


Article

Deep Convolutional Transformer Network for Stock Movement Prediction

Li Xie , Zhengming Chen and Sheng Yu

School of Information Engineering, Provincial Demonstration Software Institute, Shaoguan University, Shaoguan 512005, China

* Correspondence: sgxy_lily@sgu.edu.cn

Abstract: The prediction and modeling of stock price movements have been shown to possess considerable economic significance within the finance sector. Recently, a range of artificial intelligence methodologies, encompassing both traditional machine learning and deep learning approaches, have been introduced for the purpose of forecasting stock price fluctuations, yielding numerous successful outcomes. Nonetheless, the identification of effective features for predicting stock movements is considered a complex challenge, primarily due to the non-linear characteristics, volatility, and inherent noise present in financial data. This study introduces an innovative Deep Convolutional Transformer (DCT) model that amalgamates convolutional neural networks, Transformers, and a multi-head attention mechanism. It features an inception convolutional token embedding architecture alongside separable fully connected layers. Experiments conducted on the NASDAQ, Hang Seng Index (HSI), and Shanghai Stock Exchange Composite (SSEC) employ Mean Absolute Error (MAE), Mean Square Error (MSE), Mean Absolute Percentage Error (MAPE), accuracy, and Matthews Correlation Coefficient (MCC) as evaluation metrics. The findings reveal that the DCT model achieves the highest accuracy of 58.85% on the NASDAQ dataset with a sliding window width of 30 days. In terms of error metrics, it surpasses other models, demonstrating the lowest average prediction error across all datasets for MAE, MSE, and MAPE. Furthermore, the DCT model attains the highest MCC values across all three datasets. These results suggest a promising capability for classifying stock price trends and affirming the DCT model's superiority in predicting closing prices.

Keywords: stock movement prediction; deep learning; transformer; convolutional neural networks; separable fully connected



Citation: Xie, L.; Chen, Z.; Yu, S. Deep Convolutional Transformer Network for Stock Movement Prediction.

Electronics **2024**, *13*, 4225. <https://doi.org/10.3390/electronics13214225>

Academic Editor: M-Tahar Kechadi

Received: 19 August 2024

Revised: 16 October 2024

Accepted: 24 October 2024

Published: 28 October 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

Stock markets are regarded as a fundamental component of the global economic landscape, with fluctuations involving the exchange of billions of dollars [1,2]. Consequently, precise predictions of future trends in financial markets hold significant value across multiple domains. Given these attributes, the stock market has attracted heightened interest from investors, researchers, and institutions alike [3,4].

A variety of stock prediction methodologies have been established, including statistical analysis, machine learning techniques, and deep learning networks [5–7]. Statistical analysis methods were among the first utilized for forecasts in the stock market and have demonstrated considerable success [8]. Nonetheless, these approaches rely on the premise of a linear and stationary time series, which does not align with the non-linear and dynamic characteristics inherent in actual stock market behavior. Consequently, statistical methods exhibit several limitations.

In addition to traditional statistical techniques, machine learning approaches have been introduced for the prediction of stock movement trends, primarily due to their proficiency in handling non-linear and dynamic datasets. Among the commonly utilized machine learning methods for forecasting stock prices are Support Vector Machines (SVMs) [9] and

Artificial Neural Networks (ANNs). However, a significant challenge associated with these methods in practical applications is the tendency to overfit, which arises from their capacity for non-linear mapping and fitting.

Recent advancements in deep learning models, including Convolutional Neural Networks (CNNs) [10], Recurrent Neural Networks (RNNs) [11], and Long Short-Term Memory (LSTM) networks [12], have positioned these architectures as highly effective due to their robust nonlinear generalization capabilities. However, CNNs exhibit limitations in processing time series data, which adversely affects their accuracy in stock market forecasting. In contrast, RNNs are specifically designed to handle time series data within a neural network framework, with LSTM being the most prevalent RNN architecture. The LSTM model incorporates a gating mechanism that emulates human memory, enabling it to autonomously manage the retention of pertinent information while discarding irrelevant data. A comparative analysis of stock movement trend prediction utilizing CNN, RNN, and LSTM methodologies reveals that LSTM outperforms the other approaches, attributable to its capacity for long-term memory retention in stock data series.

Despite the significant advancements achieved by deep learning models such as CNNs and LSTM networks, these approaches exhibit certain limitations [13,14]. In CNNs, each layer performs non-linear mapping; however, as the depth of the network increases, the features undergo increasingly complex non-linear transformations, which may enhance the suitability of deeper feature maps for specific tasks. Nonetheless, CNNs are inherently limited in their ability to process time series data, which poses challenges in effectively leveraging the temporal features of stock data for forecasting purposes. Additionally, the pooling and convolution operations within CNNs primarily concentrate on local information, potentially resulting in the oversight of part-whole relationships. Conversely, while LSTMs are adept at handling time series information, they are prone to the problems of vanishing and exploding gradients during the backpropagation phase.

More recently, attention mechanism models [15,16], particularly Transformer structures [17–19], have achieved satisfactory results in processing sequential information. The attention mechanism can be conceptualized as a representation of human cognitive focus, enabling individuals to prioritize relevant information while disregarding extraneous details. In [20], the attention mechanism was integrated into a CNN-LSTM framework for the purpose of stock prediction. Vaswani et al. [17] proposed an innovative end-to-end deep network called the Transformer, which makes use of the attention mechanism instead of LSTM architectures and has achieved massive success in the field of Natural Language Processing (NLP) [17]. In contrast to LSTM-based approaches, the Transformer architecture facilitates parallel training, thereby enhancing its capacity to capture global temporal information. The Transformer model is widely utilized in various fields, including audio processing [21,22], object detection and recognition [23–25], image classification [26,27], and video processing [28,29], owing to its robust self-attention learning capabilities.

Inspired by the success of the Transformer in processing sequential data in NLP [17], several Transformer architectures have been applied to forecast stock movement trends [3,30]. Zhang et al. [30] proposed the integration of social media text for stock trend prediction, introducing a Transformer encoder-based attention network architecture. Similarly, Muhammad et al. [31] introduced the Transformer model to predict future stock prices. Furthermore, Wang et al. [3] utilized a deep Transformer model to forecast stock market indices, demonstrating predictive performance that significantly surpasses that of traditional methodologies. Despite the potential of the Transformer architecture for stock market prediction, its performance has not shown substantial improvement when compared to similarly sized RNNs and CNNs. Additionally, the direct application of this architecture to stock market prediction presents challenges for two primary reasons: first, these models tend to prioritize global information while overlooking local features that are essential for understanding the intricate dynamics of the stock market; second, the substantial computational and memory demands associated with these models limit their practicality for large-scale stock data analysis.

In response to the identified limitations, we propose the development of a Deep Convolutional Transformer (DCT) model. The innovative nature of our approach is highlighted by several significant features. Firstly, we introduce an inception convolutional token embedding architecture, which is specifically designed to capture both global high-level features and local fine-grained information. This architecture mimics the convolution operation in a manner that is more congruent with the characteristics of stock data, thereby facilitating more effective feature extraction than traditional Transformer architectures. By effectively capturing local features, the model improves its ability to understand short-term fluctuations and patterns within the stock market, which is critical for achieving accurate predictions.

Secondly, we integrate separable fully connected layers into the DCT model. These layers play a vital role in enhancing feature representation by maintaining the temporal order of input features while mapping them to a high-dimensional feature space, thus enriching the overall feature set. Concurrently, they contribute to a reduction in computational complexity when compared to conventional fully connected layers, enabling the model to process large-scale stock data more efficiently and reducing the likelihood of overfitting.

Lastly, our DCT model incorporates a multi-head attention mechanism, which allows the model to concurrently focus on historical information from various representation subspaces. This mechanism is instrumental in capturing the complex interrelationships among different factors in the stock market, including the interactions between various stocks, prevailing market trends, and economic indicators. By considering these multiple dimensions, the model is capable of generating more accurate predictions than existing methodologies that rely exclusively on either global or local features.

The rest of this paper is organized as follows. In Section 2, we discuss the related works on applying machine learning and deep neural networks to stock data. In Section 3, the Transformer-based model is detailed. Section 4 discusses the experiments, including data processing, parameter setting, evaluation metrics, and the performance of the experiments. The conclusions and future work are given in Section 5.

2. Related Works

Stock trend prediction plays a critical role in the field of financial data analysis. Different approaches in the field of stock forecasting can be broadly classified into two principal categories: fundamental analysis and technical analysis [2,7,32]. Fundamental analysis is predicated on the examination of textual data, including macroeconomic indicators, financial news, and earnings reports. In contrast, technical analysis predominantly emphasizes the evaluation of historical stock prices, trading volumes, and other pertinent data to forecast stock movements.

2.1. Fundamental Analysis

Fundamental analysis predominantly depends on company annual reports, financial news, and financial commentary from platforms such as Twitter and Yahoo Finance. In their research, Kohli et al. [33] utilized macroeconomic factors, including historical market data and foreign exchange rates, to predict trends within the Bombay Stock Exchange. Nguyen et al. [34] investigated the relationship between social media sentiment and stock market fluctuations, concluding that sentiment analysis enhances the accuracy of stock forecasts. In [35], financial indicators were used to select the best stocks from the Taiwan stock market. Malandri et al. [36] demonstrated that public sentiment is correlated with financial markets and has an impact on optimal asset allocation. Rajput et al. [37] classified the text information into positive, negative, and neutral categories, and then predicted the stock trend based on the sentiment classification. Sohangir et al. [38] used several typical neural network methods, such as RNN and CNN, to improve the performance of sentiment analysis in classification models for stock prediction. Inspired by the success of attention mechanisms, Shi et al. [20] proposed a hybrid architecture that combines CNN-LSTM

and XGBoost with an attention mechanism. The hybrid model focuses on critical textual information for stock prediction. Ali et al. [39] designed a model for opinion mining to extract investors' opinions using a part-of-speech graph model.

2.2. Technical Analysis

Technical analysis-based methods posit that all information pertaining to the stock market will be reflected in price fluctuations. It is sufficient to use historical stock data directly to predict future stock trends. Specifically, assuming there is a time series $x_{t-1}, x_{t-2}, \dots, x_{t-m}$, the objective is to forecast future values $y_{t-1}, y_{t-2}, \dots, y_{t-m}$ using the gathered historical data [40]. There are numerous methods currently employed for stock market forecasting, primarily based on machine learning techniques, which can be divided into the following more specific categories of methods: traditional machine learning and deep learning [41,42].

In the field of stock forecasting, many conventional machine learning models have been widely utilized [43–45]. Traditional machine learning methods include support vector machines [43,46], decision trees [47], Naive Bayes [48,49], and so on. However, the inability to extract deep features from stock data to bridge the gap in predictive ability between machines and humans is one of the primary limitations of traditional machine learning methods. Beyond that, the performance of machine learning methods largely depends on the quality of features. Unsuitable features are likely to result in poor model performance.

Recently, there has been active research on deep learning-based methods for stock prediction, which have achieved remarkable performance [45,50–52]. The primary distinction between traditional machine learning methods and deep learning is that the network structure can independently learn effective features according to the target task. Selvin et al. [53] proposed a hybrid deep model consisting of three modules: RNN, LSTM, and CNN, to predict stock prices. In [54], RNN architecture is used to forecast stock market trends. Using real data from ten Nikkei companies as examples, the validity of the method is verified. In [55], the authors used gold price, gold volatility index, crude oil price, and the crude oil price volatility index. This combination has demonstrated strong performance in predicting future stock price trends. Zhao et al. [50] proposed a novel deep model that consists of an emotion-enhanced convolutional neural network, denoising autoencoder models, and LSTM for stock price movement forecasting. Long et al. [56] introduced a multi-filter neural network to extract effective features from financial time series samples. In [41], Lee et al. developed an end-to-end architecture for stock forecasting, which includes two feature extractor networks designed to learn high-level features from time series stock data. Qin et al. [57] proposed a temporal attention mechanism model for predicting stock prices.

Graph Neural Networks (GNNs) fundamentally operate on the principle that the state of any given node depends on the states of its neighboring nodes. This assumption allows GNNs to effectively capture the spatial interdependencies among nodes while preserving the temporal dynamics of the data. GNNs have proven effective in capturing spatial dependencies across a wide range of domains. However, their application in stock market prediction has been somewhat limited compared to other areas. The proposed MG-Conv model integrates one-dimensional and multi-graph convolutional networks within a three-layer framework [58]. The one-dimensional convolutional layer normalizes the data and extracts features, while the graph convolutional layer constructs both static and dynamic graphs to perform convolution. Performance evaluations conducted on 42 Chinese indices demonstrate that the MG-Conv model reduces the average prediction error by 5.11% compared to alternative methods. However, overfitting remains a concern, and future research may focus on enhancing generalization and improving the fusion of index trends. Wang et al. [59] tackle the challenge of stock index prediction by addressing the high noise and dynamic nature of stock data. The inadequacy of existing methods to capture local spatial-temporal correlations prompted the development of the LoGCN model, which integrates graph construction, convolution, and pooling mechanisms. Experimental results on 42 Chinese stock indices demonstrate that the LoGCN model outperforms traditional methods such as MLP and LSTM across various evaluation metrics, including regression,

classification, and market back-testing. Future research directions will focus on enhancing classification accuracy and devising innovative investment strategies. Ma et al. [60] aim to address the challenges of stock price prediction. Existing methods often rely on predetermined graphs or overlook certain correlations. The proposed VGC-GAN model generates multiple correlation graphs from historical data, utilizing techniques such as Pearson correlation, Spearman rank correlation, and FastDTW. Additionally, it employs a Variational Mode Decomposition (VMD) algorithm with optimized parameters obtained through a Genetic Algorithm (GA) to manage data non-stationarity. The model's framework includes a generative adversarial module comprising a generator (Multi-GCN and GRU) and a discriminator, along with components for relationship graph generation and stock sequence decomposition. Experiments conducted on ETF, SSE, and DJIA datasets demonstrate that VGC-GAN outperforms methods such as GRU and GCGRU in terms of prediction performance and computational efficiency. Liu et al. [61] propose a novel model, ECHO-GL, for predicting stock movements. Existing models often exhibit insufficient or static stock relationships. ECHO-GL addresses these limitations by leveraging earnings calls. It constructs a heterogeneous graph (E-Graph) with various types of nodes and edges, employing mechanisms such as time assignment and sliding windows to ensure its dynamism. Qian et al. [62] focuses on stock investment prediction. The stock market is complex, with prices affected by various factors. Traditional sequential and graph-based methods have limitations in capturing multifaceted and temporal influences. The proposed MDGNN framework uses a discrete dynamic graph to capture stock relations and their evolution. It includes an intra-day graph snapshot with a multi-relational graph construction and a hierarchical graph embedding layer. The inter-day temporal extraction layer uses a Transformer structure to handle temporal evolution. The prediction layer estimates the probability of a stock's positive return.

The Transformer is another deep learning method used in stock market prediction, following CNN and LSTM. Its ability to process time series information has been proven in many other domains as well. Li et al. [63] proposed an LSTM and attention-based model, which consists of a multi-input LSTM and an attention layer, followed by several ReLU layers to obtain the final prediction. Results show significant improvements over several LSTM or CNN-based methods on the CSI-300 index dataset with a time horizon of 4 years. Li et al. [64] designed a hybrid deep learning model consisting of a Transformer, LSTM, Gate Recurrent Unit (GRU), and a high-frequency data adaptive decomposition architecture. The model is named Frequency Decomposition and includes a GRU Transformer (FDG-trans). Wang et al. [3] utilize the Deep Transformer (DT) model to predict stock market indices, and the prediction performance was significantly better than that of other classic methods. Muhammad et al. [31] developed a Transformer-based network with two input layers, three Transformer layers, one pooling layer, followed by two dropout layers, and two dense layers to make predictions on the Dhaka Stock Exchange, including daily, weekly, and monthly share prices. The results are promising. Sridhar and Sanagavarapu [65] combine the Transformer with CNN to predict stock trend movement. The network structure comprises two multi-head layers, one Transformer, and two convolutional layers. The experiment showed that the model outperforms several models based on CNN or RNN on the S&P 500 dataset. Financial stock prices are complex to predict due to a multitude of influencing factors. Current methodologies, such as Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTM), Gated Recurrent Units (GRUs), and Transformers, each have their own limitations. The HPMG-Transformer model [66] integrates the Hodrick–Prescott (HP) filter with a multi-scale Gaussian transformer. The HP filter decomposes stock time series into long-term and short-term fluctuations, while the multi-scale Gaussian transformer enhances the extraction of local features. Additionally, a multi-scale Gaussian prior is incorporated into the self-attention mechanism to improve the extraction of local contextual information. Li et al. [67] focus on stock price forecasting, a challenging task due to market volatility. Existing methods have limitations in modeling stock correlations and accounting for market variations. The proposed MASTER

model consists of five steps: market-guided gating, intra-stock aggregation, inter-stock aggregation, temporal aggregation, and prediction. It models both momentary and cross-time stock correlations and utilizes market information for automatic feature selection. MASTER offers a novel perspective on stock correlation modeling and effectively leverages market information for stock price prediction. Future research could explore improved methods for stock correlation mining and additional applications of market information. The efficient market hypothesis has inspired various methods for stock prediction; however, challenges remain in effectively integrating numerical and textual data. To address this issue, Zhang et al. [68] proposed the CoATSMP model, which includes text and price feature extraction, a Transformer-based soft fusion method, and joint feature processing utilizing LSTM and a temporal attention mechanism. This model employs datasets from diverse sources and evaluates performance using metrics such as accuracy (ACC), Matthews correlation coefficient (MCC), F1 score, and area under the curve (AUC). Li et al. [69] propose an innovative approach to enhancing stock price prediction by integrating Generative Adversarial Networks (GANs) with Transformer-based attention mechanisms. This study underscores the importance of accurate stock price forecasting in the financial sector, enabling investors to make informed decisions. The authors acknowledge the limitations of traditional statistical methods and position machine learning and deep learning as promising alternatives. The methodology includes data preprocessing to address imbalances in the distribution of news data and to enrich the dataset with technical indicators. The model architecture features a Variational Autoencoder (VAE) for feature extraction, GANs for generating synthetic data, and a Transformer-based attention mechanism for focused data analysis.

Compared to the other deep learning-based methodologies previously discussed, FDG-trans [64], Transformer Network (TN) [31], and DT [3] are most closely related to the Deep Convolutional Transformer (DCT). In [64], FDG-trans is built by integrating GRU, LSTM, and multi-head attention Transformers. Both TN and DT are stock prediction models based on the standard Transformer architecture. In contrast, our DCT model incorporates the separable fully connected layer and inception convolutional token embedding into the standard Transformer structure to enhance information richness and learn fine-grained features.

Due to the robust capabilities of deep learning models in autonomously extracting features from unprocessed data, contemporary methodologies for forecasting stock price trends favor these models. Table 1 presents a comprehensive overview of research papers that utilize deep learning, categorizing them according to the types of features and model architectures employed. Nevertheless, there remains significant potential for enhancement in existing deep learning architectures aimed at predicting stock movements based on textual data and historical stock information. Forecasting stock price trends necessitates the analysis of time-series data, which is inherently reliant on stock-related information over time. Consequently, the primary challenge is to enhance the accuracy of stock movement analysis and prediction utilizing financial data, such as historical stock prices, while effectively addressing the issue of temporal dependence. The impetus for our research is to bridge this gap and investigate the efficacy of stock time series engineering through the application of the Transformer architecture.

Table 1. Analysis based on feature types and model architectures.

Reference	Year	Feature Types	Model Architectures
[62]	2024	Basic features	Graph Neural Network
[60]	2024	Basic features, Technical indicators	Graph Neural Network
[30]	2022	Basic features, Technical indicators	Transformer-based
[3]	2022	Basic features	Transformer-based
[31]	2022	Basic features	Transformer-based
[64]	2022	Basic features	GRU-Transformer
[65]	2021	Basic features	Multi-Head Self-Attention

Table 1. Cont.

Reference	Year	Feature Types	Model Architectures
[41]	2020	Basic features	CNN-LSTM
[42]	2022	Basic features	CNN-LSTM
[50]	2023	Basic features, Technical indicators	CNN, Autoencoder, LSTM
[51]	2018	Basic features	CNN, LSTM, RNN
[52]	2019	Basic features, Technical indicators	CNN
[53]	2017	Basic features	CNN, LSTM, RNN
[54]	2014	Technical indicators	RNN-RBM
[55]	2021	Technical indicators	CNN
[56]	2019	Basic features	CNN-RNN
[57]	2017	Technical indicators	RNN
[63]	2018	Basic features, Technical indicators	LSTM

3. Methodology

In this section, we first formalize the stock trend forecasting problem. Then, we introduce the DCT architecture.

3.1. Problem Statement and Formulation

Let y assume the closing stock price for a given day, where \hat{y} denotes the prediction value. The variables x_1, x_2, \dots, x_T correspond to the input data features at various time intervals. The objective is to minimize the function of $V(y, \hat{y})$.

$$\hat{y} = f(\theta, x_1, x_2, \dots, x_T) \quad (1)$$

$x_t = x_{t1}, x_{t2}, \dots, x_{tN}$ is a feature vector at time t , and θ denotes the parameter of the function $f(\cdot)$. $V(\cdot, \cdot)$ is a similarity function used to measure the proximity of the estimated value to the true value.

Furthermore, we aim to forecast the stock trend of s on trading day t using historical stock transaction data from a lag period of $[d - \Delta d, d - 1]$, where Δd represents a fixed lag size in this paper. The objective is to assess the movement direction, and the labeling methodology is described as follows:

$$target = \begin{cases} 1, & Close_{t+1} \geq Close_t \\ 0, & else \end{cases} \quad (2)$$

where 1 indicates a rise and 0 indicates a fall, and $Close_t$ represents the closing price of the stock on day t .

3.2. The Proposed Model: DCT

Convolutional neural networks were first introduced in [70] by LeCun et al. for processing images. At present, methodologies based on CNNs have attained leading performance levels in the domains of image and video processing.

This study utilizes a Transformer-based architecture for stock movement prediction, diverging from the conventional convolutional neural network model. In this section, we will present a comprehensive introduction to our DCT architecture. Figure 1 illustrates the framework of the DCT model, which consists of three inception convolution layers, two separable fully connected layers, and Transformer components. The ViT-Base model, as described in [71], has been selected for the Transformer structure, with specific parameter configurations detailed in Table 2.

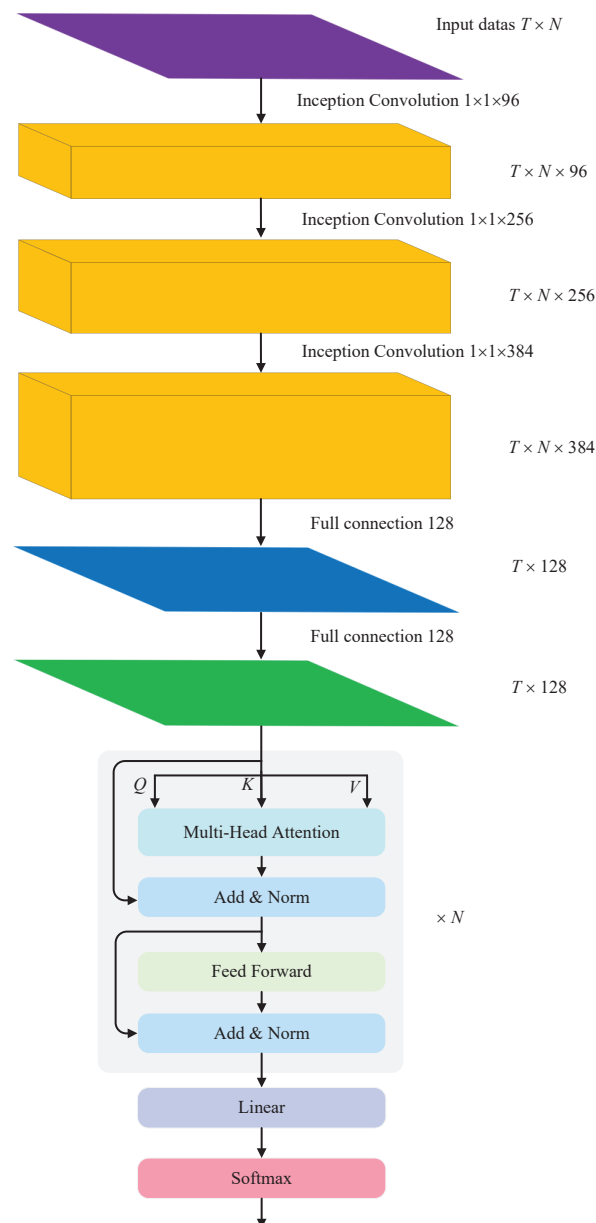


Figure 1. The overall architecture of the DCT method. Here, T is the lookback trading day, and N represents the number of features of the stock index.

Table 2. Details of Transformer model.

Model	Layers	Hidden Size D	MLP Size	Heads
ViT-Base	12	768	3072	8

3.2.1. Inception Convolutional Token Embedding

The inception convolution operation is designed to capture local features by converting low-level data into higher-order semantic representations through a multi-level hierarchical framework. The 1×1 convolution kernel is characterized by having a single parameter for each channel, without incorporating any adjacent pixels from the input. Consequently, the application of multiple 1×1 filters enables the modification of the number of summaries generated from the input feature map, thereby facilitating an increase in the depth of the feature maps as required.

In a formal context, when provided with a one-dimensional stock feature vector $\mathbf{x}_t \in \mathbb{R}^{1 \times N}$ as input to the initial inception convolutional layer, our objective is to develop a function $f(\cdot)$ that transforms \mathbf{x}_t into new tokens represented as $f(\mathbf{x}_t)$, characterized by a channel size of C_i . We refer to this embedding method as inception convolutional token embedding.

Specifically, the output generated by the inception convolutional layer can be represented as:

$$x_j^l = \phi \left(\sum_{i \in M_j} x_i^{l-1} \times w_{ij}^l + b_j^l \right) \quad (3)$$

where M_j is a set of input maps, w_{ij}^l is the convolutional kernel of the l th layer, and b_j^l is bias. Furthermore, the activation function ϕ can use a tanh function or a sigmoid function. More specifically, the convolution kernel size is 1×1 , and the input feature dimension of the first inception convolutional layer is 1×5 in the DCT model.

The primary inception convolutional layer is designed to extract a range of low-level features from the input stock data. In contrast, the subsequent two inception convolutional layers are adept at capturing more intricate high-level semantic features. In summary, the inception convolutional token embedding layers facilitate the construction of a convolutional projection of a series of input feature maps, enabling the adjustment of the number of feature maps at each stage by modifying the parameters associated with the inception convolution operation.

3.2.2. Separable Fully Connected Layer and Softmax

As illustrated in Figure 2, the output of the first fully connected layer is the input of the last convolutional layer. To preserve the temporal order of the input features, the fully-connected operation is limited to the temporal channels only, a concept we refer to as separable full connection. Specifically, the input feature mapping $F \in \mathbb{R}^{T \times N \times 384}$ is initially decomposed into T channels based on the temporal axis. Subsequently, two fully connected operations are performed within each individual channel, and finally, the fully connected feature channels are merged into $A \in \mathbb{R}^{T \times 128}$.

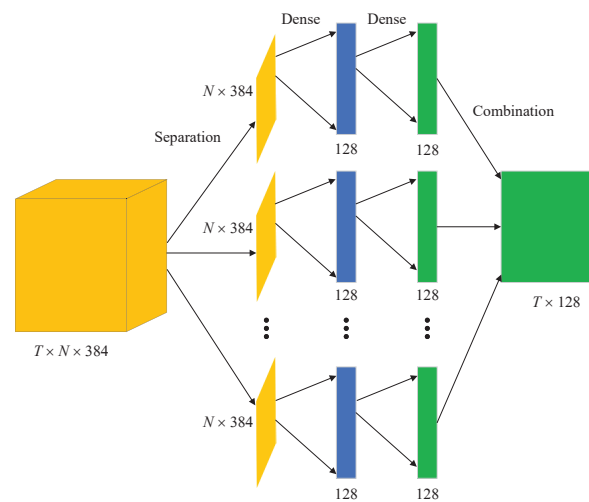


Figure 2. Separable fully connected layer.

The Softmax function is usually used to conduct the class prediction. Specifically, the Softmax function is written by

$$\text{softmax}(z)_j = e^{z_j} / \sum_{k=1}^K e^{z_k} \quad j = 1, \dots, K \quad (4)$$

where K is the dimension of the z feature vector. In this approach, inception convolutional token embedding and fully connected operations are utilized to extract discriminative information, followed by Transformer for further analysis. This structure offers a good balance between acquiring more information and minimizing overfitting.

3.2.3. Self-Attention

In recent years, attention mechanisms have become the most exciting novel approach in deep learning [72–74]. The attention mechanism architecture focuses limited attention on crucial areas to conserve computing resources and rapidly learn the most useful features. In early research, attention mechanisms were often combined with CNN and RNN architectures [75–77]. Vaswani et al. [17] first demonstrated that deep learning architectures could move beyond CNN and RNN architectures by exclusively using a self-attention mechanism to propose the Transformer model.

According to the self-attention mechanism in [17], Q , K , and V represent the query, key, and value matrices, respectively. The self-attention mechanism can be formulated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (5)$$

where $\sqrt{d_k}$ is the scaling factor, $Q \in \mathbb{R}^{L \times d}$, $K \in \mathbb{R}^{L \times d}$, $V \in \mathbb{R}^{L \times d}$, L is the number of days and d is the dimension of the features. Figure 3 shows the formation process of the Q , K , and V matrix, where A is the output of the separable fully connected layer.

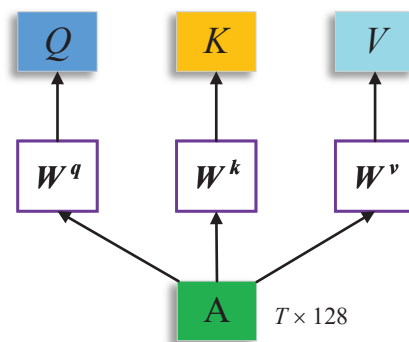


Figure 3. The architecture of Q , K , and V matrices are constructed.

3.2.4. Multi-Head Attention

Self-attention can focus on useful information, but a single self-attention mechanism may struggle to capture sufficient information to enhance the performance of stock trend forecasting. Instead of performing a single attention function, multi-head attention allows the network to collectively focus on information from different representation subspaces at different positions. The multi-head attention function is implemented by

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^O \quad (6)$$

where $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ is the weight matrix, and $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$. In this work, we use $h = 12$ parallel heads. For each of these, we employ $d_k = d_v = \frac{d_{\text{model}}}{h} = 8$. In terms of total computational cost, it is similar to single-head attention with full dimensionality because of the reduced dimension of each head. Figure 4 illustrates the architecture of the multi-head attention mechanism.

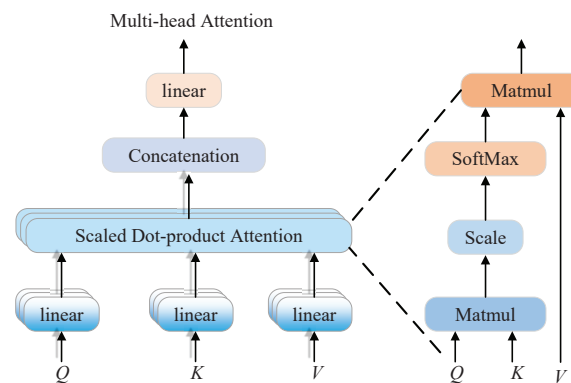


Figure 4. The architecture of the multi-head attention mechanism.

4. Experiments

In this section, we experimentally demonstrate the predictive capability of the DCT model. In Section 4.1, we introduce the data processing method used in the experiments. The parameter settings and evaluation criteria are introduced in Sections 4.2 and 4.3, respectively. In Section 4.4, the predicted performance of the DCT model is evaluated.

4.1. Data Processing

The present study examines three prominent global stock indices: the Hang Seng Index (HSI), the Shanghai Stock Exchange Composite Index (SSEC), and the NASDAQ. These indices are representative of significant financial markets across various global regions, thereby providing a comprehensive assessment of the model's performance under diverse economic conditions. The HSI encapsulates the Hong Kong market, influenced by both Western and Asian economic factors. In contrast, the SSEC reflects the rapid growth of China's market, offering insights into one of the world's most dynamic economies. The NASDAQ, recognized for its concentration on technology-oriented listings, provides a unique perspective on the U.S. market, a fundamental component of global finance. By integrating these indices, this research seeks to evaluate the robustness and generalizability of the predictive model across different market dynamics, thereby ensuring that the findings possess broader implications for stock market forecasting. The dataset spans from 1 January 2013 to 31 August 2024 and encompasses five fundamental stock market features: opening price, closing price, lowest price, highest price, and trading volume. The study employs a look-back methodology for forecasting closing prices, utilizing a look-back trading period of $T = 30$ days. Consequently, the input data for the Discrete Cosine Transform (DCT) model is structured as a two-dimensional matrix of size $T \times N$, where N denotes the number of features associated with the stock index, and T signifies the look-back trading days. The initial 80% of the data, arranged chronologically, is allocated for training the network parameters, while the remaining 20% serves as the test set.

In order to mitigate the volatility of the features and enhance the robustness of the model, we implement Equation (7) to standardize the original training and testing datasets.

$$\hat{x}_t = \frac{x_t - \mu}{\sigma} \quad (7)$$

where \hat{x}_t is the normalized features at time t , x_t is the original features at time t , and μ and σ denote the mean value and standard deviation of the original features in the training set.

In each experimental forecast, we employ the historical stock transaction data from the preceding period, denoted as T , to project the closing price for the subsequent trading day, referred to as $T + 1$. This process involves generating features and labels from the observed time series data using a sliding window methodology.

4.2. Parameters Setting

In the conducted experiments, a mini-batch training approach was employed with a batch size of 16. The loss function utilized for this study is the mean squared error, which evaluates the discrepancy between actual and predicted values. The model was optimized using the Adam optimizer, initialized with a learning rate of 0.01. Furthermore, a decay rate of 0.9 was established, with a decay step set at 100. To mitigate the risk of overfitting, the dropout technique was implemented across each sub-layer, with a dropout rate of 0.2. The initial weights and biases of the neural networks were drawn from a Gaussian distribution characterized by a mean of 0 and a variance of 10^{-2} . For the input stock data, the width of the sliding window was configured to $T = 30$.

The DCT model was developed utilizing the methodology presented in Algorithm 1. This approach employs the stochastic gradient descent technique, focusing on the minimization of logarithmic loss as the primary update mechanism. Algorithm 1 delineates the training protocol for the DCT model, which aims to achieve a high degree of accuracy in predicting stock movements. The algorithm incorporates the initial parameters of the network and systematically refines them through iterative updates informed by the training data. The following is a comprehensive step-by-step elucidation:

Inputs: Let x_k denote a sequence of stock data with a total length of T , where k serves as an index for the dataset. The symbol θ signifies the initial parameters of the network model, encompassing both weights and biases utilized within the DCT model.

Outputs: The notation $\hat{\theta}$ indicates the parameters that have been trained upon the conclusion of the training process.

Algorithm Steps:

(1) **Epoch Loop:** The outer loop iterates through a predetermined number of epochs, with each epoch representing a complete traversal of the entire training dataset, thereby enabling the model to assimilate knowledge from all available data.

(2) **Data Loop:** Within the epoch loop, an inner loop processes each individual data point x_k from the dataset. This nested structure guarantees that the model updates its parameters for every data point encountered.

(3) **Length of Data:** For each data point, the algorithm first determines the length, which corresponds to the time steps or the sequence length of the stock data.

(4) **DCT Application $P(\theta)$:** The DCT model is applied to the stock data x_k using the current parameters θ . This process involves passing the data through the inception convolutional layers, separable fully connected layers, and the Transformer architecture to generate predictions.

(5) **Loss Calculation:** The loss function is calculated by aggregating the logarithmic errors over all time steps t from 1 to $l - 1$. The logarithmic loss is employed in this context as it is a prevalent choice for regression tasks, effectively penalizing larger errors more severely. This step is critical as it quantifies the discrepancy between predicted and actual values, thereby directing the model towards enhanced predictive accuracy.

(6) **Parameter Update:** The model parameters θ are updated using gradient descent. The gradients of the loss with respect to the parameters are computed, and the parameters are adjusted in the opposite direction of the gradient, scaled by the learning rate η . This iterative process fine-tunes the model parameters to minimize the loss function, thereby improving the model's predictive performance.

(7) **End of Training:** Upon the completion of all epochs, the algorithm yields the trained parameters $\hat{\theta}$, which represent the optimized weights and biases of the DCT model following the training phase.

Algorithm 1 DCT training, $\hat{\theta} \leftarrow \text{Training}(x_{1:K_{data}}, \theta)$.

Input: $\{x_k\}_{k=1}^{K_{data}}$, x_k is a stock data sequence of length T .
Input: θ , initial network model parameters.
Output: $\hat{\theta}$, the trained parameters.
Hyperparameters: $M_{epochs} \in \mathbb{R}, \eta \in (0, \infty)$
For $i = 1, 2, \dots, M_{epochs}$ **do**
 For $j = 1, 2, \dots, K_{data}$ **do**
 $l \leftarrow \text{length } x_j$
 $P(\theta) \leftarrow \text{DCT}(x_j|\theta)$
 $\text{loss}(\theta) = -\sum_{t=1}^{l-1} \log P(\theta)[x_j[t+1], t]$
 $\theta \leftarrow \theta - \eta \cdot \nabla \text{loss}(\theta)$
 end
end
 $\hat{\theta} = \theta$

4.3. Evaluation Metrics

In the conducted experiments, the performance of the DCT method was assessed utilizing prediction accuracy, the Matthews correlation coefficient (MCC), and three distinct loss metrics.

The Matthews correlation coefficient (MCC) is a metric used to evaluate the correlation between actual and predicted classifications, and it is particularly effective in addressing issues related to data imbalance. The definitions of accuracy and MCC are provided as follows:

$$\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (9)$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. The range of MCC is $[-1, 1]$. The higher the correlation between the true value and the predicted value, the better the prediction.

Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE) are three metrics employed for the assessment of loss error values. MAE and MSE quantify the absolute discrepancies in predicted values, whereas MAPE reflects the relative discrepancies. Lower values of these three metrics signify superior performance, indicating a more robust alignment of the model with the observed data.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i| \quad (10)$$

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad (11)$$

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{\hat{y}_i - y_i}{y_i} \right| \times 100\% \quad (12)$$

where N is the number of the sample data, \hat{y}_i is the predicted value, and y_i is the true value.

4.4. Performance of the Experiments

We evaluated the accuracy of different widths of the sliding windows (T ranging from 5 to 45 in increments of 5) for the DCT model. Table 3 shows the accuracy on the NASDAQ. The DCT model achieved the highest accuracy 58.85% with $T = 30$. As the width of the sliding window increases, the accuracy decreases. The potential reason for this phenomenon could be the excessive inclusion of noise information due to inputting longer

sequences of stock data into the model, which adversely affects its recognition performance. Another potential issue is that the DCT model may struggle to process extended timing information, which could result in an inability to improve recognition performance even with longer input. Based on the model results, we will use a sliding window width of $T = 30$ for the remainder of this paper.

Table 3. Results of the NASDAQ dataset using various widths of the sliding window.

	$T = 5$	$T = 10$	$T = 15$	$T = 20$	$T = 25$	$T = 30$	$T = 35$	$T = 40$	$T = 45$
Accuracy (%)	57.63	58.11	58.21	58.32	58.66	58.85	58.72	58.32	57.95

The cross-validation results presented in Table 4 offer detailed insights into the predictive performance of the DCT, LSTM, and Transformer models across five folds for the SSEC index, utilizing the Mean Absolute Error (MAE) metric. The parameters of the comparison algorithms are meticulously configured to align with the specifications outlined in the original literature. The DCT model exhibits consistent performance, with an average MAE of 0.0136, which is significantly lower than that of the LSTM and Transformer models. Additionally, the low standard deviation of the DCT model's MAE across folds underscores its robustness and reliability across various subsets of the data.

Table 4. The MAE of cross-validation results for the SSEC index.

Model	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
DCT	0.0138	0.0137	0.0137	0.0135	0.0135	0.0136
LSTM [63]	0.0168	0.0163	0.0167	0.0161	0.0164	0.0165
Transformer [31]	0.0145	0.0142	0.0145	0.0147	0.0143	0.0144

Table 5 presents three indicators from the testing process, revealing that our DCT method surpasses other models in performance. In terms of absolute error (MAE and MSE) and relative error (MAPE), DCT achieves the smallest average prediction error compared to all other models across all datasets. Our DCT architecture has two advantages over LSTM [63] and the Transformer [31] and has demonstrated performance superiority. Firstly, it employs inception convolutional token embedding to extract valuable low-level features for stock prediction. Second, it utilizes separable fully connected and multi-head attention methods to extract sophisticated features from basic features. In summary, these four indicators demonstrate that the proposed DCT method yields good prediction results.

Table 5. The means of three indicators for three datasets in the test process.

Index	Models	MAE	MSE	MAPE
SSEC	LSTM [63]	0.0165	0.0057	2.5627
	Transformer [31]	0.0144	0.0043	2.3685
	DCT	0.0136	0.0039	1.4121
NASDAQ	LSTM [63]	0.0362	0.0053	3.234
	Transformer [31]	0.0228	0.0031	3.0963
	DCT	0.0188	0.0026	2.5627
HSI	LSTM [63]	0.0669	0.0065	7.1267
	Transformer [31]	0.0632	0.0062	7.0159
	DCT	0.0602	0.0058	6.8165

In Table 6, DCT achieves the highest accuracy across all models in the three datasets. Stock movement prediction is a challenging task, and even small improvements can lead to significant profits. Generally, a result of more than 58.8% is considered satisfactory for stock movement prediction. Compared to the Transformer, the DCT shows improvements of more than 1.99%, 2.19%, and 2.38% on the three datasets, respectively. In fact, the prediction

accuracy is about 2.0% higher on average compared to other architectures. The possible reason is that designing the structure of the inception convolution token embedding with separable fully connected layers is crucial for stock movement prediction. In all three datasets, the DCT model also achieved the maximum MCC values. The superior performance demonstrates the discriminative ability of the proposed DCT and proves that the deep inception convolutional Transformer architecture is beneficial for stock prediction. The results for accuracy and MCC indicate that the DCT model consistently shows better improvement, demonstrating the robustness of our method.

Table 6. Comparison with previous state-of-the-art methods in accuracy and MCC.

Models	SSEC		NASDAQ		HSI	
	Accuracy (%)	MCC	Accuracy (%)	MCC	Accuracy (%)	MCC
LSTM [63]	56.46	0.1175	56.46	0.1166	56.26	0.1169
Transformer [31]	57.06	0.1285	56.66	0.1207	56.46	0.1203
SA-DLSTM [50]	56.33	0.1206	55.46	0.1136	56.32	0.1183
TEANet [30]	55.59	0.1326	57.02	0.1238	55.37	0.1265
FDG-Trans [64]	57.56	0.1536	56.84	0.1354	57.18	0.1536
DCT	59.05	0.1656	58.85	0.1596	58.85	0.1638

The proposed DCT method and two baseline methods (LSTM and Transformer) are compared. In Figures 5–7, stock closing prices are predicted for HSI, NASDAQ, and SSEC. In these figures, “Ground Truth” refers to the actual values of the stock index, while “Predictions” denotes the forecasted stock index value. The results indicate that the actual curve closely aligns with the predicted curve across various stock indexes, and the peaks of the two curves are identical. Our proposed method demonstrates the ability to predict stock closing prices more accurately than existing methods. In Figures 5–7, compared to the Transformer, DCT is closer to the actual value. This suggests that the inception convolutional token embedding and separable full connection can learn more effective features from the limited original stock data and improve the prediction accuracy.

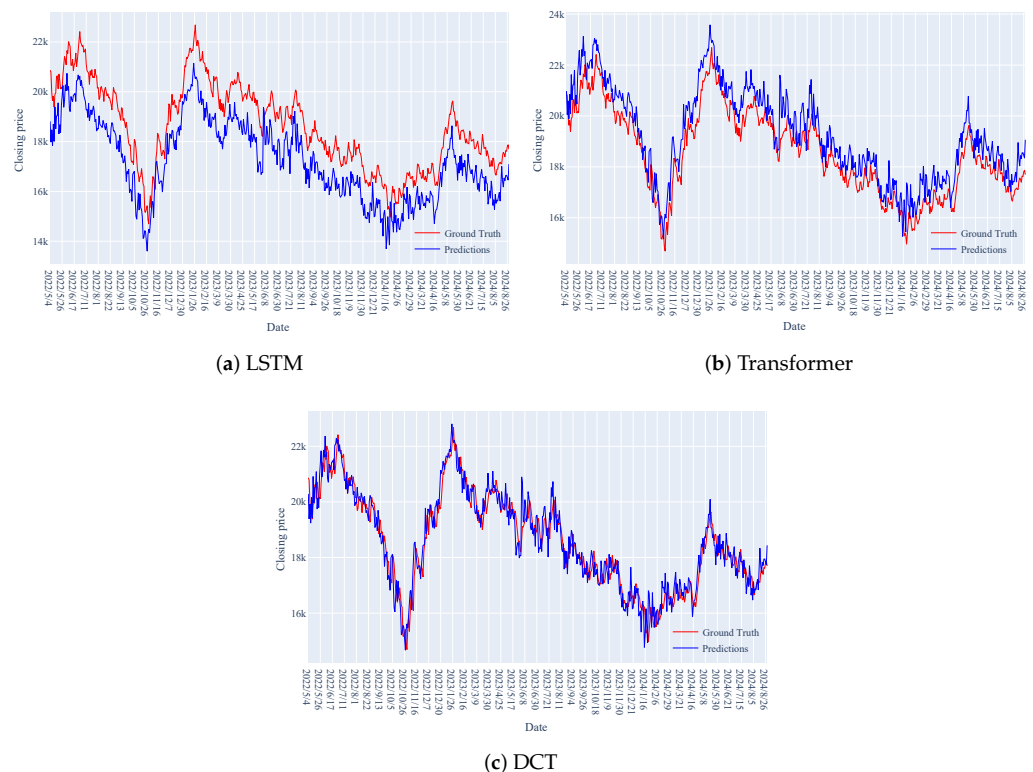


Figure 5. Prediction of stock close prices of HSI for various stock indexes.

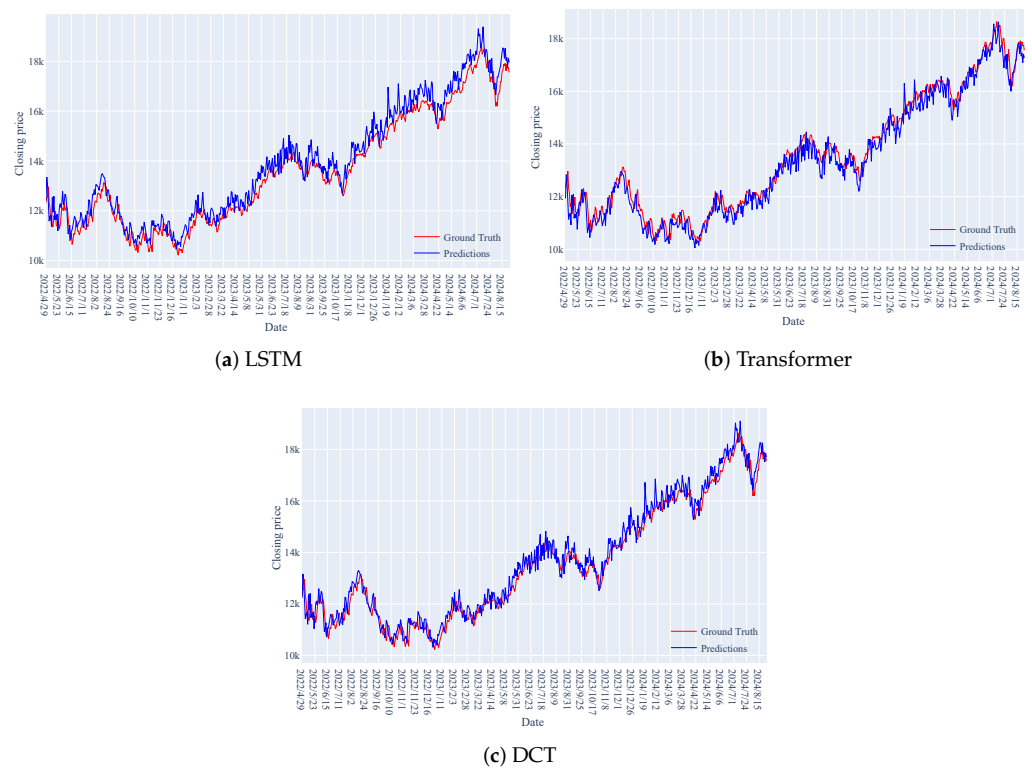


Figure 6. Prediction of stock close prices of NASDAQ for various stock indexes.

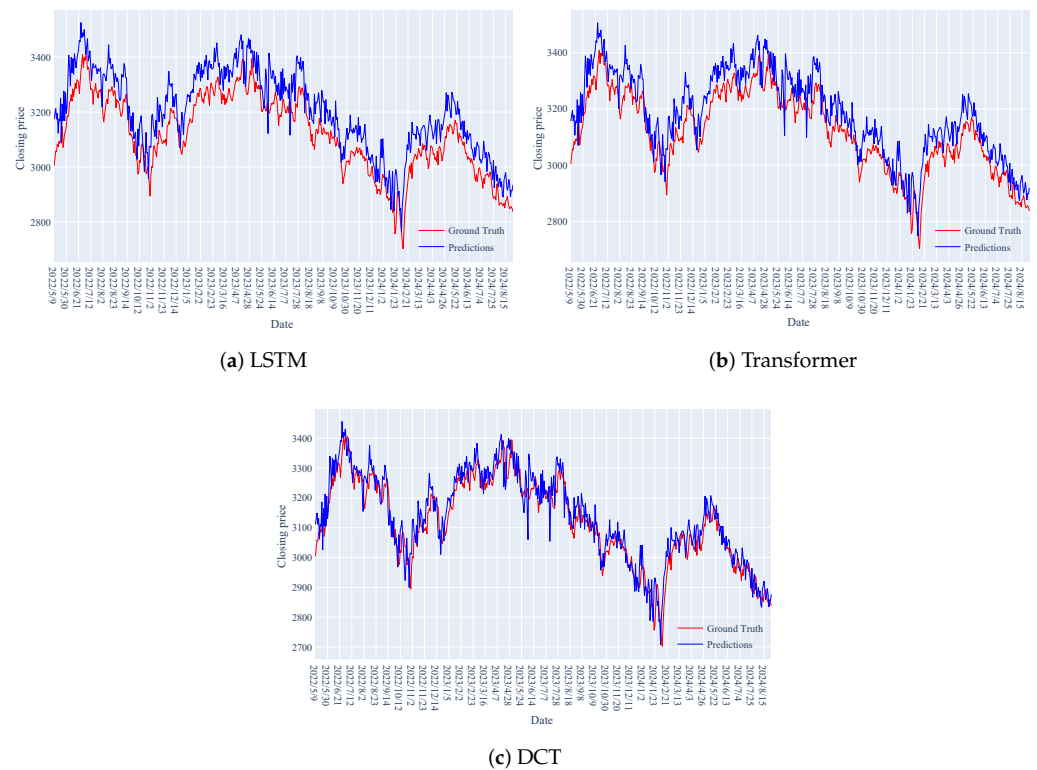


Figure 7. Prediction of stock close prices of SSEC for various stock indexes.

5. Conclusions and Future Work

Predicting and modeling stock movements hold significant economic value. In this study, we propose a novel Deep Convolutional Transformer (DCT) model for stock move-

ment prediction. The DCT model incorporates several innovative elements that enhance its performance.

Firstly, regarding model architecture, the inception convolutional token embedding and separable fully connected layers play crucial roles. The inception convolutional token embedding allows the model to capture both global high-level features and local fine-grained information. It transforms low-level data into valuable high-order semantic primitives, providing a comprehensive understanding of stock data. Conversely, the separable, fully connected layers preserve the temporal order of input features and map them to a high-dimensional feature space. This approach not only enhances the richness and diversity of features but also improves the model's ability to adapt to the complex characteristics of the stock market.

Secondly, the implementation of the multi-head attention mechanism represents another significant contribution. This mechanism enables the network to simultaneously focus on information from various representation subspaces. By utilizing this multi-dimensional attention approach, the model can more effectively capture the intricate relationships among different factors in the stock market, thereby improving both its predictive accuracy and stability.

Through extensive experiments conducted on multiple stock indices, including NASDAQ, HSI, and SSEC, the DCT model has demonstrated both its effectiveness and superiority. In terms of prediction accuracy, it achieved a peak accuracy of 58.85% on the NASDAQ dataset with a sliding window width of 30 days. When compared to other state-of-the-art methods, the DCT model shows improvements of more than 1.99%, 2.19%, and 2.38% across the three datasets, respectively. Regarding error metrics, the DCT model outperforms other models by exhibiting the smallest average prediction error across all datasets for MAE, MSE, and MAPE. Furthermore, it achieves the highest MCC values in all three datasets, indicating a strong correlation between the predicted and actual values.

In conclusion, the DCT model proposed in this research offers significant scientific contributions and practical value in the field of stock prediction. It serves as a more reliable and accurate tool for investors and researchers to forecast stock movements, thereby enabling them to make more informed decisions in the complex stock market environment.

Author Contributions: Methodology, L.X.; Writing—original draft, L.X.; writing—review and editing, Z.C.; supervision, figures, experiments, S.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work supported by the Natural Science Foundation of Guangdong Province, Grant/Award Number (2021A1515011803, 2020A1515010923); the Guangdong Province Universities' Characteristic Innovation Projects (2024KTSCX064); the Teaching Reform Project of Guangdong Province; the Shaoguan Science and Technology Plan project, Grant/Award Numbers (210728104530586, 210728114530796); the Shaoguan University Research Project, Grant Number: SZ2023KJ07.

Data Availability Statement: Data are available upon request due to restrictions, e.g., privacy or ethical. The data presented in this study are available upon request from the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

DCT	Deep Convolutional Transformer
HSI	Hang Seng Index
SSEC	Shanghai Stock Exchange Composite
MAE	Mean Absolute Error
MSE	Mean Square Error
MAPE	Mean Absolute Percentage Error
MCC	Matthews Correlation Coefficient
SVM	Support Vector Machine

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
NLP	Natural Language Processing

References

1. Htun, H.H.; Biehl, M.; Petkov, N. Survey of feature selection and extraction techniques for stock market prediction. *Financ. Innov.* **2023**, *9*, 26. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Mintarya, L.N.; Halim, J.N.; Angie, C.; Achmad, S.; Kurniawan, A. Machine learning approaches in stock market prediction: A systematic literature review. *Procedia Comput. Sci.* **2023**, *216*, 96–102. [\[CrossRef\]](#)
3. Wang, C.; Chen, Y.; Zhang, S.; Zhang, Q. Stock market index prediction using deep Transformer model. *Expert Syst. Appl.* **2022**, *208*, 118128. [\[CrossRef\]](#)
4. Lu, M.; Xu, X. TRNN: An efficient time-series recurrent neural network for stock price prediction. *Inf. Sci.* **2024**, *657*, 119951. [\[CrossRef\]](#)
5. Liu, G.; Ma, W. A quantum artificial neural network for stock closing price prediction. *Inf. Sci.* **2022**, *598*, 75–85. [\[CrossRef\]](#)
6. Yang, Z.; Zhao, T.; Wang, S.; Li, X. MDF-DMC: A stock prediction model combining multi-view stock data features with dynamic market correlation information. *Expert Syst. Appl.* **2024**, *238*, 122134. [\[CrossRef\]](#)
7. Shaban, W.M.; Ashraf, E.; Slama, A.E. SMP-DL: A novel stock market prediction approach based on deep learning for effective trend forecasting. *Neural Comput. Appl.* **2024**, *36*, 1849–1873. [\[CrossRef\]](#)
8. Chatzis, S.P.; Siakoulis, V.; Petropoulos, A.; Stavroulakis, E.; Vlachogiannakis, N. Forecasting stock market crisis events using deep and statistical machine learning techniques. *Expert Syst. Appl.* **2018**, *112*, 353–371. [\[CrossRef\]](#)
9. Tanveer, M.; Rajani, T.; Rastogi, R.; Shao, Y.H.; Ganaie, M. Comprehensive review on twin support vector machines. *Ann. Oper. Res.* **2022**, *339*, 1223–1268. [\[CrossRef\]](#)
10. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012), Lake Tahoe, NV, USA, 3–6 December 2012.
11. Medsker, L.R.; Jain, L. Recurrent neural networks. *Des. Appl.* **2001**, *5*, 2.
12. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M.S. Deep learning for visual understanding: A review. *Neurocomputing* **2016**, *187*, 27–48. [\[CrossRef\]](#)
14. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [\[PubMed\]](#)
15. Jin, C.; Shi, Z.; Lin, K.; Zhang, H. Predicting miRNA-disease association based on neural inductive matrix completion with graph autoencoders and self-attention mechanism. *Biomolecules* **2022**, *12*, 64. [\[CrossRef\]](#)
16. Li, W.; Qi, F.; Tang, M.; Yu, Z. Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. *Neurocomputing* **2020**, *387*, 63–77. [\[CrossRef\]](#)
17. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
18. Arroyo, D.M.; Postels, J.; Tombari, F. Variational transformer networks for layout generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13642–13652.
19. Li, Y.; Yao, T.; Pan, Y.; Mei, T. Contextual transformer networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 1489–1500. [\[CrossRef\]](#)
20. Shi, Z.; Hu, Y.; Mo, G.; Wu, J. Attention-based CNN-LSTM and XGBoost hybrid model for stock prediction. *arXiv* **2022**, arXiv:2204.02623.
21. Lu, W.T.; Wang, J.C.; Won, M.; Choi, K.; Song, X. SpecTNT: A time-frequency transformer for music audio. *arXiv* **2021**, arXiv:2110.09127.
22. Gong, Y.; Chung, Y.A.; Glass, J. Ast: Audio spectrogram transformer. *arXiv* **2021**, arXiv:2104.01778.
23. Zou, C.; Wang, B.; Hu, Y.; Liu, J.; Wu, Q.; Zhao, Y.; Li, B.; Zhang, C.; Zhang, C.; Wei, Y.; et al. End-to-end human object interaction detection with hoi transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11825–11834.
24. Ye, T.; Zhang, J.; Li, Y.; Zhang, X.; Zhao, Z.; Li, Z. CT-Net: An efficient network for low-altitude object detection based on convolution and transformer. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–12. [\[CrossRef\]](#)
25. Yang, F.; Zhai, Q.; Li, X.; Huang, R.; Luo, A.; Cheng, H.; Fan, D.P. Uncertainty-guided transformer reasoning for camouflaged object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 4146–4155.
26. Bhojanapalli, S.; Chakrabarti, A.; Glasner, D.; Li, D.; Unterthiner, T.; Veit, A. Understanding robustness of transformers for image classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10231–10241.

27. Shao, Z.; Bian, H.; Chen, Y.; Wang, Y.; Zhang, J.; Ji, X. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 2136–2147.
28. Yan, S.; Xiong, X.; Arnab, A.; Lu, Z.; Zhang, M.; Sun, C.; Schmid, C. Multiview transformers for video recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 3333–3343.
29. Yang, J.; Dong, X.; Liu, L.; Zhang, C.; Shen, J.; Yu, D. Recurring the transformer for video action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 14063–14073.
30. Zhang, Q.; Qin, C.; Zhang, Y.; Bao, F.; Zhang, C.; Liu, P. Transformer-based attention network for stock movement prediction. *Expert Syst. Appl.* **2022**, *202*, 117239. [\[CrossRef\]](#)
31. Muhammad, T.; Aftab, A.B.; Ahsan, M.; Muhu, M.M.; Ibrahim, M.; Khan, S.I.; Alam, M.S. Transformer-Based Deep Learning Model for Stock Price Prediction: A Case Study on Bangladesh Stock Market. *arXiv* **2022**, arXiv:2208.08300. [\[CrossRef\]](#)
32. Liu, W.J.; Ge, Y.B.; Gu, Y.C. News-driven stock market index prediction based on trellis network and sentiment attention mechanism. *Expert Syst. Appl.* **2024**, *250*, 123966. [\[CrossRef\]](#)
33. Kohli, P.P.S.; Zargar, S.; Arora, S.; Gupta, P. Stock prediction using machine learning algorithms. In *Proceedings of the Applications of Artificial Intelligence Techniques in Engineering: SIGMA 2018*; Springer: Singapore, 2019; Volume 1, pp. 405–414.
34. Nguyen, T.H.; Shirai, K.; Velcin, J. Sentiment analysis on social media for stock movement prediction. *Expert Syst. Appl.* **2015**, *42*, 9603–9611. [\[CrossRef\]](#)
35. Chen, Y.J.; Chen, Y.M.; Lu, C.L. Enhancement of stock market forecasting using an improved fundamental analysis-based approach. *Soft Comput.* **2017**, *21*, 3735–3757. [\[CrossRef\]](#)
36. Malandri, L.; Xing, F.Z.; Orsenigo, C.; Vercellis, C.; Cambria, E. Public mood-driven asset allocation: The importance of financial sentiment in portfolio management. *Cogn. Comput.* **2018**, *10*, 1167–1176. [\[CrossRef\]](#)
37. Rajput, V.; Bobde, S. Stock market forecasting techniques: Literature survey. *Int. J. Comput. Sci. Mob. Comput.* **2016**, *5*, 500–506.
38. Sohangir, S.; Mojra, A. A Numerical Study on Fluid Flow inside the Knee Joint through a Porous Media Approach. In Proceedings of the 2018 25th National and 3rd International Iranian Conference on Biomedical Engineering (ICBME), Qom, Iran, 29–30 November 2018; pp. 1–5.
39. Derakhshan, A.; Beigy, H. Sentiment analysis on stock social media for stock price movement prediction. *Eng. Appl. Artif. Intell.* **2019**, *85*, 569–578. [\[CrossRef\]](#)
40. Xie, L.; Yu, S. Unsupervised feature extraction with convolutional autoencoder with application to daily stock market prediction. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e6282. [\[CrossRef\]](#)
41. Lee, S.W.; Kim, H.Y. Stock market forecasting with super-high dimensional time-series data using ConvLSTM, trend sampling, and specialized data augmentation. *Expert Syst. Appl.* **2020**, *161*, 113704. [\[CrossRef\]](#)
42. Aldhyani, T.H.; Alzahrani, A. Framework for predicting and modeling stock market prices based on deep learning algorithms. *Electronics* **2022**, *11*, 3149. [\[CrossRef\]](#)
43. Das, S.P.; Padhy, S. Support vector machines for prediction of futures prices in Indian stock market. *Int. J. Comput. Appl.* **2012**, *41*, 22–26.
44. Hegazy, O.; Soliman, O.S.; Salam, M.A. A machine learning model for stock market prediction. *arXiv* **2014**, arXiv:1402.7351.
45. Jiang, W. Applications of deep learning in stock market prediction: Recent progress. *Expert Syst. Appl.* **2021**, *184*, 115537. [\[CrossRef\]](#)
46. Wen, Q.; Yang, Z.; Song, Y.; Jia, P. Automatic stock decision support system based on box theory and SVM algorithm. *Expert Syst. Appl.* **2010**, *37*, 1015–1022. [\[CrossRef\]](#)
47. Panigrahi, S.; Mantri, J. Epsilon-SVR and decision tree for stock market forecasting. In Proceedings of the 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), Greater Noida, India, 8–10 October 2015; pp. 761–766.
48. Mahajan Shubhrata, D.; Deshmukh Kaveri, V.; Thite Pranit, R.; Samel Bhavana, Y.; Chate, P. Stock market prediction and analysis using Naïve Bayes. *Int. J. Recent Innov. Trends Comput. Commun.* **2016**, *4*, 121–124.
49. Rahul; Sarangi, S.; Kedia, P.; Monika. Analysis of various approaches for stock market prediction. *J. Stat. Manag. Syst.* **2020**, *23*, 285–293. [\[CrossRef\]](#)
50. Zhao, Y.; Yang, G. Deep Learning-based Integrated Framework for stock price movement prediction. *Appl. Soft Comput.* **2023**, *133*, 109921. [\[CrossRef\]](#)
51. Hiransha, M.; Gopalakrishnan, E.A.; Menon, V.K.; Soman, K. NSE stock market prediction using deep-learning models. *Procedia Comput. Sci.* **2018**, *132*, 1351–1362.
52. Hoseinzade, E.; Haratizadeh, S. CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Syst. Appl.* **2019**, *129*, 273–285. [\[CrossRef\]](#)
53. Selvin, S.; Vinayakumar, R.; Gopalakrishnan, E.; Menon, V.K.; Soman, K. Stock price prediction using LSTM, RNN and CNN-sliding window model. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 1643–1647.
54. Yoshihara, A.; Fujikawa, K.; Seki, K.; Uehara, K. Predicting stock market trends by recurrent deep neural networks. In Proceedings of the PRICAI 2014: Trends in Artificial Intelligence: 13th Pacific Rim International Conference on Artificial Intelligence, Gold Coast, Australia, 1–5 December 2014; Proceedings 13; Springer: Cham, Switzerland, 2014; pp. 759–769.

55. Chen, Y.C.; Huang, W.C. Constructing a stock-price forecast CNN model with gold and crude oil indicators. *Appl. Soft Comput.* **2021**, *112*, 107760. [\[CrossRef\]](#)
56. Long, W.; Lu, Z.; Cui, L. Deep learning-based feature engineering for stock price movement prediction. *Knowl.-Based Syst.* **2019**, *164*, 163–173. [\[CrossRef\]](#)
57. Qin, Y.; Song, D.; Chen, H.; Cheng, W.; Jiang, G.; Cottrell, G. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv* **2017**, arXiv:1704.02971.
58. Wang, C.; Liang, H.; Wang, B.; Cui, X.; Xu, Y. Mg-conv: A spatiotemporal multi-graph convolutional neural network for stock market index trend prediction. *Comput. Electr. Eng.* **2022**, *103*, 108285. [\[CrossRef\]](#)
59. Wang, C.; Ren, J.; Liang, H.; Gong, J.; Wang, B. Conducting stock market index prediction via the localized spatial–temporal convolutional network. *Comput. Electr. Eng.* **2023**, *108*, 108687. [\[CrossRef\]](#)
60. Ma, D.; Yuan, D.; Huang, M.; Dong, L. VGC-GAN: A multi-graph convolution adversarial network for stock price prediction. *Expert Syst. Appl.* **2024**, *236*, 121204. [\[CrossRef\]](#)
61. Liu, M.; Zhu, M.; Wang, X.; Ma, G.; Yin, J.; Zheng, X. ECHO-GL: Earnings Calls-Driven Heterogeneous Graph Learning for Stock Movement Prediction. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 13972–13980. [\[CrossRef\]](#)
62. Qian, H.; Zhou, H.; Zhao, Q.; Chen, H.; Yao, H.; Wang, J.; Liu, Z.; Yu, F.; Zhang, Z.; Zhou, J. MDGNN: Multi-Relational Dynamic Graph Neural Network for Comprehensive and Dynamic Stock Investment Prediction. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 14642–14650. [\[CrossRef\]](#)
63. Li, H.; Shen, Y.; Zhu, Y. Stock price prediction using attention-based multi-input LSTM. In Proceedings of the Asian Conference on Machine Learning, PMLR, Beijing, China, 14–16 November 2018; pp. 454–469.
64. Li, C.; Qian, G. Stock Price Prediction Using a Frequency Decomposition Based GRU Transformer Neural Network. *Appl. Sci.* **2022**, *13*, 222. [\[CrossRef\]](#)
65. Sridhar, S.; Sanagavarapu, S. Multi-head self-attention transformer for dogecoin price prediction. In Proceedings of the 2021 14th International Conference on Human System Interaction (HSI), Gdańsk, Poland, 8–10 July 2021; pp. 1–6.
66. Huang, L. HPMG-Transformer: HP Filter Multi-Scale Gaussian Transformer for Liquor Stock Movement Prediction. *IEEE Access* **2024**, *12*, 63885–63894. [\[CrossRef\]](#)
67. Li, T.; Liu, Z.; Shen, Y.; Wang, X.; Chen, H.; Huang, S. MASTER: Market-Guided Stock Transformer for Stock Price Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 162–170. [\[CrossRef\]](#)
68. Zhang, Q.; Zhang, Y.; Bao, F.; Liu, Y.; Zhang, C.; Liu, P. Incorporating stock prices and text for stock movement prediction based on information fusion. *Eng. Appl. Artif. Intell.* **2024**, *127*, 107377. [\[CrossRef\]](#)
69. Li, S.; Xu, S. Enhancing stock price prediction using GANs and transformer-based attention mechanisms. *Empir. Econ.* **2024**, *1*–31. [\[CrossRef\]](#)
70. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. *Handb. Brain Theory Neural Netw.* **1995**, *3361*, 1995.
71. Dosovitskiy, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
72. Mnih, V.; Heess, N.; Graves, A. Recurrent models of visual attention. In Proceedings of the Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, QC, Canada, 8–13 December 2014.
73. Haut, J.M.; Paoletti, M.E.; Plaza, J.; Plaza, A.; Li, J. Visual attention-driven hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8065–8080. [\[CrossRef\]](#)
74. Prajwal, K.; Afouras, T.; Zisserman, A. Sub-word level lip reading with visual attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5162–5172.
75. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.
76. Zhou, C.; Wu, M.; Lam, S.K. SSA-CNN: Semantic self-attention CNN for pedestrian detection. *arXiv* **2019**, arXiv:1902.09080.
77. Sukhavasi, M.; Adapa, S. Music theme recognition using CNN and self-attention. *arXiv* **2019**, arXiv:1911.07041.

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.