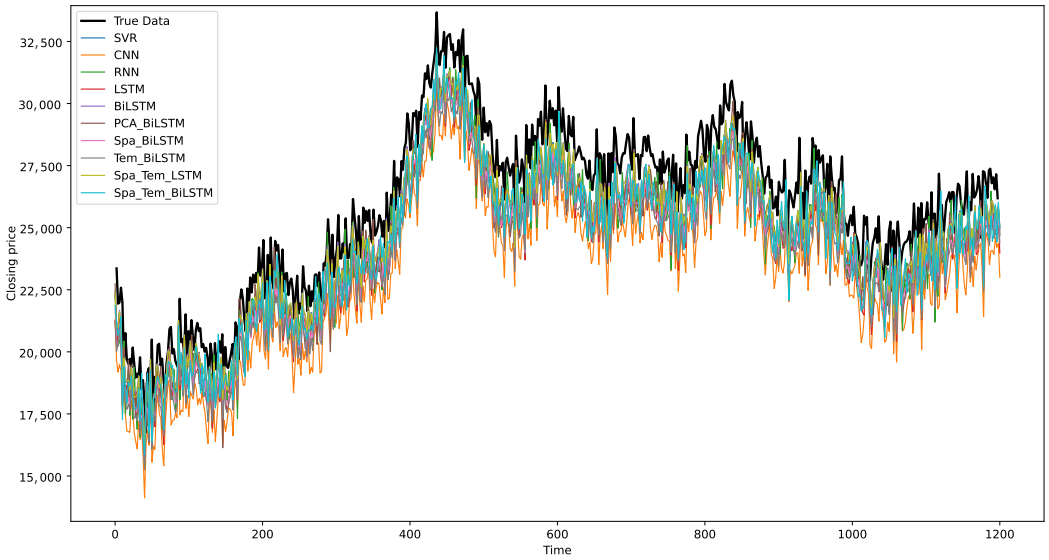


Figure 11. Comparison of the predicted and real prices for all models at T = 7.

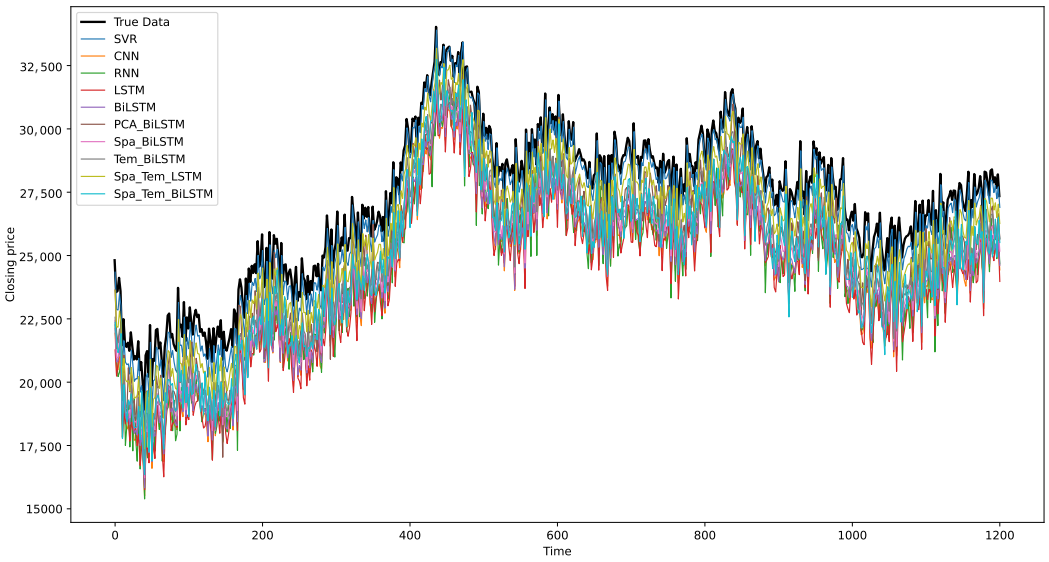


Figure 12. Comparison of predicted and real prices for all models at T = 60.

Table 4. Comparison of the experimental results.

	Bi-LSTM	LSTM	STBL
ME (10^{-4})	−2.213	−1.931	1.342
SD (10^{-4})	6.81	5.513	4.321
MAPE (%)	49.12	40.81	26.09
RMSE (10^{-4})	7.255	6.079	4.068

The Standard & Poor's 500 Index (abbreviated as S&P 500) is a stock market index comprising 500 major publicly traded companies in the United States, covering the period from 31 December 1986, to 31 December 2023. Established by Standard & Poor's in 1957, the index is designed to provide a reliable indicator of the broad fluctuations in the U.S. stock market. Compared to the smaller Dow Jones Industrial Average, the S&P 500 includes a greater number of companies, offering more diversified risk. Initially, the index consisted of 425 industrial stocks, 15 railroad stocks, and 60 utility stocks. As of 1 July 1976, the composition was adjusted to 400 industrial stocks, 20 transportation stocks, 40 utility stocks, and 40 financial stocks, providing more comprehensive coverage of various market sectors.

As can be seen from Table 5, better forecasting results can also be achieved by using the STBL model for the S&P 500 index.

Table 5. Comparison of the forecasting effectiveness of ten models on the S&P 500 for T = 7 and T = 60.

Modeling	T = 7				T = 60			
	RMSE	MAE	MAPE (%)	Dacc	RMSE	MAE	MAPE (%)	Dacc
SVR	34.2993	924.5719	3.4280	0.4988	53.3707	1251.6000	5.9328	0.5024
CNN	34.5639	930.6873	3.3999	0.5008	49.5602	1117.7268	5.5751	0.5074
GRU	31.5862	823.8761	3.0342	0.5216	35.6568	1009.0233	4.5372	0.5074
LSTM	28.9870	729.9283	2.7067	0.5196	34.8646	817.9413	3.2813	0.5114
Bi-LSTM	22.0258	459.1339	1.7321	0.5325	30.0497	572.6089	2.6052	0.5144
PCA-Bi-LSTM	20.5701	427.4808	1.7060	0.5473	29.2325	522.9164	2.1771	0.5315
Spatial-Bi-LSTM	19.1870	396.6393	1.5392	0.5633	25.9113	551.0156	1.7894	0.5385
Temporal-Bi-LSTM	13.9555	370.0519	1.4508	0.5543	15.4410	406.9584	1.5553	0.5516
Spatial–Temporal-LSTM	11.3829	303.7086	1.5362	0.5612	16.2228	330.3886	1.5061	0.5516
Spatial–Temporal-Bi-LSTM	9.9368	259.7306	1.0157	0.5643	13.3657	394.2974	1.2870	0.5546

5. Conclusions and Discussion

5.1. Conclusions

This study investigates the effectiveness of the STBL model in forecasting the closing price of the Hong Kong Hang Seng Stock Index in the short, medium, and long term. The experimental results show that the proposed STBL model achieves optimal results in forecasting at almost all time steps. Specifically, (1) the Bi-LSTM model achieves higher accuracy compared to the standard LSTM at any time step under the same conditions; (2) the Bi-LSTM model, which introduces an attention-based mechanism, outperforms all the baseline models in short-, intermediate-, and long-run stock index forecasts, thanks to its clear and effective representation and learning capability of spatiotemporal relationships; (3) the Spatial-BiLSTM models are more accurate than Bi-LSTM models, incorporating PCA in stock index forecasting and showing the effectiveness of the spatial attention mechanism in extracting data relationships; (4) Temporal-BiLSTM models perform more accurately in long-term stock index forecasting, highlighting the importance of maintaining long-term dependencies in time series forecasting.

To address the issue of overfitting in the BiLSTM model on noisy datasets, this paper has implemented the following strategies:

(1) Data re-cleaning: One possible cause of overfitting is impure data, necessitating a re-cleaning process. We re-cleaned the dataset, removing initial data with zero daily transaction volumes, and selected all transaction data from 31 December 1986 to 31 December 2023.

(2) Increasing training data volume: Another reason for overfitting could be the insufficient amount of data used for training, where the training data represent too small a proportion of the total data.

(3) Using larger datasets: Training the model with larger datasets to enhance the generalization capabilities.

(4) Employing ensemble learning methods: Integrating multiple models, including spatiotemporal attention mechanisms with the BiLSTM model, reduces the risk of overfitting for individual models, effectively mitigating excessive fitting.

5.2. Discussion

The unsupervised stock index prediction method based on the spatiotemporal attention mechanism proposed in this paper focuses on important information and extracts key features more efficiently through the introduction of the attention mechanism, significantly improving time series predictions. Future work will explore further optimization of model parameters, such as investigating the effects of different convolutional kernel sizes on network performance, adjusting the number of convolutional channels to achieve a balance between performance and efficiency, and using different loss functions to fully exploit the network's potential. In addition, subsequent work will consider increasing the input scale of the network to include data pickup tasks with longer time windows. As the input scale increases, the network will face the challenges of more noisy information and a reduced percentage of effective features, which will place higher demands on the feature extraction capabilities of the network. At the same time, the increase in input scale may also lead to a significant increase in the number of network parameters, which will reduce the recognition efficiency. Therefore, in the future, it will be necessary to modify and optimize the network to expand it into a multi-task network and combine it with advanced deep learning concepts to enhance the feature extraction capabilities of the network, to achieve fast and synchronous stock index data acquisition, and to provide more technical support for real-time data acquisition.

Author Contributions: Software, S.M.; formal analysis, B.L.; resources, J.G.; project administration, C.L. and N.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received support from the project "A study on the correlation between urban land transfer prices and the evolution of industrial structure in China" (project no. XJ20BS51). This study is the outcome of the Hubei University of Economics Practical Teaching and Innovation and Entrepreneurship Special Teaching Reform Research Project titled "AI Technology Empowers the Construction of Financial Laboratories in Colleges and Universities", and is supported by the funding from this project.

Data Availability Statement: Data are contained within the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

Autoregressive Integrated Moving Average Model	ARIMA
Long short-term memory	LSTM
Bidirectional long short-term memory	Bi-LSTM
Recursive neural network	RNN
Convolutional neural network	CNN
Deep neural network	DNN
Gated recurrent unit	GRU
Graph attention network	GAT
Attention mechanism	AM
Mean absolute percentage error	MAPE
Root mean square error	RMSE
Mean absolute error	MAE

References

1. Wu, J.L.; Tang, X.R.; Hsu, C.H. A prediction model of stock market trading actions using generative adversarial network and piecewise linear representation approaches. *Soft Comput.* **2023**, *27*, 8209–8222. [CrossRef] [PubMed]
2. Devi, B.U.; Sundar, D.; Alli, P. A study on stock market analysis for stock selection- naive investors' perspective using Data mining Technique. *Int. J. Comput. Appl.* **2011**, *34*, 19–25.
3. Rather, A.M.; Sastry, V.; Agarwal, A. Stock market prediction and Portfolio selection models: A survey. *Opsearch* **2017**, *54*, 558–579. [CrossRef]
4. Hoseinzade, E.; Haratizadeh, S. CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Syst. Appl.* **2019**, *129*, 273–285. [CrossRef]
5. Zulqarnain, M.; Ghazali, R.; Ghouse, M.G.; Hassim, Y.M.M.; Javid, I. Predicting financial prices of stock market using recurrent convolutional neural networks. *Int. J. Intell. Syst. Appl. (IJISA)* **2020**, *12*, 21–32. [CrossRef]
6. Hassanzadeh, H.R.; Sha, Y.; Wang, M.D. DeepDeath: Learning to predict the underlying cause of death with big data. In Proceedings of the 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Jeju, Republic of Korea, 11–15 July 2017; pp. 3373–3376.
7. 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]
8. Shah, J.; Jain, R.; Jolly, V.; Godbole, A. Stock market prediction using bi-directional LSTM. In Proceedings of the 2021 International Conference on Communication information and Computing Technology (ICCICT), Mumbai, India, 25–27 June 2021; pp. 1–5.
9. Chaudhari, S.; Mithal, V.; Polatkan, G.; Ramanath, R. An attentive survey of attention models. *ACM Trans. Intell. Syst. Technol. (TIST)* **2021**, *12*, 1–32. [CrossRef]
10. Zhang, Y.; Tumibay, G.M. Stock Price Prediction Based on the Bi-GRU-Attention Model. *J. Comput. Commun.* **2024**, *12*, 72–85. [CrossRef]
11. Shih, W.; Shire, S.; Chang, Y.C.; Kasari, C. Joint engagement is a potential mechanism leading to increased initiations of joint attention and downstream effects on language: JASPER early intervention for children with ASD. *J. Child Psychol. Psychiatry* **2021**, *62*, 1228–1235. [CrossRef] [PubMed]
12. Tourille, J.; Ferret, O.; Neveol, A.; Tannier, X. Neural architecture for temporal relation extraction: A Bi-LSTM approach for detecting narrative containers. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 224–230.
13. Padmavathi, P.; Harikiran, J. Wireless Capsule Endoscopy Infected Images Detection and Classification Using MobileNetV2-BiLSTM Model. *Int. J. Image Graph.* **2023**, *23*, 2350041. [CrossRef]
14. Huang, Z.; Zhao, S.; Ye, R. Research on Att-NFM Recommendation Algorithm Based on Attention Mechanism. *J. Phys. Conf. Ser.* **2023**, *2504*, 012011. [CrossRef]
15. Li, X.; Wang, X.; Sun, S.; Wang, Y.; Li, S.; Li, D. Predicting the Wildland Fire Spread Using a Mixed-Input CNN Model with Both Channel and Spatial Attention Mechanisms. *Fire Technol.* **2023**, *59*, 2683–2717. [CrossRef]
16. Wen, J.; Zhang, Z.; Fei, L.; Zhang, B.; Xu, Y.; Zhang, Z.; Li, J. A survey on incomplete multiview clustering. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *53*, 1136–1149. [CrossRef]
17. Wen, J.; Liu, C.; Deng, S.; Liu, Y.; Fei, L.; Yan, K.; Xu, Y. Deep double incomplete multi-view multi-label learning with incomplete labels and missing views. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 11396–11408. [CrossRef] [PubMed]
18. Deng, S.; Wen, J.; Liu, C.; Yan, K.; Xu, G.; Xu, Y. Projective incomplete multi-view clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 10539–10551. [CrossRef] [PubMed]
19. Rao, K.V.; Ramana Reddy, B.V. Hm-smf: An efficient strategy optimization using a hybrid machine learning model for stock market prediction. *Int. J. Image Graph.* **2024**, *24*, 2450013. [CrossRef]
20. Hu, Y.; Yang, K. An overview of behavioral finance research in China and abroad—Bibliometric analysis based on Gephi and Cite Space. *Manag. Innov.* **2023**, *1*, 1–8. [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.