Hindawi
IET Signal Processing
Volume 2023, Article ID 6623718, 15 pages
https://doi.org/10.1049/2023/6623718

Research Article

GLAD: Global–Local Approach; Disentanglement Learning for Financial Market Prediction

Humam M. Abdulsahib and Foad Ghaderi

Human Computer Interaction Lab, Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran

Correspondence should be addressed to Foad Ghaderi; fghaderi@modares.ac.ir

Received 6 July 2023; Revised 28 September 2023; Accepted 7 October 2023; Published 10 November 2023

Academic Editor: Asad Ullah

Accurate prediction of financial market trends can have a great impact on maximizing profits and avoiding risks. Conventional methods, e.g., regression or SVR, or end-to-end training approaches, coined as deep learning algorithms, have restraints as a consequence of capturing noisy and unnecessary data. Financial market's data are composed of stock's price time series that are correlated, and each time series has both global and local dynamics. Inspired by recent advancements in disentanglement representation learning, in this paper, we present a promising model for predicting financial markets that learn disentangled representations of features and eliminate those features that cause interference. Our model uses the informer encoder to extract features, capturing global–local patterns by using the time and frequency domains, augmenting the clean features with time and frequency-based features, and using the decoder to predict. To be more specific, we adopt contrastive learning in the time and frequency domains to learn both global and local patterns. We argue that our methodology, disentangling and learning the influential factors, holds the potential for more accurate predictions and a better understanding of how time series move and behave. We conducted our experiments using the S&P 500, CSI 300, Hang Seng, and Nikkei 225 stock market datasets to predict their next-day closing prices. The results showed that our model outperformed existing methods in terms of prediction error (mean squared error and mean absolute error), financial risk measurement (volatility and max drawdown), and prediction net curves, which means that it may enhance traders' profits.

1. Introduction

Financial markets act as a significant element of economic systems, and predicting them is both a critical ingredient and a challenging problem for market traders and scholars. Many methods, including technical and fundamental analysis [1, 2], have been employed to study the historical behavior of financial markets. Sound financial decision-making is a massive challenge since it depends on accurate prediction, transparency, and trust [3]. Given how quickly machine learning (ML) and, in particular, deep learning (DL) techniques are growing, studying, and understanding stock market movements is an interesting task for ML experts. Indeed, DL methods combined with financial time series prediction methods are among the most attractive research topics as a result of the complex nature of the financial markets [4].

In recent literature, recurrence, convolutional, or hybrid models have proven to be outstanding frameworks for financial market prediction. However, they are not devoid of limitations, including ignoring the long-sequence terms of time series that may cause valuable information to be missed [5], vanishing and exploding gradients [3], high time complexity [6], weak of parallelism, lack of explainability, and so on. In order to connect various data points of a sequence model and build a representation of them, self-attention, sometimes called intra-attention, was introduced to eliminate shortcomings of earlier frameworks [7]. Multihead attention, as the innovation of transformer [7], brings several strengths to the table, like capturing global dependencies, efficient parallelism [8], and context-aware representations [9], making it a valuable tool in a range of applications, like text classification [10] and time series prediction [11]. Researchers should be aware of the limitations of the self-attention mechanism used in transformers [7]. These include the fact that its core operation, the scaled dot product, has quadratic computational complexity and the slowness of inference. Several endeavors have been undertaken to get around this limitation by studying the correlation between the key and the query, which are the fundamental constituents of the attention value. As a case in point, Child et al. [12] suggested that the self-attention probability may display potential sparsity, whereas this paper [13] has evinced that the softmax function can be represented as a probability distribution. One method used for addressing such limitations is ProbSparse self-attention, also denoted as informer [5], which employs the concept of probability through Kullback–Leibler divergence to measure query sparsity. Informer demonstrated its effectiveness in different fields, including time series prediction [14], heating load prediction [15], and wind power prediction [16].

These methods rely on the end-to-end training of the models using observable data. This means it often picks up unwanted or noisy data, propagates errors, and cannot be interpreted. In contrast, the concept of disentanglement learning [17] is relatively new and aims to improve explainability and interpretability by capturing and isolating the main important beneficial factors that make up the data. Learning disentangled representation has been used in a set of studies and yielded impressive results, including developing a new fashion design based on disentangling the features of the images [18], video prediction [19], and medical imaging [20].

Time series are typically modeled through two fundamental methods: (1) in the time domain, which captures temporal correlations between individual data points and helps identify patterns and trends, and (2) in the frequency domain, which extracts relevant data that display periodic or quasiperiodic patterns and provides the spectral content of a time series [21, 22].

Each financial market movement is composed of specific movements of the stocks, called local movements, and changes related to the overall market, called global movements [17]. These movements can be helpful in providing insights for making decisions about the models and providing answers to people who demand explanations for specific decisions. Capturing and separating the underlying factors that explain the data can provide advantages such as reducing sample complexity, offering interpretation potential, and overcoming some DL challenges like the black-box nature of such algorithms. For this reason, learning disentangled global–local representations, which are more valuable for financial market prediction, is the aim of this work.

Unsupervised disentanglement learning is challenging, and that is why contrastive learning, as a promising approach to self-supervised learning, is used to enhance the results by setting similar samples close to each other while dissimilar ones are pushed far apart [23, 24].

Financial markets may have a scarcity of labeled data, which is essential for a good DL model, and data augmentation is a useful approach for improving both the quantity and the quality of the training data [22] by applying transformations or perturbations. Conventional augmenting methods, like scaling and shifting, may result in a mismatch between the augmented data and the target [24].

In this paper, we propose GLAD (Global–Local Approach; Disentanglement Learning for Financial Market Prediction), a prediction model that disentangles financial market movements into global and local patterns. Our framework makes use of an informer module to capture the temporal feature relationships between historical data. Afterward, time and frequency domains are used to capture global–local representation, i.e., (1) a mixture of autoregressive experts is used to extract global representations, and (2) a discrete Fourier transform (DFT) is applied to represent local features. We will use (1) time-based augmentation on global representation and (2) frequency-based augmentation on local representation to augment the extracted features. Contrastive learning, inspired by Woo et al. [25], is used to train global and local representations. We use more prominent stock market indexes, such as the S&P 500, Hang Seng, Shanghai and Shenzhen 300 (CSI 300), and Nikkei 225. In all of the experiments, the accuracy metrics (mean squared error (MSE) and mean absolute error (MAE)), risk measurement, and prediction net curves show that our results are better than the state of the art.

To sum up, our significant contributions include the following:

(1) We propose GLAD, a novel approach for predicting financial market movements using global and local disentanglement with time–frequency contrastive learning.

(2) We overcome the black-box aspect of DL and offer interpretability, how the algorithm works, and explainable artificial intelligence (XAI) by separating the underlying factors.

(3) To tackle the problem of limited available data, we augmented the extracted features in both time and frequency domains.

(4) Our model can provide traders with the source of changes in the financial markets, which will fundamentally increase profits and improve people's decision-making reliability.

This paper's outline is as follows: the literature that has relevance to our work is discussed in the next section. In Section 3, we provide an overview of the theoretical underpinnings of our model. Our model's framework is detailed in Section 4. Section 5 describes the experiments, including parameter settings and evaluation criteria. Section 6 concludes this work.

2. Related Works

Over the recent years of renovation, interest among many academics and market participants has grown exponentially in the financial markets, and this has motivated the use of DL methods to provide more accurate solutions for predicting financial markets [1]. As a result of their inherent complexity, financial markets are always at the core of challenging problems. Prior works in the field of DL usage for analyzing financial markets may be divided into two main categories:

(1) End-to-end learning: These techniques make use of observed data. They are proficient prediction tools, but they possess notable limitations, like learning unnecessary or noisy data. For stock closing price prediction, Lu et al. [26] integrated convolutional neural networks (CNN), bidirectional long short-term memory (BiLSTM), and attention mechanism (AM) to proposed a CNN-BiLSTM-AM architecture. In Cheung et al.'s [27] study, 3D-CNN was used to find a more comprehensive set of elements that may affect crop output and price variations. Using CNN and long short-term memory (LSTM) models to increase accuracy rate, Chen and Huang [28] used eight different input features. They conclude that the proposed method can improve prediction accuracy significantly. By converting technical indicators into 2D pictures and analyzing the pictures using a CNN-LSTM-ResNet architecture, Khodaee et al. [29] predicted stock market turning points. In a more recent work, Wang et al. [3] used a transformer model for forecasting stock market indices. Furthermore, in order to capture the temporal dependency of financial data, Zhang et al. [30] used features for five consecutive calendar days in a transformer architecture and reported favorable results. Some scholars have also shown interest in sentiment analysis for financial market prediction, e.g., Köksal and Özgür [31] predicted market trends by analyzing social media comments and news. Numerous scholarly investigations have been conducted to explore the concept of feature engineering, like this paper [2], which demonstrates the use of the discrete wavelet transform by the authors for decomposing the financial time series data into approximation and detail coefficients. Furthermore, the researchers used chicken swarm optimization as an optimization technique in order to determine the most optimal subset of characteristics.

(2) Disentanglement methods: Relying on the premise that the observed data consists of the interaction of various sources, these techniques prioritize capturing the essential factors and diverse explanatory sources from the observed data and isolating them from each other. The concept of encoder–decoder was used in Hadad et al.'s [32] study to map the data of a financial market to its specified and unspecified components. Chen and Huang [17] focused on disentangling excess and market returns of stocks and showed that, using this approach, the prediction outcome was improved. By disentangling news into positive and negative sentiment, Costola et al. [33] investigated the impact of news on financial markets. Using the generalized autoregressive conditional heteroscedasticity (GARCH-MIDAS) model, this study [34] found that oil supply shocks and oil consumption demand shocks had a comparable effect on the stock market volatilities in Nigeria and South Africa.

Even though generally sound conclusions have been drawn from the prior efforts, limitations like the capture of noisy data or incorrect correlations, as well as constraints related to model capacity, have been pointed out. As of late, self-attention mechanisms have shown exceptional skill in modeling the complex dependencies of time-series data. Financial markets data usually contain temporal correlations [35], while end-to-end learning could be used to model these correlations, it does not provide interpretable predictions, as does disentanglement learning. This work delves explicitly into these limitations and strengthens positive points for better prediction results as well as interpretability, which is crucial for many downstream tasks [36]. This effort is based on the promising progress of disentanglement methods and similar ideas.

3. Global–Local Disentanglement and Its Interpretation

Financial time series are intricate, noisy, and frequently show significant correlation, which means that a variable's past values have weight on its current value along with the impact of other stocks movements. As a result, it is essential to understand the main factors that generate the observed data for analysis and prediction purposes. Our work will begin based on the following three theoretical pillars:

(1) Data from financial markets are highly correlated [37], and this is a result of multiple stock market factors being cointegrated, which leads to both rich and complicated observed data. Disentangling these factors leads to the extraction of meaningful explanatory sources and has a significant impact on the interpretability of financial market prediction models as well as their predictability [36–38].

(2) Data from financial markets consist of both clean features and noise [28]. We can make the best predictions if we find features that accurately describe both local and global patterns [17].

(3) When analyzing the fact that each stock's feature represents one dimension in a financial market dataset, it is quickly concluded that the prediction task is a high-dimensional problem, which is in general a great challenge [39]. To improve prediction rates, it is necessary to take into account both global and local representations [39], as well as the benefits for traders.

4. Problem Formulation

Financial market data consist of data generated from market behavior or overall patterns, referred to as global representation, and specific stock movements, called local dynamics [17]. Suppose X is the price records of a group of stocks in a market, i.e.:

$$X = \left\{ x_{i,1:T_o} \right\}_{i=1}^{n}, \tag{1}$$

where $X_{i,\,1:T_o} = \left\{ x_{i,\,1}, x_{i,\,2}, \ldots, x_{i,\,t_o}, \ldots, x_{i,\,T_i} \right\}$, T_0 is to the number of time points in a stock, $x_{i,\,t_o}$is the observation of stock i at time step t_o, and n is the number of input variables. The ultimate objective of a prediction model is to inference the output $y_{i,\,1:T_o} = \left\{ y_{i,\,1}, y_{i,\,2}, \ldots, y_{i,\,t_o}, \ldots, y_{i,\,T_i} \right\}$, where $y_{i,\,1:T_o}$ represents the future values based on the input time series.

4.1. Encoder–Decoder Models. In several prevalent models, the observed data are encoded into hidden representations, and from the hidden representations, the output representations $y_{i,\,t}$ are inferred. The precision of the results depends on how well the necessary data is captured and on their interdependencies.

4.2. Disentangled Representation. These models aim to capture the underlying factors that generate the raw data. Each x_i has global and local representations. In other words, stock market prices can be decomposed into two separate parts as follows:

(1) Local representations ($x_i -$ local) are the specific stock's movements that reflect the movement of a stock itself.

(2) Global representation (X-Global) is the behavior of a market that consists of overall stock movement and the shared value with other stocks in the same market, which is sometimes called the trend.

Our objective is to improve financial market prediction and help traders figure out the source of variation in stock movements. Separating financial market data into global and local patterns with effective dependency capture can accomplish this.

5. Methodology

In light of what has been said above, a prediction framework that disentangles financial market movements into global and local representations, called GLAD: Global–Local Approach; Disentanglement Learning for Financial Market Prediction, is proposed in this paper. Instead of end-to-end learning from observed data, this work aims to capture and learn usable features from observed data. Please refer to Figure 1 for an overview of our approach, in which an informer encoder–decoder [5] is the backbone of the model.

5.1. Encoder: Extraction of Sequential Input Dependencies. First, the input is fed to an informer encoder, which learns the mapping connection between the long-term relationships of the sequential input. To this end, the input sequence is shaped into a matrix $X \in \mathbb{R}^{d_{\text{model}} \times x_T}$, where d_{model} represents the model dimension. For the embedding layer, we use a time feature to understand the sequence of our stock prices over time, and we use sine and cosine functions to encode data location. Encoder layers receive a combined input of positional encoding and embedded input. The self-attention mechanism in Vaswani et al.'s [7] study is based on the tuple of q_i, k_i, and v_i for each row, which represent query, key, and value, respectively.

In the standard transformer [7], all elements of the input sequence attend to each other, resulting in a dense attention matrix, which makes the complexity of computing the attention scores for all pairs of elements prohibitively high. In place of using a scaled dot product, the lens of a kernel-based attention study [13] suggests viewing attention mechanisms as implicitly defining a kernel or similarity measure between elements in the input sequence. The attention mechanism can be seen as implicitly computing a kernel matrix that quantifies the similarity or relevance between pairs of elements. At the same time, some works, like [12], indicated that there are sparse patterns in self-attention. It was proposed to use "selective" counting techniques on all $p\left(k_j | q_i|\right)$ to determine which elements are attended to and which are ignored.

In view of what was found above, Tsai et al. [13] and Child et al. [12], informer [5], as utilized in our model, proposed ProbSparse attention. ProbSparse attention determines which queries are "important" by comparing the probability of the key-query pair $p(k_i|q_i|)$ with that of a uniform distribution $q\left(k_j|q_i|\right)$ through Kullback–Leibler divergence. Multihead attention addresses the issue of lost information by enabling each head to produce distinct sparse query-key pairs for every head. ProbSparse attention allows the model to focus on the most important elements and capture long-range dependencies without attending to all pairwise combinations. As a result of the ProbSparse self-attention mechanism, a 1D convolutional filter is used on the temporal dimension, along with an activation function. This is done to capture the better features and avoid the redundant combinations of value V. Following that, a max-pooling layer is added with a stride value of 2.

5.2. Decoder: Dynamically Inferring the Output. Instead of inferring the output state step by step from the hidden state, as most traditional decoders do, our decoder, as used in Zhou et al.'s [5] study, outputs by one forward procedure by feeding the following vectors as: $X_{de}^{t} = Concat\left(x_{\text{token}}^{t}, x_{o}^{t}\right) \in \mathbb{R}^{\left(L_{\text{token}} + L_y\right) \times d_{\text{model}}}$, where X_{de}^{t} is the decoder inputs, X_{token}^{t} is the start token length of decoder, L_{token} is a sequence in the input, L_y is the output's length, x_o is the target sequence that pads to zero, and *Concat* is the concatenation operator. In other words, decoder's input is *Concat* (start token length of the decoder, zero padding of target elements). After that, the weight will be measured, and the output will be inferred. The final layer will be a fully connected layer, and the prediction type determines whether the result is univariant or multivariate.

5.3. Global Feature Disentangler. The global feature disentangler is designed to capture global representation. It receives the output of the informer encoder as input and passes it through a mixture of $L + 1$ autoregressive experts, which consist of a 1D causal convolution layer as it can effectively capture the continuous representation within a time series [40] and the kernel size of the ith expert is 2^i which determines the receptive field of the convolutional operation. The causal nature of these layers makes them

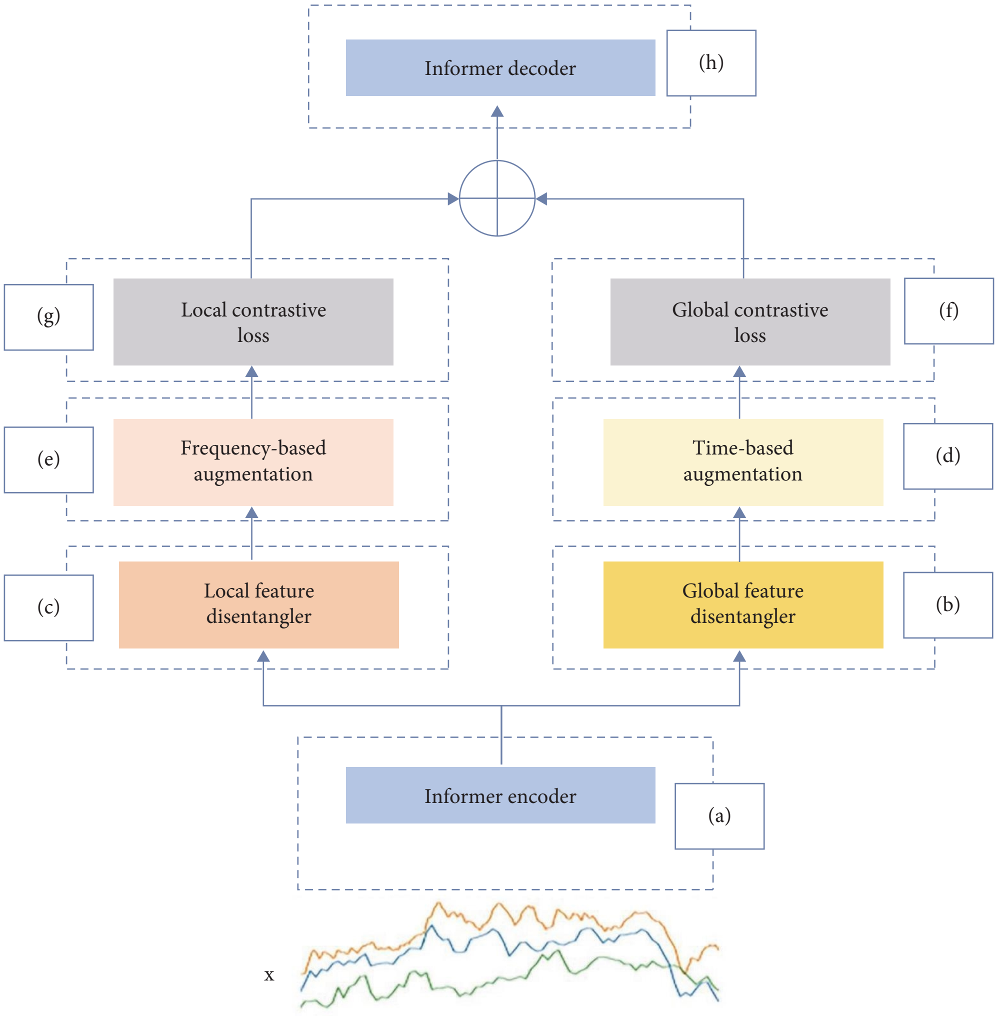


Figure 1: Overview of GLAD architecture. The model is made up of the following components: (a) an informer encoder to capture the temporal relationship between historical data, (b) a global feature disentangler that uses a mixture of autoregressive experts, (c) a local feature disentangler that acts in frequency domain, (d) time-based augmentation is used to augment the global feature by jitter, shift, and scale, (e) frequency-based augmentation by adding frequency, (f) time contrastive loss is used as discriminative global learning, (g) frequency contrastive loss which is used to discriminate local learning, and (h) an informant decoder which is used for final prediction.

particularly suitable for this module, where temporal order and causality are important. Finally, an average-pooling operation, for retaining important information and feature aggregation, is used to get the final global representations.

5.4. Time-Based Augmentation. Through a time-based augmentation, which consists of scaling, shifting, and jittering techniques, as three typical augmentation methods. For scaling and shifting, we used $\tilde{x}_t = \epsilon x_t$ and $\tilde{x}_t = \epsilon + x_t$, respectively, while jittering was performed as follows: $\tilde{x}_t = \epsilon_t + x_t$, where $\tilde{x}_t$ is the output of an augmented method, x_t is a time step, ϵ is a sample of random scalar value $\epsilon \sim \mathcal{N}(0, 0.5)$, and $\tilde{\epsilon}_t$ is Gaussian noise from distribution $\epsilon_t \sim \mathcal{N}(0, 0.5)$.

5.5. Time Domain Contrastive Loss. We used the concept of a dynamic dictionary of MoCo [41] as contrastive learning to learn discriminative global representations. MoCo uses the momentum principle, which leverages a momentum-based update mechanism and contrastive learning to train deep neural networks on unlabeled data, to obtain the positive pairs, which represent samples of data augmentations, and a dynamic dictionary that contains a queue of negative pairs obtained by considering all other samples as negative samples. From Woo et al.'s [25] study, we took the loss function for similarity measured by dot product as shown below:

$$\mathscr{L}_{\text{global}} = \sum_{i=1}^{N} -\log \frac{\exp\left(q_i \cdot \frac{k_i}{\tau}\right)}{\exp\left(q_i \cdot \frac{k_i}{\tau}\right) + \sum_{j=1}^{K} \exp\left(q_i \cdot \frac{k_j}{\tau}\right)}, \qquad (2)$$

where τ is the temperature hyperparameter, q is an encoded query, and k is a set of encoded samples.

5.6. Local Feature Disentangler. This section's primary objective is to obtain a local representation of the data by using the DFT in view of its ability to capture intrafrequency interactions [42] and detect periodic patterns. DFT is used to map the time-domain representations into the frequency domain along the temporal dimension by converting a discrete sequence of N time-domain samples into N frequency-domain components. The resulting frequency components represent the amplitudes and phases of sinusoidal signals. The Fourier transform coefficients are learned using a learnable Fourier layer, which is realized using a per-element linear layer. The inverse DFT method was used to transform the representation back to the time domain. As a result of this layer, we get a matrix that represents the local feature representation. The equation utilized in Woo et al.'s [25] study was employed to represent the constituent of the i, kth element of the output:

$$V_{i,k}^{(L)} = \mathcal{F}^{-1}\left(\sum_{j=1}^{d} A_{i,j,k}\mathcal{F}\left(\widetilde{V}\right)_{i,j} + B_{i,k}\right), \tag{3}$$

where F is the number of frequencies, d is the latent dimension, $A \in \mathbb{C}^{F\times d\times d_l}$ and $B \in \mathbb{C}^{F\times d_l}$ are the parameters, and d_l is the local dimension.

5.7. Frequency-Based Augmentation. We took the idea for frequency-based augmentation from Zhang et al.'s [24] study (https://github.com/mims-harvard/TFC-pretraining.git), whereby frequencies are added or removed based on the frequency's characteristics and generate frequency-based representations. First, we randomly choose E to represent the number of frequency components (amplitude and phase). We will reduce the amplitude of frequency components to zero if we want to remove them, and increase it to $\alpha \cdot A_m$ for adding, where A_m is the maximum frequency–amplitude and α is a predefined constant.

5.8. Frequency-Domain Contrastive Loss. To discriminate between different local patterns, we apply the loss function that was previously employed in Woo et al.'s [25] study.

$$\mathscr{L}_{\text{amp}} = \frac{1}{FN}\sum_{i=1}^{F}\sum_{j=1}^{N} -\log\frac{\exp\left(\left|F_{i,:}^{(j)}\right| \cdot \left|\left(F_{i,:}^{(j)}\right)'\right|\right.}{\exp\left(\left|F_{i,:}^{(j)}\right| \cdot \left|\left(F_{i,:}^{(j)'}\right)\right| + \sum_{k\neq j}^{N}\exp\left(\left|F_{i,:}^{(j)}\right| \cdot \left|F_{i,:}^{(k)}\right|\right)\right)}, \tag{4}$$

$$\mathscr{L}_{\text{phase}} = \frac{1}{FN}\sum_{i=1}^{F}\sum_{j=1}^{N} -\log\frac{\exp\left(\phi F_{i,:}^{(j)}\right) \cdot \phi\left(F_{i,:}^{(j)}\right)'}{\exp\phi\left(F_{i,:}^{(j)}\right) \cdot \phi\left(\left(F_{i,:}^{(j)}\right)'\right) + \sum_{k\neq j}^{N}\exp\phi\left(F_{i,:}^{(j)} \cdot \phi F_{i,:}^{(k)}\right)}, \tag{5}$$

where F is the number of frequencies and $F_{i,:}^{j}$, and $\left(F_{i,:}^{j}\right)'$ are the frequency components and their augmentations, respectively.

6. Experiments and Discussion

In the sections that follow, we discuss the results of our in-depth empirical study of the model and how it compares to other methods of predicting financial markets.

6.1. Datasets. We conducted extensive experiments on four financial market indices, i.e., the S&P 500, the Nikkei 225, the CSI, and the HSI, to demonstrate the predictive ability of our model. We model these daily indices over the period from January 1, 2010 to December 31, 2020.

6.1.1. Features Setting. We set the "close" feature as the target value for our prediction while the input data consists of two scenarios: (1) univariate input, which was the "close," and (2) multivariate features, including "close, open, high, low, adj close, and volume".

6.1.2. Data Processing. The raw data for each feature are a 1D time series; to achieve good data quality, we scale the features to unit variance and zero mean to decrease volatility.

6.1.3. Data Setup. We use the time feature with a fixed-size rolling window to ensure whether the values are taken at equal intervals or not. For the input data, we set the input length for the encoder to 9 and the decoder to 2 to predict the next day's price.

6.2. Experimental Details. The basic information about components and setups is summarized in the following sections.

6.2.1. Metrics. MSE and MAE on each prediction window (averaging in the multivariate case) were used to evaluate this work, with the dataset split 70/30 between train and test, as shown in the following equations:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(\widehat{y}_i - y_i\right)^2, \tag{6}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\widehat{y}_i - y_i\right|, \tag{7}$$

where y_i, $\widehat{y}_i$, and N are the actual value, the predicted value, and the sample size, respectively.

Table 1: The experimental environments' setups.

	Configuration
Processor	2.9 GHz Quad-Core Intel Core i7
Operating system	macOS Ventura Version 13.2.1
Python version	Python 3.9.13 64-bit
Pytorch version	Pytorch 1.8.0

6.2.2. Risks Measurement. In addition to accuracy, it is imperative to evaluate pertinent facets of the trading process, like the measurement of risks related to stock returns, which may be calculated using both real y and predicted values $\hat{y}$. The expression for the return R at time $t+1$ may be written as [3]:

$$R_{t+1} = \ln \frac{y_t + 1}{y_t} \times \text{sign}\left(\hat{y}_{t+1} - y_t\right). \quad (8)$$

where $\text{sign}\left(\hat{y}_{t+1} - y_t\right)$ is the sing function. In this work, two risk-related concepts will be utilized as follows:

(1) Volatility: It is a statistical indicator of how much stock returns have variated over time [3, 43] and can be expressed as:

$$\text{Volatility} = \sigma(R_i). \quad (9)$$

(2) Max drawdown: It measures the most adverse potential result that may arise throughout a trade [3, 44]. It might be written as:

$$\text{Max drawdown} = \max_{i<j} \frac{\text{NV}_j - \text{NV}_i}{\text{NV}_i}, \quad (10)$$

where NV(.) represents the total return.

6.2.3. Environment Configurations. The experimental environment and settings are described in Table 1.

6.2.4. Hyperparameter Tuning. For our model, the backbone encoder used is an informer [5]. We used the Adam optimizer with learning rate starting from $1e^{-4}$, and set the batch size to 32, temperature to 0.07, momentum to 0.999, $\alpha = 0.5$, and $E = 1$ for frequency augmentation. The number of heads is 8. The encoder contains a three-layer stack, while the decoder consists of two layers. The kernel width of distillation is 3.

6.3. Baselines. We benchmarked our model with state-of-the-art approaches to demonstrate how well our model performed. The CoST and informer results are based on our replication with dataset modifications for day, week, and year, while the results of the transformer from the paper are as is. The details are as follows:

(1) End-to-end learning methods (transformer and informer): These methods are based on self-attention mechanisms and end-to-end training.

   (a) Transformer [3]: This work predicts univariate stocks using a transformer encoder–decoder architecture.

   (b) Informer [5]: This paper was used to predict ETT (https://github.com/zhouhaoyi/ETDataset.gi), ECL(https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014), and weather (https://www.ncei.noaa.gov/data/local-climatologicaldata/) dataset and we used the open source implementation (https://github.com/zhouhaoyi/Informer2020.gi) as is.

(2) Disentanglement methods: The main concept behind these techniques is to capture the underlying components of the observed data in the form of clean features and use these features to learn the model.

   (a) CoST [25]: This model (https://github.com/salesforce/CoST.gi) was used to predict ETT, electricity, and weather by using TS2Vec [45] as backbone, capturing seasonal-trend by time–frequency analysis, and learning them by contrastive loss.

   (b) GLAD_a: As a backbone, we utilize the informer [5], and using time–frequency domain characteristics, we were able to capture the global–local patterns.

   (c) GLAD_b: We used the transformer [7] as backbone and captured the global–local by using time–frequency domain features.

   (d) GLAD_c: We used the structure of GLAD_b and improve the results by contrastive learning.

6.4. Interpretability and Explainability of Our Model. There is a fine line between the concepts of interpretability and explainability. Even with their importance, they do not receive the same level of research attention in time series prediction applications as they do in other fields, like computer vision. Our model attempts to attain both, and a brief discussion of each is provided below.

6.4.1. Interpretability: Understanding a Cause with an Effect. The interpretability principle, which is essential for downstream tasks, is about how well decision-makers, like researchers or traders, can understand why a decision was made [46]. To make a model that can be understood, the most important representations must be extracted, and then the model must be trained to learn them [47]. The underlying factors of financial market data, as with other time series, may be represented by time–frequency domains. In prior works, the raw time series were separated into their factors as they were in the original input domain like [17]. This means that the interpretability was based on the time domain. CoST [25] performed separation in

Table 2: Results of univariate price prediction on four datasets.

Stock		End-to-end learning		Representation learning				
	Metric	Transformer [3]	Informer [5]	CoST [25]	GLAD_b	GLAD_c	GLAD_a	GLAD
S&P 500 index	MSE	0.0037	0.0029	0.0038	0.0035	(0.0026)	0.0027	**0.0025**
	MAE	0.0131	0.0125	0.0129	0.0128	(0.0117)	0.0121	**0.0111**
Nikkei 225	MSE	0.0002	0.00015	0.0004	0.00018	(0.00009)	0.00012	**0.00006**
	MAE	0.0017	0.0013	0.0019	0.0015	0.0006	(0.0012)	**0.0006**
Hang Seng HSI	MSE	0.0005	0.00045	0.00091	0.00049	(0.00028)	0.00039	**0.00023**
	MAE	0.0025	0.0022	0.003	0.0023	0.0011	(0.0016)	**0.0011**
CSI	MSE	0.0004	0.00029	0.0009	0.00035	(0.00019)	0.00026	**0.00015**
	MAE	0.0025	0.0021	0.0029	0.0022	(0.0015)	0.0018	**0.0012**

The best results are highlighted in boldface, while the second-best results are enclosed within brackets.

Table 3: Multivariant prices prediction results on four datasets.

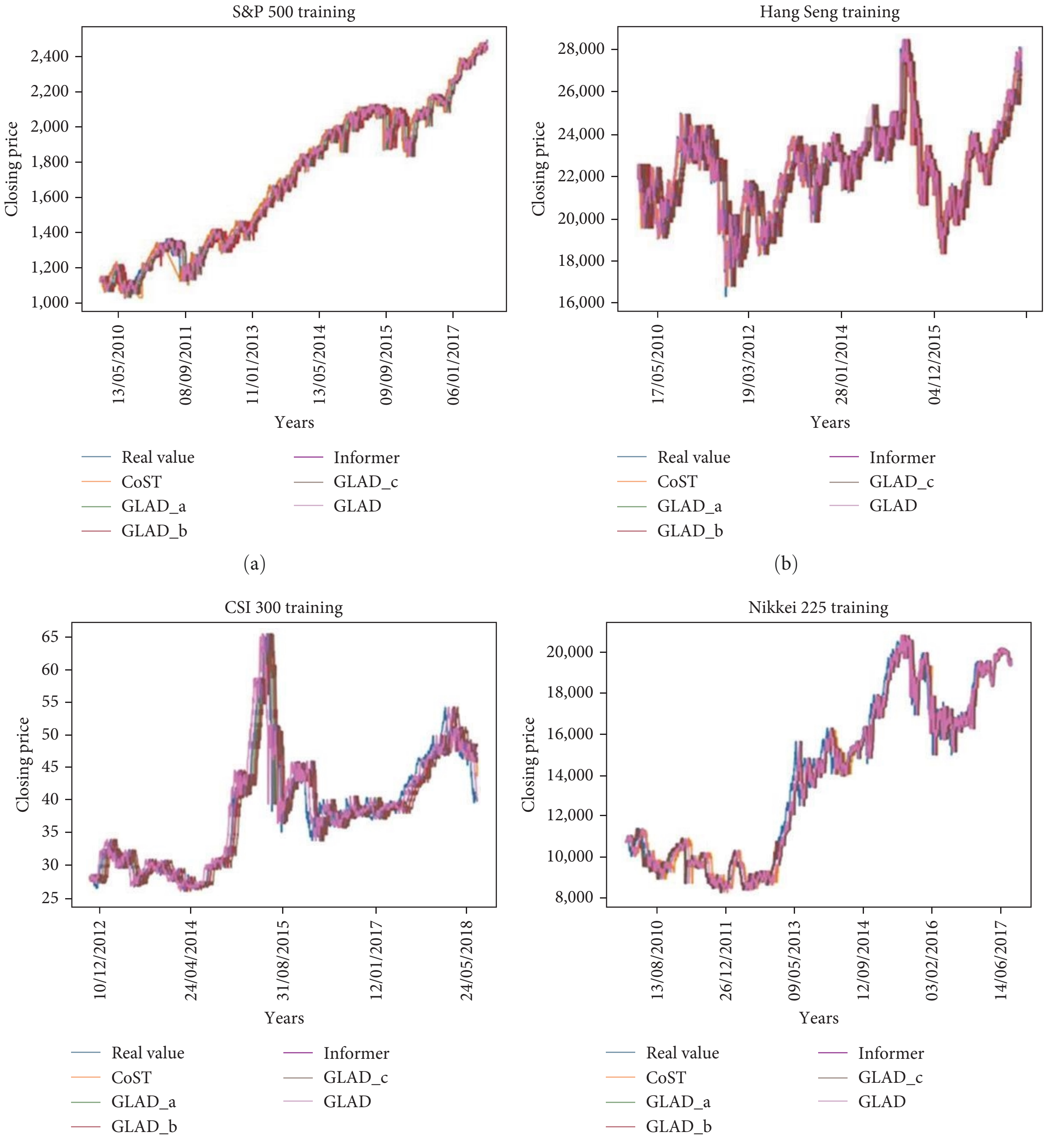
	S&P 500 index			Nikkei 225	
	Univariant	Multivariant		Univariant	Multivariant
MSE	0.0025	0.0031	MSE	0.00006	0.00095
MAE	0.0111	0.0159	MAE	0.0006	0.0073
	HSI			CSI	
	Univariant	Multivariant		Univariant	Multivariant
MSE	0.00023	0.00092	MSE	0.00015	0.00092
MAE	0.0011	0.0084	MAE	0.0012	0.0075

the latent space, not the raw data, utilizing the time–frequency domain, whereas the augmentations are conducted in the time domain. Our method separated the latent space, not the raw data, using time–frequency domains and augmented the clean features, which means the time and frequency domains were augmented.

6.4.2. Explainability: Providing Meaningful Explanations for the Model's Decisions. One problem with traditional DL models is that we cannot see inside them to find out what contents they hold. As a result of the black box, even the people who created it cannot explain why a certain result was obtained. A model is considered explainable if its learning content is understandable, and decomposability models, like the disentanglement approach, are used to achieve explainability. A multistep model and a good latent representation of data inside the model, as well as extracting the key elements of this representation, provide an explanation of the content of the model [48], and this is what our model includes, which tries to overcome the black box nature of DL.

6.5. Results and Analysis. Our experimental results on four datasets are demonstrated in Tables 2 and 3. The best results are highlighted in boldface, while the second-best results are enclosed within brackets. Table 2 provides a summary of the GLAD's results and the top-performing baselines for the univariate setting, while Table 3 reports the results of the multivariate setting of the GLAD. Figures 2 and 3 present the fitted curves, in training and testing sets, generated by GLAD and other models for four main stock market indices. Figure 4 presents a sample of the fitted curves (40 days) in testing, and our GLAD was closer to the real data. The predicted values are close to the real data in both the training and testing sets. In addition to evaluating the accuracy of the model, we employed max drawdown and volatility, two commonly used financial market risk indicators, to appraise its performance. Table 4 presents the outcomes for volatility and max drawdown, indicating that GLAD has a competitive performance in these measures.

6.5.1. Ablation Study. To evaluate the efficacy of each GLAD module, we design two main approaches: (a) end-to-end learning that utilizes raw data in the learning, whether via transformer [3] or informer [5] and (b) representation learning with two scenarios: (1) disentangle the output features from the encoder into global–local representations and (2) implement contrastive learning. The results demonstrate that all of GLAD's components are indispensable, as may be seen in Table 2 and Figure 2. In addition to this, we observed the following other phenomena: (1) self-attention mechanisms, transformer and informer, showed close results to CoST, with informer having superiority, (2) the global–local disentanglement models outperformed transformer and informer as examples of end-to-end learning, (3) informer's



(a)

(b)

(c)

(d)

Figure 2: The predicted curves of the models in training set. (a) S&P 500, (b) Hang Seng, (c) CSI 300, and (d) Nikkei 225.

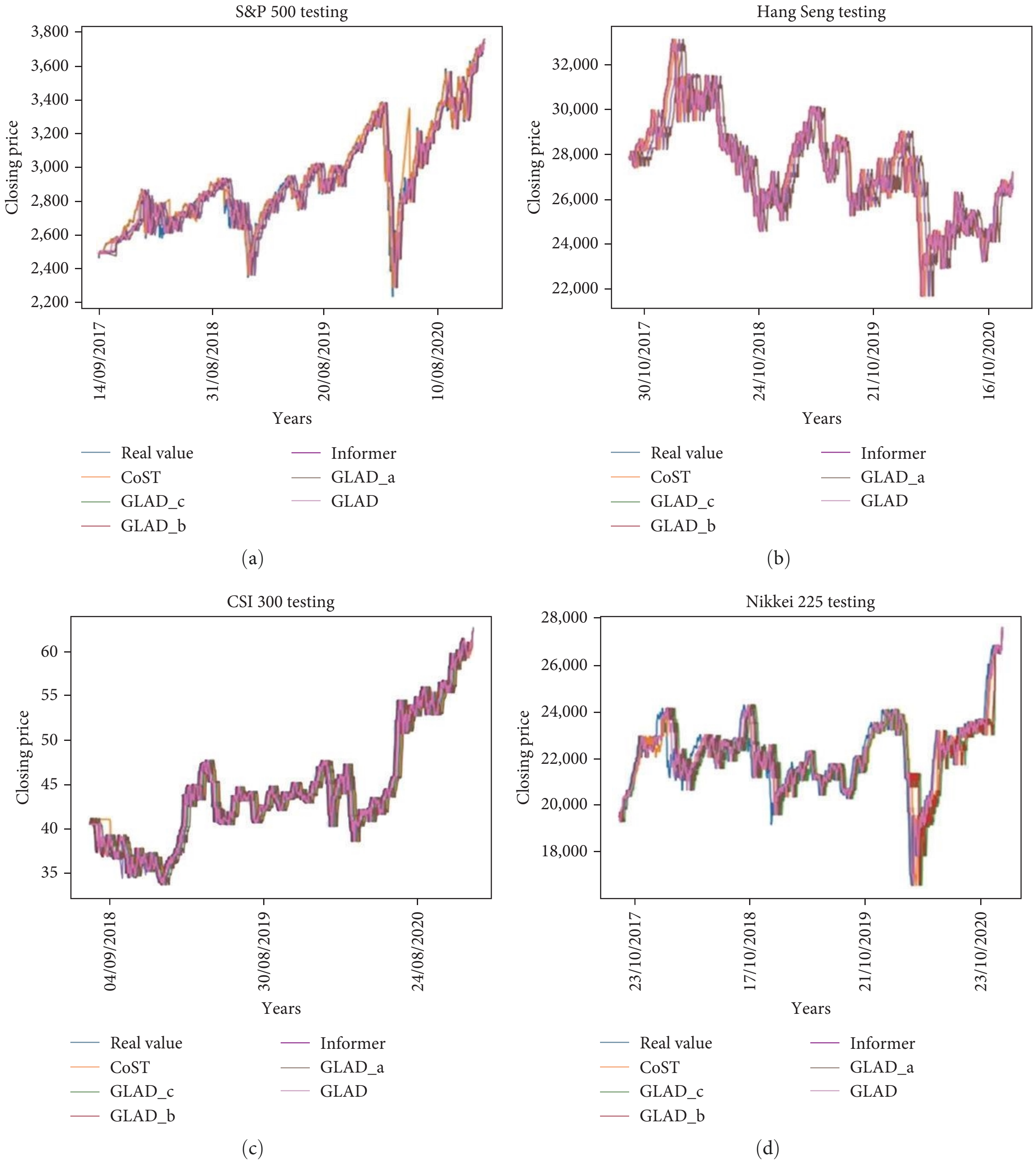
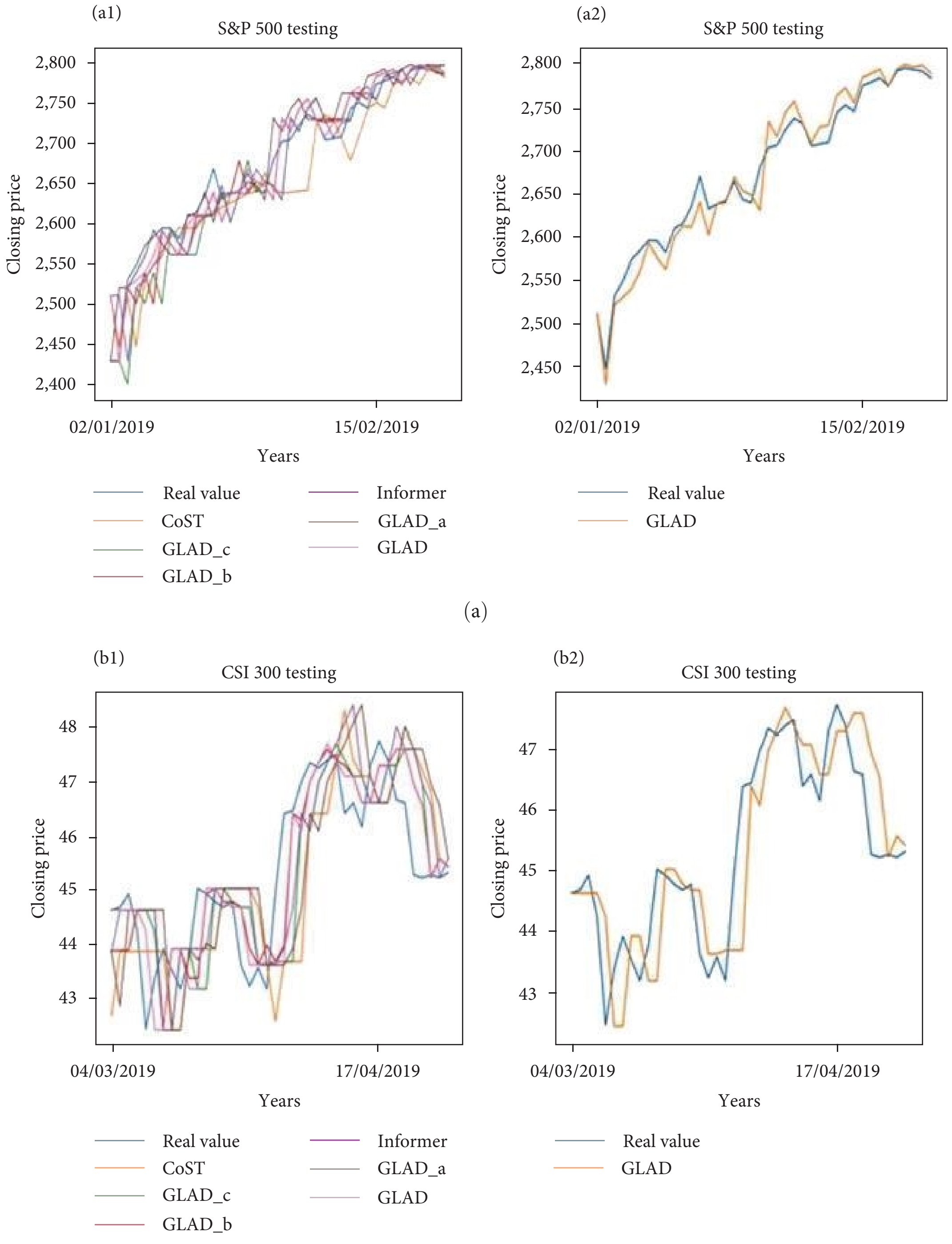


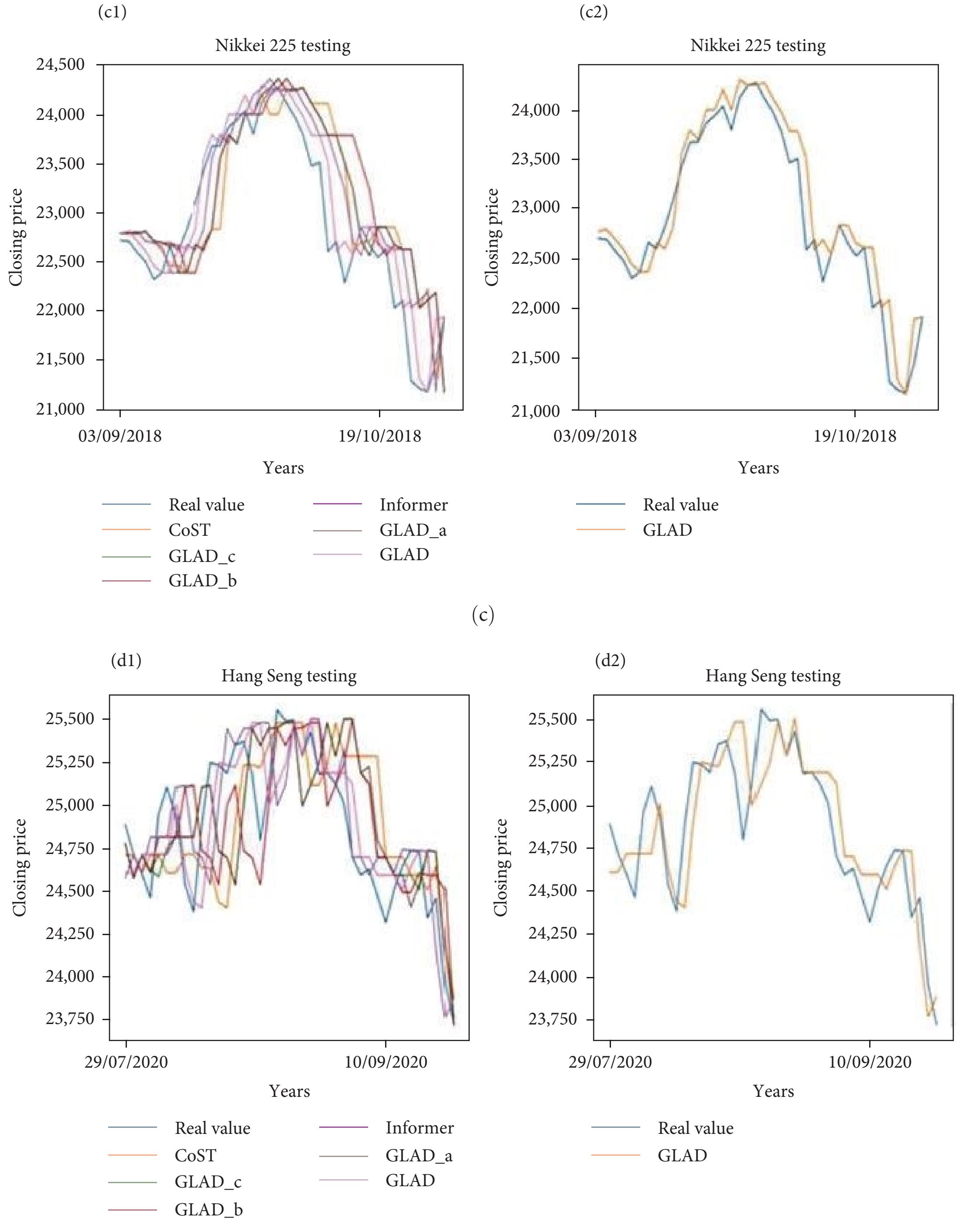
FIGURE 3: The predicted curves of the models in testing set. (a) S&P 500, (b) Hang Seng, (c) CSI 300, and (d) Nikkei 225.



(a)

(b)

Figure 4: Continued.



(c)

(d)

Figure 4: The predicted curves of a sample of data. (a) S&P 500, (b) CSI 300, (c) Nikkei 225, and (d) HSI. (a1), (b1), (c1), and (d1) are samples of fitted curves for GLAD and other models in the testing set. (a2), (b2), (c2), and (d2) are samples of fitted curves for GLAD in the testing set.

Table 4: The maximum drawdown (%) and volatility (%) for four datasets.

Stock		End-to-end learning		Representation learning				
	Metric	Transformer [3]	Informer [5]	CoST [25]	GLAD_b	GLAD_c	GLAD_a	GLAD
S&P 500 index	Volatility	1.62	1.617	1.619	1.619	1.615	(1.61)	**1.60**
	Max drawdown	−28.50	−27.43	−28.26	−26.82	(−24.27)	(−24.27)	**−23.07**
Nikkei 225	Volatility	1.34	1.334	1.337	1.334	**1.321**	(1.322)	**1.321**
	Max drawdown	−19.08	−17.58	−18.57	−17.82	(−15.28)	−15.69	**−14.92**
Hang Seng HSI	Max drawdown	−28.87	−26.41	−27.26	−26.32	**−16.59**	(−17.68)	(−17.68)
CSI 300	Volatility	1.33	1.32	1.33	1.328	(1.315)	1.316	**1.311**
	Max drawdown	−17.14	−16.71	−17.14	−16.63	**−15.66**	−16.34	**−15.66**

The best results are highlighted in boldface, while the second-best results are enclosed within brackets.

performance is superior to that of the transformer, and (4) contrastive learning improves performance over the baselines.

7. Conclusion and Future Work

Our findings point out that disentangling, when it comes to stock market prediction, is a more productive model than conventional end-to-end methods in both prediction error (7.21% improvement in MSE and 4.53% improvement in MAE) and net value analysis, along with financial risk measurement. We have based our work on theoretical background through the nature of financial market movements and experimentally verified it, which showed that our model outperformed the state-of-the-art approaches. For financial market prediction, we introduced GLAD, a framework that disentangles global and local representations. Augmenting financial markets data is a challenge because of the timestamps, which may generate a mismatch between the augmented data (generated by methods such as shifts, scale, and others) and the target. In this paper, we inspired the idea of Zhang et al. [24], but we augmented the extracted features in (1) the time domain, where we adopt shifts, scale, and jitter on global representation, and (2) the frequency domain, in which we add or remove frequency on local representation. Empirical results demonstrated that contrastive learning may improve both learning and the prediction model. Our results make it easier for real-world users to understand what's going on by showing them where the variance and influencing factors come from. In our future research, we will investigate (1) whether this model can predict new stocks to solve the data scarcity challenge by generating local parameters and (2) whether our model has the capability to extend to other time-series datasets.

Data Availability

Data were taken from public sources.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, "Machine learning techniques and data for stock market forecasting: a literature review," *Expert Systems with Applications*, vol. 197, Article ID 116659, 2022.

[2] S. Verma, S. P. Sahu, and T. P. Sahu, "Discrete wavelet transform-based feature engineering for stock market prediction," *International Journal of Information Technology*, vol. 15, pp. 1179–1188, 2023.

[3] C. Wang, Y. Chen, S. Zhang, and Q. Zhang, "Stock market index prediction using deep transformer model," *Expert Systems with Applications*, vol. 208, Article ID 118128, 2022.

[4] D. Kumar, P. K. Sarangi, and R. Verma, "A systematic review of stock market prediction using machine learning and statistical techniques," *Materials Today: Proceedings*, vol. 49, Part 8, pp. 3187–3191, 2022.

[5] H. Zhou, S. Zhang, J. Peng et al., "Informer: beyond efficient transformer for long sequence time-series forecasting," in *The Thirty-Fifth AAAI Conference on Artificial Intelligence*, pp. 11106–11115, AAAI, 2021.

[6] M. Mansouri, K. Dhibi, M. Hajji, K. Bouzara, H. Nounou, and M. Nounou, "Interval-valued reduced rnn for fault detection and diagnosis for wind energy conversion systems," *IEEE Sensors Journal*, vol. 22, no. 13, pp. 13581–13588, 2022.

[7] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 1–11, Neural Information Processing System, Long Beach, CA, USA, 2017.

[8] X. Miao, Y. Wang, Y. Jiang et al., "Galvatron: efficient transformer training over multiple gpus using automatic parallelism," *Proceedings of the VLDB Endowment*, vol. 16, no. 3, pp. 470–479, 2022.

[9] Y. Gou, Y. Lei, L. Liu, Y. Dai, and C. Shen, "Contextualize knowledge bases with transformer for end-to-end task-oriented dialogue systems," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4300–4310, Association for Computational Linguistics, November 2021.

[10] T. Jiang, D. Wang, L. Sun, H. Yang, Z. Zhao, and F. Zhuang, "Lightxml: transformer with dynamic negative sampling for high-performance extreme multi-label text classification," in *The Thirty-Fifth AAAI Conference on Artificial Intelligence*, pp. 7987–7994, AAAI, 2021.

[11] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep transformer models for time series forecasting: the influenza prevalence case," arXiv preprint arXiv: 2001.08317, 2020.

[12] R. Child, S. Gray, A. Radford, and I. Sutskever, “Generating long sequences with sparse transformers,” arXiv preprint arXiv: 1904.10509, 2019.

[13] Y.-H. H. Tsai, S. Bai, M. Yamada, L.-P. Morency, and R. Salakhutdinov, “Transformer dissection: a unified understanding of transformer’s attention via the lens of kernel,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4344–4353, Association for Computational Linguistics, Hong Kong, China, 2019.

[14] Z. Yang, L. Liu, N. Li, and J. Tian, “Time series forecasting of motor bearing vibration based on informer,” *Sensors*, vol. 22, no. 15, Article ID 5858, 2022.

[15] M. Gong, Y. Zhao, J. Sun, C. Han, G. Sun, and B. Yan, “Load forecasting of district heating system based on informer,” *Energy*, vol. 253, Article ID 124179, 2022.

[16] X. Huang and A. Jiang, “Wind power generation forecast based on multi-step informer network,” *Energies*, vol. 15, no. 18, Article ID 6642, 2022.

[17] H. Tang, L. Wu, W. Liu, and J. Bian, “Add: augmented disentanglement distillation framework for improving stock trend forecasting,” arXiv preprint arXiv: 2012.06289, 2020.

[18] N. Fang, L. Qiu, S. Zhang, Z. Wang, K. Hu, and K. Wang, “A novel DAGAN for synthesizing garment images based on design attribute disentangled representation,” *Pattern Recognition*, vol. 136, Article ID 109248, 2023.

[19] T. Pan, Z. Jiang, J. Han, S. Wen, A. Men, and H. Wang, “Taylor saves for later: disentanglement for video prediction using taylor representation,” *Neurocomputing*, vol. 472, pp. 166–174, 2022.

[20] X. Liu, P. Sanchez, S. Thermos, A. Q. O’Neil, and S. A. Tsaftaris, “Learning disentangled representations in the imaging domain,” *Medical Image Analysis*, vol. 80, Article ID 102516, 2022.

[21] S.-Y. Shih, F.-K. Sun, and H.-Y. Lee, “Temporal pattern attention for multivariate time series forecasting,” *Machine Learning*, vol. 108, no. 8-9, pp. 1421–1441, 2019.

[22] Q. Wen, L. Sun, F. Yang et al., “Time series data augmentation for deep learning: a survey,” arXiv preprint arXiv:2002.12478, 2020.

[23] L. Chen, W. Chen, B. Wu, Y. Zhang, B. Wen, and C. Yang, “Learning from multiple time series: a deep disentangled approach to diversified time series forecasting,” arXiv preprint arXiv: 2111.04942, 2021.

[24] X. Zhang, Z. Zhao, T. Tsiligkaridis, and M. Zitnik, “Self-supervised contrastive pre-training for time series via time–frequency consistency,” arXiv preprint arXiv: 2206.08496, 2022.

[25] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, “CoST: contrastive learning of disentangled seasonal-trend representations for time series forecasting,” arXiv preprint arXiv: 2202.01575, 2022.

[26] W. Lu, J. Li, J. Wang, and L. Qin, “A CNN-BiLSTM-AM method for stock price prediction,” *Neural Computing and Applications*, vol. 33, pp. 4741–4753, 2021.

[27] L. Cheung, Y. Wang, A. S. M. Lau, and R. M. C. Chan, “Using a novel clustered 3d-cnn model for improving crop future price prediction,” *Knowledge-Based Systems*, vol. 260, Article ID 110133, 2023.

[28] Y.-C. Chen and W.-C. Huang, “Constructing a stock-price forecast CNN model with gold and crude oil indicators,” *Applied Soft Computing*, vol. 112, Article ID 107760, 2021.

[29] P. Khodaee, A. Esfahanipour, and H. M. Taheri, “Forecasting turning points in stock price by applying a novel hybrid CNN-LSTM-ResNet model fed by 2D segmented images,” *Engineering Applications of Artificial Intelligence*, vol. 116, Article ID 105464, 2022.

[30] Q. Zhang, C. Qin, Y. Zhang, F. Bao, C. Zhang, and P. Liu, “Transformer-based attention network for stock movement prediction,” *Expert Systems with Applications*, vol. 202, Article ID 117239, 2022.

[31] A. Köksal and A. Özgür, “Twitter dataset and evaluation of transformers for turkish sentiment analysis,” in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, Istanbul, Turkey, 2021.

[32] N. Hadad, L. Wolf, and M. Shahar, “A two-step disentanglement method,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 772–780, IEEE, Salt Lake City, UT, USA, 2018.

[33] M. Costola, O. Hinz, M. Nofer, and L. Pelizzon, “Machine learning sentiment analysis, COVID-19 news and stock market reactions,” *Research in International Business and Finance*, vol. 64, Article ID 101881, 2023.

[34] M. M. Tumala, A. A. Salisu, and A. I. Gambo, “Disentangled oil shocks and stock market volatility in Nigeria and South Africa: a GARCH-MIDAS approach,” *Economic Analysis and Policy*, vol. 78, pp. 707–717, 2023.

[35] L. Du, R. Gao, P. N. Suganthan, and D. Z. W. Wang, “Bayesian optimization based dynamic ensemble for time series forecasting,” *Information Sciences*, vol. 591, pp. 155–175, 2022.

[36] Y. Li, X. Lu, Y. Wang, and D. Dou, “Generative time series forecasting with diffusion, denoise, and disentanglement,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 23009–23022, 2022.

[37] L. Yao and J. Li, “Disentangling the effects of open innovation in the time of financial crisis: a strategic choice perspective,” *Journal of Engineering and Technology Management*, vol. 68, Article ID 101746, 2023.

[38] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: a review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[39] R. Sen, H.-F. Yu, and I. S. Dhillon, “Think globally, act locally: a deep neural network approach to high-dimensional time series forecasting,” in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 4837–4846, The ACM Digital Library, 2019.

[40] M. Liu, A. Zeng, M. Chen et al., “Scinet: time series modeling and forecasting with sample convolution and interaction,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 5816–5828, 2022.

[41] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9729–9738, IEEE, Seattle, WA, USA, 2020.

[42] A. Collins Jackson and S. Lacey, “The discrete fourier transformation for seasonality and anomaly detection of an application to rare data,” *Data Technologies and Applications*, vol. 54, no. 2, pp. 121–132, 2020.

[43] N. Dehouche, “Revisiting the volatility of bitcoin with approximate entropy,” *Cogent Economics & Finance*, vol. 10, no. 1, Article ID 2013588, 2022.

[44] A. K. Iyer, S. A. Hoelscher, and C. L. Mbanga, “Target date funds, drawdown risk, and central bank intervention: evidence during the COVID-19 pandemic,” *Journal of Risk and Financial Management*, vol. 15, no. 9, Article ID 408, 2022.

[45] Z. Yue, Y. Wang, J. Duan et al., “Ts2vec: towards universal representation of time series,” in *The Thirty-Sixth AAAI Conference on Artificial Intelligence*, pp. 8980–8987, AAAI, 2022.

[46] T. Miller, “Explanation in artificial intelligence: insights from the social sciences,” *Artificial Intelligence*, vol. 267, pp. 1–38, 2019.

[47] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, “Definitions, methods, and applications in interpretable machine learning,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 44, pp. 22071–22080, 2019.

[48] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser et al., “Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI,” *Information Fusion*, vol. 58, pp. 82–115, 2020.