A hybrid dual-branch model with recurrence plots and transposed transformer for stock trend prediction

Jingyu Su, Haoyu Li, Ruiqi Wang, Wei Guo, Yushi Hao, Jürgen Kurths, Zhongke Gao

Originally published as

Su, J., Li, H., Wang, R., Guo, W., Hao, Y., Kurths, J., Gao, Z. (2025): A hybrid dual-branch model with recurrence plots and transposed transformer for stock trend prediction. - Chaos, 35, 1, 013125.
https://doi.org/10.1063/5.0233275

Chaos
An Interdisciplinary Journal of Nonlinear Science

RESEARCH ARTICLE | JANUARY 10 2025

A hybrid dual-branch model with recurrence plots and transposed transformer for stock trend prediction

Jingyu Su ; Haoyu Li; Ruiqi Wang; Wei Guo ; Yushi Hao; Jürgen Kurths ; Zhongke Gao

Chaos 35, 013125 (2025)
https://doi.org/10.1063/5.0233275

Chaos ARTICLE pubs.aip.org/aip/cha

A hybrid dual-branch model with recurrence plots and transposed transformer for stock trend prediction

Cite as: Chaos 35, 013125 (2025); doi: 10.1063/5.0233275
Submitted: 13 August 2024 · Accepted: 19 December 2024 ·
Published Online: 10 January 2025

View Online Export Citation CrossMark

Jingyu Su,[1] Haoyu Li,[1] Ruiqi Wang,[1] Wei Guo,[1] Yushi Hao,[1] Jürgen Kurths,[2,a)] and Zhongke Gao[1,b)]

AFFILIATIONS

[1]School of Electrical Engineering and Automation, Tianjin University, Tianjin 300072, China
[2]Research Department Complexity Science, Potsdam Institute for Climate Impact Research, 14473 Potsdam, Germany

[a)]**Also at:** Institute of Physics, Humboldt University of Berlin, 12489 Berlin, Germany; School of Mathematical Sciences, SCMS, and CCSB, Fudan University, Shanghai 200433, China
[b)]**Author to whom correspondence should be addressed:** zhongkegao@tju.edu.cn.

ABSTRACT

Stock trend prediction is a significant challenge due to the inherent uncertainty and complexity of stock market time series. In this study, we introduce an innovative dual-branch network model designed to effectively address this challenge. The first branch constructs recurrence plots (RPs) to capture the nonlinear relationships between time points from historical closing price sequences and computes the corresponding recurrence quantification analysis measures. The second branch integrates transposed transformers to identify subtle interconnections within the multivariate time series derived from stocks. Features extracted from both branches are concatenated and fed into a fully connected layer for binary classification, determining whether the stock price will rise or fall the next day. Our experimental results based on historical data from seven randomly selected stocks demonstrate that our proposed dual-branch model achieves superior accuracy (ACC) and F1-score compared to traditional machine learning and deep learning approaches. These findings underscore the efficacy of combining RPs with deep learning models to enhance stock trend prediction, offering considerable potential for refining decision-making in financial markets and investment strategies.

The prediction of stock price trend has received considerable attention due to its potential benefits for investors in making informed decisions and risk management. However, the uncertainty and high volatility of stock time series pose substantial challenges to predict stock trends. It is crucial to have a method that can effectively capture dependencies within stock time series with high accuracy and robustness. We propose a dual-branch model that combines recurrence plots (RPs) and transposed transformers. The RP captures the non-stationarity and chaos of stock price sequences, while the transposed transformer analyzes the intrinsic connections within the multivariate time series of stocks. By integrating both branches, the model predicts the trend of target stocks. Experimental results demonstrate that this framework achieved very good classification performance.

I. INTRODUCTION

Stock trend prediction has attracted much attention from researchers in finance and computer science due to its potential to maximize investment returns. The efficient market hypothesis (EMH) proposed by Fama[1] suggests that security markets, including stock markets, are efficient in processing information. Market prices rapidly and accurately reflect all available information, including both public and non-public data. Consequently, historical stock trading data are a crucial source for predicting future stock price trends, aiding researchers in devising profitable trading strategies for investors. However, the complex dynamics of financial markets and the constant flux of economic indicators pose challenges for predicting stock market movements.[2–4]

In the nascent stages of stock price trend prediction, statistical time series models such as ARIMA and GARCH were common, capturing price volatility patterns through time series statistical properties.[5–8] Babu and Reddy introduced a linear hybrid ARIMA-GARCH model, enhancing multi-step forecasting while preserving data trends, particularly evaluated on the Indian Stock Market.[9] However, the linearity inherent in traditional models, combined with their limited adaptability to the non-linear complexities and high-dimensionality of financial data, often restricts their effectiveness. This limitation results in difficulties in accurately reflecting the intricate dynamics of stock price movements. With the emergence of AI, machine learning methods such as SVM, XGBoost, and Random Forest (RF) have been increasingly used for stock prediction.[4,10–12] Teixeira and De Oliveira proposed a hybrid model combining nearest neighbor classification with technical analysis for intelligent stock trading decisions.[13] Yuan *et al.* demonstrated the effectiveness of integrating feature selection algorithms with machine learning models, particularly the random forest algorithm, for optimizing stock selection in the Chinese A-share market.[14]

Deep learning models have been extensively applied across various domains attributed to its ability to automatically learn complex feature representations from large datasets and handle non-linear problems. Research on stock trend prediction based on deep learning can be categorized into fundamental analysis and technical analysis.[15] Fundamental analysis relies on the analysis of a company's financial statements, industry conditions, economic indicators, and political events. Chen *et al.* improved stock movement prediction by integrating a professionally curated dictionary of fine-grained financial events into a neural model, which combines news, event structures, and trade data. They further improved generalizability through a multi-task framework supervised by the extracted events.[16] Zhang *et al.* introduced the CoATSMP model, which employs a collaborative attention mechanism to adaptively fuse financial news text with numerical stock price data for more accurate stock movement prediction.[17] Technical analysis is predicated on the examination of historical stock data to project forthcoming trends in the financial markets. Yu *et al.* applied a local linear embedding (LLE) algorithm for dimensionality reduction of stock price factors, followed by a back propagation (BP) neural network for stock price prediction.[18] Zhao and Yang proposed the SA-DLSTM model, which integrates an emotion-enhanced convolutional neural network (ECNN) for sentiment analysis, a denoising autoencoder (DAE) for feature extraction, and a long short-term memory (LSTM) network for stock market prediction.[19]

Recurrent neural network (RNN) and its variants, renowned for their capacity to exploit temporal information in time series data, have been widely adopted in financial forecasting. Nelson *et al.* investigated LSTM for stock price prediction, utilizing historical prices and technical indicators to forecast future trends.[20] Jiang *et al.* combined technical indicators with macroeconomic factors in an ensemble framework, integrating RF, Light Gradient Boosting Machine, and LSTM for stock index direction prediction.[21] Zhao *et al.* further refined stock price forecasting by incorporating attention mechanisms into RNN, LSTM, and GRU models, enhancing the extraction of key information from time series data.[22]

In recent years, a variety of time series models based on transformers have emerged, showing promising results in different time series forecasting tasks.[23–25] Ding *et al.* introduced a transformer-based approach for stock movement prediction. Their approach incorporated a multi-scale Gaussian prior to address locality, orthogonal regularization for efficient attention mechanism, and a trading gap splitter to capture features in high-frequency financial data. These enhancements effectively tackle the challenge of modeling long-term dependencies in financial time series.[15] The key difference between transformers and other deep learning models is its use of a multi-head self-attention mechanism to globally learn relationships between different positions and combine features extracted from multiple subspaces to predict the target, thereby enhance the ability to learn long-term dependencies. Transformer-based models have gained popularity in time series prediction tasks and are now regarded as the mainstream approach. However, existing methods may struggle with learning from multi-variate time series and exhibit insufficient local perception. Specifically, the global self-attention mechanism in traditional transformers may overlook local context, and their interpretability and computational complexity in long time series can limit their effectiveness in predicting high-frequency stock time series.[26]

Despite the advancements achieved by deep learning in stock prediction, current methods still lack interpretability and in-depth understanding of market dynamics. Recurrence plot (RP) is a powerful tool for visualizing nonlinear time series data,[27] revealing complex dynamics and periodic behaviors within the sequences. Recurrence quantification analysis (RQA) is a statistical quantification of RP,[28] providing an objective means of analysis through quantification indicators. RP and RQA have been successfully applied across various fields like physics and biology.[29–34] In finance, they are used to study deterministic dependencies within financial data like stocks, exchange rates, and electricity prices. Bastos *et al.* employed RP and RQA to investigate deterministic dependencies in international stock markets, unveiling distinct dynamic characteristics of emerging markets during financial crises.[35] Yin and Shang introduced multiscale recurrence plot (MRP) and multiscale recurrence quantification analysis (MRQA) methods, applying them to the multiscale analysis of nonlinear characteristics hidden in the stock market.[36] Niu and Wang utilized RP and RQA to study the complex dynamics within a financial price model based on a two-dimensional continuum percolation system.[37] These techniques are crucial for identifying cyclical patterns, anomalies, and phase synchronization in financial markets. They serve as robust tools for gaining deep insights into and quantifying the dynamics of financial markets.

In this work, we propose a dual-branch stock trend prediction model that uses RQA and transposed transformers. This approach capitalizes on the transformer's ability to capture inter-relationships within multivariate sequences, complemented by RQA's effectiveness in uncovering complex patterns in financial data. One branch utilizes RP to capture the hierarchical structure and recursive relationships in the closing price sequences of high-frequency stock data, followed by computation of the corresponding RQA measures. The other branch incorporates transposed Transformer, treating each sequence in the multivariate time series as an individual variable to analyze the interrelations between them. Subsequently, the outputs from both branches are concatenated, and these integrated data are then processed through a fully connected layer to predict the fluctuations in the target stock's price. By combining these two branches, we not only capture the complex dynamics of stock prices through RQA but also benefit from the transformer's ability to understand and utilize the intricate interactions between multiple financial indicators. In the conducted experiments, our proposed model has shown superior performance over all baseline models in terms of average accuracy and F1-score on randomly selected stocks.

II. DATA ACQUISITION

Following the approach of Ma *et al.*,[38] we randomly selected seven stocks from different industries in China's A-share market to ensure the representativeness and generalizability of our findings. The selected stocks are shown in Table I. We collected historical data for these stocks spanning from 2018 to 2022, including minute-by-minute information on open, close, high, low, and trading volume. Unlike the daily-level data commonly used in previous studies, our dataset is based on minute-level granularity. Specifically, the minute-level opening price refers to the first transaction price recorded in a given minute, while the minute-level closing price refers to the last transaction price of that minute. The minute-level high price, low price, and trading volume are calculated based on the highest price, lowest price, and the total volume traded within each minute, respectively. Each year contains approximately 250 trading days, with each trading day comprising 240 minutes of trading time. The data from 2018 to 2021 were used as the training set, while the data from 2022 were used as the test set. To standardize the dataset and mitigate volume-related variations, we applied z-score normalization to the collected data.

For a given trading day t, its data are represented as $X_t = \left[x_t^1, x_t^2, \ldots, x_t^l\right] \in \mathbb{R}^{l\times p}$, where $l = 240$ denotes the number of minutes in a trading day, $p = 5$ represents the number of variables at each time point (open, close, high, low, and trading volume), and $x_t^j \in \mathbb{R}^p$ represents the data at the jth minute of the trading day t. To predict the closing price trend of trading day t, we used the data from the preceding five trading days as the input. This input is expressed as $X_t^{input} = [X_{t-T/l}, X_{t-T/l+1}, \ldots, X_{t-1}] \in \mathbb{R}^{T\times p}$, where $T = 5l$ represents the total number of minutes over the past five trading days. The input can also be expanded into a minute-based form, i.e., $X_t^{input} = \left[x_{t-T/l}^1, x_{t-T/l}^2, \ldots, x_{t-T/l}^l, \ldots, x_{t-1}^1, x_{t-1}^2, \ldots, x_{t-1}^l\right]$. To simplify, we rearrange the indices and represent it as $X_t^{input} = \left[x^1, x^2, \ldots, x^T\right]$. Our objective was to predict the trend of the stock price on day t, specifically whether it will gain or loss. Following the approach described in previous studies,[38,39] the label is assigned a value of 1 when the closing price rises and 0 when it falls,

$$y_t = \begin{cases} 1, & x_t^{l,c} - x_{t-1}^{l,c} \geq 0, \\ 0, & x_t^{l,c} - x_{t-1}^{l,c} < 0. \end{cases} \tag{1}$$

In this study, the price fluctuations are based on the closing price. Previous research typically utilized daily data, calculating the price change by comparing the current day's closing price with that of the previous day. Following their approach, we compute the difference using the closing price recorded at the last minute of trading for each trading day. Therefore, $x_t^{l,c}$ represents the last minute-level closing price on the t-th trading day.

TABLE I. The information of selected target stocks.

Stock code	Stock name	Industry
002603.SZ	Yiling Pharmaceutical	Traditional Chinese medicine
002432.SZ	Andon Health	Medical equipment
002564.SZ	THVOW Technology	Professional services
002407.SZ	Do-Fluoride New Materials	Chemical products
000151.SZ	Do-Fluoride New Materials	Trade industry
002864.SZ	Panlong Pharmaceutical	Traditional Chinese medicine
600 152.SH	Veken Technology., Ltd.	Battery industry

III. METHODOLOGY

In our proposed framework, we implement a dual-branch architecture as shown in Fig. 1. The first branch generates RP from the high-frequency stock closing price time series, followed by RQA to extract underlying complex dynamics. Simultaneously, the second branch employs a transposed transformer to capture the

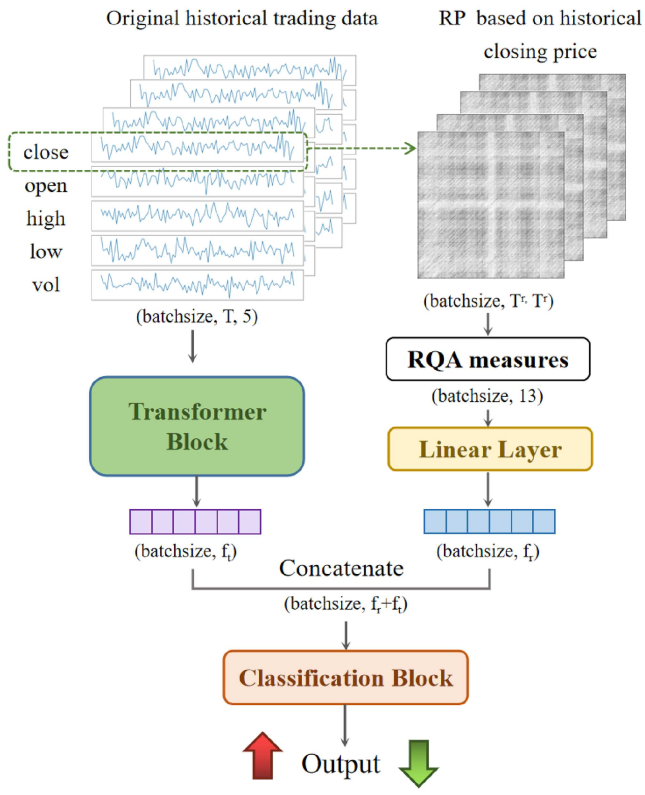

FIG. 1. The framework of the proposed model.

interrelationships among the multivariate raw time series, specifically the interactions between the five time series of closing price, opening price, high price, low price, and trading volume.

A. Recurrence plots and recurrence quantification analysis

RP serves as a valuable tool for visualizing the complexity of trajectories within the phase space of dynamic systems, revealing the system's temporal dynamics by identifying and marking periodic or chaotic-noisy patterns. To analyze the historical closing price sequence of the input sample on day t, denoted as $X_t^{input,c} = \left[x^{1,c}, x^{2,c}, \ldots, x^{T,c}\right] \in \mathbb{R}^T$, we reconstruct the phase space. For clarity, we represent the closing price sequence as $X^c = \left[x_1^c, x_2^c, \ldots, x_T^c\right]$. The extracted trajectories X_i^c are defined as

$$X_i^c = \left[x_i^c, x_{i+\tau}^c, x_{i+2\tau}^c, \ldots, x_{i+(m-1)\tau}^c\right], \quad \forall i \in \{1, \ldots, T-(m-1)\tau\} \tag{2}$$

where m represents the embedding dimension and τ is the time delay. The total number of reconstructed vectors is given by $T^r = T-(m-1)\tau$. These vectors form the basis for constructing a recurrence plot $R \in \mathbb{R}^{T^r \times T^r}$, where connections between any two trajectories X_i^c and X_j^c are determined based on their Euclidean distance,

$$dist\left(X_i^c, X_j^c\right) = \sqrt{\textstyle\sum_{k=1}^m \left(x_{i,k}^c - x_{j,k}^c\right)^2}. \tag{3}$$

A threshold ϵ is set to establish connections between different trajectories within the plot. The Heaviside function Θ is used to represent these connections. Specifically, R_{ij} denotes the element of the recurrence plot and is defined as

$$R_{ij} = \Theta(\epsilon - dist(X_i^c, X_j^c)). \tag{4}$$

If the distance between points i and j is less than the threshold ϵ, then $R_{ij} = 1$, indicating a recurrence between these time points, otherwise, $R_{ij} = 0$.

In this study, we choose a time delay of $\tau = 1$, following previous research[36,40] and determine the embedding dimension m as 7 using the nearest neighbor method. Additionally, the recurrence threshold is set at 10% of the maximum phase space diameter.[37,41]

RPs provide a qualitative time series analysis, while RQA offers a detailed, quantitative, and multidimensional understanding of a system's inherent characteristics and behaviors. Various RQA measures have been proposed to capture different aspects of the system's behavior.[29] In this study, we utilized the PyRQA library to perform RQA measures,[42,43] and the indicators we have selected are outlined in Table II.

Figure 2 provides examples of recurrence plots for three distinct time windows of the stock 000151.SZ and the corresponding RQA measures are shown in Table III. The RR and TT for the first time window are at moderate levels, indicating a certain degree of recurrence in the time series during this period, but it

TABLE II. The details of selected RQA measures.

Measure	Description
Recurrence Rate (RR)	The frequency with which the system's state revisits previous states. A high RR value indicates a high degree of repetitiveness in the system's state space.
Determinism (DET)	Measures the proportion of the system's behavior that is predictable. A high DET value means that the system's future states can largely be inferred from its past states.
Average diagonal line length (L)	The average length of diagonal lines, reflecting the average time delay between system states.
Longest diagonal line (L_{max})	The length of the longest diagonal line, indicating the longest period during which the system remains unchanged in its state space.
Divergence (DIV)	Quantifies the degree to which trajectories in the system's state space diverge, correlating with the system's unpredictability.
Entropy of diagonal lines (L_{entr})	The entropy of the distribution of diagonal line lengths, used to measure the complexity and unpredictability of system states.
Laminarity (LAM)	Represents the proportion of the system's state space where it remains for extended periods, reflecting the stability and periodicity of the system's behavior.
Trapping time (TT)	The average time the system remains within a particular region of its state space, similar to LAM but with a focus on the duration within specific areas.
Longest vertical line (V_{max})	The length of the longest vertical line, indicating the longest duration that the system's state remains unchanged in the time series.
Entropy of vertical lines (V_{entr})	The entropy of the distribution of vertical line lengths, used to measure the complexity of state changes in the time series.
Average white vertical line length (W)	The average length of white vertical line (i.e., recurrence time), which may be related to the system's randomness or noise.
Longest white vertical line (W_{max})	Length of the longest white vertical line in the recurrence matrix, indicating maximum recurrence time in the time series.
Entropy of white vertical lines (W_{entr})	The entropy of the probability to find a white vertical line of exactly length in the RP, reflects the complexity of the RP with respect to white vertical lines.

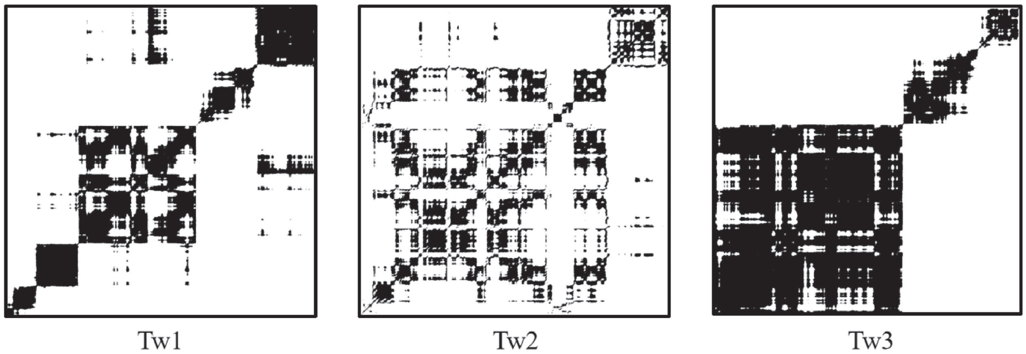

FIG. 2. Recurrence plots of closing price on stock 000151.SZ across three different time windows.

is not strongly pronounced. This suggests that the dynamic pattern is relatively stable, although there is still some randomness and uncertainty. While L_{max} and V_{max} are not particularly remarkable, L_{entr} and V_{entr} are relatively high, reflecting considerable complexity within the system, which may be attributed to short-term fluctuations.

In contrast, the second time window exhibits more complex and unstable characteristics. The RR and TT are considerably lower than in the first window, indicating a reduction in recurring patterns that may be attributed to increased random fluctuations. Additionally, the DIV is relatively high, suggesting that volatility in the series is more pronounced during this period, potentially reflecting a turbulent or rapidly changing market state. The shorter L and L_{max} further support the notion of instability, characterized by brief and frequent dynamic changes.

In the third time window, both the RR and TT increase markedly, indicating a notable rise in recurrence and stability in the system. The significant growth in L_{max} and V_{max} suggests more prolonged state changes. However, the high L_{entr} and V_{entr} levels suggest that while the system is more stable, it still exhibits a significant level of complexity, which might imply the presence of various dynamic elements contributing to its behavior.

The dynamic changes across the different time windows indicate that market behavior may have experienced transitions in stability. By analyzing RQA measures, we can identify variations in recurrence, complexity, and predictability within the system over time.

After computing the RQA measures, we further extract features using two consecutive fully connected layers. The first layer has an input dimension of 13, representing the number of obtained RQA measures, and an output dimension of f_l. A dropout layer is added after the first layer to prevent overfitting. The second layer has an input dimension of f_l and an output dimension of f_r. Here, we set f_l to 128, and f_r is 32.

B. Transformer block

To analyze multi-dimensional time series data, we utilize the transformer encoder, aiming to capture intricate interactions between different variables. As illustrated in the right half of Fig. 3, time series data are always segmented based on timestamp in conventional transformer models, which may lead to the loss of inter-variable information. To address this concern, we adopt a transposed form, treating each time series as a unique token. This strategy enables a more comprehensive analysis of the relationships among multiple variables.

Initially, the original multi-dimensional time series $X_t^{input} = [x^1, x^2, \ldots, x^T] \in \mathbb{R}^{T\times p}$, is processed by applying an initial linear transformation to each time series to obtain the embedding $X_e \in \mathbb{R}^{p\times d_{model}}$, which can be represented as

$$X_e = \left(X_t^{input}\right)^{\top} W_e + b_e, \tag{5}$$

where $W_e \in \mathbb{R}^{T\times d_{model}}$ and $b_e \in \mathbb{R}^{p\times d_{model}}$ are weight and bias, respectively. Here, we set d_{model} to 256.

These preliminary mapped sequences are then fed into a stack of encoders. Each encoder consists of a multi-head attention module followed by a feed-forward network. In the multi-head attention module, the time series is divided into h heads, each with a dimension of $\frac{d_{model}}{h}$. In this work, we employed 8 heads. For the ith head, we compute the mappings for queries $Q_i \in \mathbb{R}^{p\times d_k}$, keys $K_i \in \mathbb{R}^{p\times d_k}$, and values $V_i \in \mathbb{R}^{p\times d_k}$ to calculate the self-attention output. This

TABLE III. RQA measures of closing price on stock 000151.SZ across three different time windows.

	RR	DET	L	L_{max}	DIV	L_{entr}	LAM	TT	V_{max}	V_{entr}	W	W_{max}	W_{entr}
Tw1	0.1857	0.9974	30.0773	733	0.0014	4.2214	0.9978	42.2928	229	4.3451	160.6731	1153	4.8691
Tw2	0.1856	0.9935	13.7142	320	0.0031	3.4723	0.9931	17.8076	124	3.7148	71.4237	1193	4.655
Tw3	0.3558	0.9992	49.7922	1033	0.0010	4.6826	0.9994	72.7067	708	4.7133	124.6286	1190	4.2477

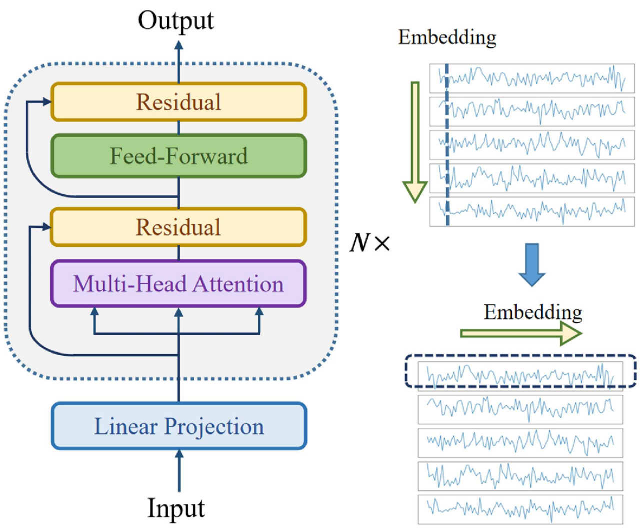

FIG. 3. The architecture of transformer's encoder (left) and transposed form adopted (right).

procedure can be formulated as follows:

$$\text{Attention}_i = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i. \tag{6}$$

The dimension $d_k = \frac{d_{model}}{h}$ represents the keys' dimension and $\sqrt{d_k}$ serves as a scaling factor to prevent gradient vanishing and stabilize the training process. The attention outputs from all heads are then concatenated and passed through a linear layer, which can be described by the following expression:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{Attention}_i, \ldots, \text{Attention}_h) W_O. \tag{7}$$

$W_O \in \mathbb{R}^{d_{model} \times d_{model}}$ is the output weight matrix that integrates the information from all attention heads. The output of the multi-head attention module is then passed through a feed-forward layer to yield the final output of the module. The process can be summarized as follows:

$$\text{FFL}(X) = \text{GeLU}(\text{MHA}(X) W_1 + b_1) W_2 + b_2. \tag{8}$$

$W_1 \in \mathbb{R}^{d_{model} \times d_f}$, $W_2 \in \mathbb{R}^{d_f \times d_{model}}$, $b_1 \in \mathbb{R}^{p \times d_f}$, and $b_2 \in \mathbb{R}^{p \times d_{model}}$ are learnable parameters. Here, we set d_f to 512. MHA(·) and FFL(·) refer to the multi-head attention module and the feed-forward layer, respectively. Additionally, residual connections and layer normalization are applied to the output following the multi-head attention module and the feed-forward network, aiming to mitigate the problem of gradient vanishing or exploding during training.

After obtaining the output $X_t \in \mathbb{R}^{p \times d_{model}}$ after N stacked encoders, we flatten the resulting features to obtain a one-dimensional vector. Then, a fully connected layer is used to reduce its dimensionality to f_t. In our work, we set $f_t = 64$ and N is 2.

C. Classification block

The classification block aims to concatenate features extracted along the first dimension from two branches, followed by a refinement process through two fully connected layers with progressively reduced dimensions. The input dimension of the first fully connected layer is $(f_t + f_r)$, with an output dimension of 20. Subsequent to the first fully connected layer, a dropout layer is integrated to mitigate the risk of overfitting. The second fully connected layer, taking the output of the first as its input, has an input dimension of 20 and an output dimension of 2, specifically targeting a binary classification prediction.

The detailed structure of the proposed model is shown in Table IV.

IV. RESULTS

A. Training setup and evaluation metrics

Due to the distinct market characteristics, volatility patterns, and trading behaviors of each stock, we conducted separate training for the seven selected stocks to better capture their unique features. The model was trained and evaluated on an Nvidia GeForce RTX 3050 GPU using the PyTorch framework. Across all experiments, we employed the Adam optimizer with a learning rate of $3e10^{-6}$

TABLE IV. The detailed structure of the proposed model.

Block	Layer	Parameters	Input shape	Output shape
RQA-branch	Linear_1	dropout: 0.2	(batch size, 13)	(batch size, 128)
	Linear_2		(batch size, 128)	(batch size, 32)
TF-branch	Embedding linear		(batch size, 1200, 5)	(batch size, 256, 5)
	Multi-head attention	heads: 8, dropout: 0.3	(batch size, 256, 5)	(batch size, 256, 5)
	Add and layerNorm			
	Feed-forward	dropout: 0.3	(batch size, 256, 5)	(batch size, 256, 5)
	Add and layerNorm			
	Flatten and linear		(batch size, 256, 5)	(batch size, 64)
Classification	Linear_1	dropout: 0.3	(batch size, 32+64)	(batch size, 20)
	Linear_2		(batch size, 20)	(batch size, 2)

and a batch size of 16. The cross-entropy function served as the loss function, and to prevent overfitting, we applied a weight decay of 0.1.

For stock price trend prediction, we selected three key performance indicators: accuracy (ACC), F1-score, and ROC-AUC, which are defined as follows:

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}. \quad (9)$$

ACC quantifies the proportion of correct predictions made by the classifier. It is calculated by evaluating the number of true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

In addition to accuracy, we also consider the F1-score, which provides a more balanced measure by combining precision and recall,

$$precision = \frac{TP}{TP + FP}, \quad (10)$$

$$recall = \frac{TP}{TP + FN}, \quad (11)$$

$$F_1 = \frac{2 \times precision \times recall}{precision + recall}. \quad (12)$$

The F1-score balances the trade-off between precision and recall, providing a single measure of the model's overall performance. Higher scores indicate better-performing models.

Finally, we use ROC-AUC as a metric to assess the model's ability to distinguish between positive and negative classes. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR), and the AUC is the area under this curve. A higher AUC indicates a better model in distinguishing the two classes.

B. Comparative experiment

To comprehensively evaluate the performance of the proposed model, we compared it with six benchmark models: ARIMA, Random Forest, XGBoost, LSTM, Non-stationary Transformers, and DLinear. Brief introductions to these models are presented below.

ARIMA:[44] a traditional statistical method for time series analysis. It builds predictive models by identifying autocorrelations, trends, and seasonal patterns in the data.

Random Forest:[45] an ensemble technique that enhances predictive accuracy by averaging the results of multiple decision trees.

XGBoost:[46] a gradient boosting framework known for its efficiency and effectiveness in handling complex predictive modeling tasks.

LSTM:[47] a widely used model deep learning model for sequence prediction. It learns long-term dependencies from historical time series data, using unique gating mechanisms to update internal states, thereby enabling the prediction of future trends.

Non-stationary transformers:[48] proposed by Liu *et al.*, effectively addresses the performance degradation of traditional Transformers when handling non-stationary time series data. This is achieved by introducing sequence stabilization and de-stabilization attention mechanisms.

TABLE V. Performance of baselines and the proposed model in accuracy and F1-score.

		000151.SZ	002407.SZ	002432.SZ	002564.SZ	002603.SZ	002864.SZ	600152.SH	Avg.
ARIMA	Acc	52.20 ± 1.71	50.44 ± 4.26	43.36 ± 2.58	53.85 ± 3.70	48.67 ± 2.82	52.21 ± 2.65	53.98 ± 3.15	50.68 ± 4.62
	F1	0.52 ± 0.01	0.53 ± 0.02	0.42 ± 0.01	0.50 ± 0.03	0.53 ± 0.01	0.51 ± 0.01	0.54 ± 0.02	0.51 ± 0.04
	AUC	**0.54 ± 0.01**	0.50 ± 0.00	0.51 ± 0.06	0.53 ± 0.03	0.47 ± 0.02	0.54 ± 0.01	0.51 ± 0.02	0.51 ± 0.04
RF	Acc	49.56 ± 2.15	53.9 ± 0.04	52.21 ± 1.45	56.73 ± 2.35	51.33 ± 2.17	49.56 ± 2.50	51.33 ± 0.71	52.10 ± 3.13
	F1	0.48 ± 0.03	0.54 ± 0.00	0.53 ± 0.02	0.55 ± 0.02	0.57 ± 0.04	0.46 ± 0.04	0.58 ± 0.02	0.53 ± 0.05
	AUC	0.54 ± 0.03	0.51 ± 0.03	0.49 ± 0.03	0.52 ± 0.03	0.51 ± 0.03	0.52 ± 0.02	0.51 ± 0.02	0.51 ± 0.03
Xgboost	Acc	47.79 ± 1.45	51.33 ± 2.10	48.67 ± 0.42	49.04 ± 0.72	51.33 ± 0.72	50.44 ± 2.83	53.98 ± 2.25	50.37 ± 2.63
	F1	0.55 ± 0.01	0.58 ± 0.03	0.56 ± 0.02	0.56 ± 0.07	0.58 ± 0.02	0.55 ± 0.02	0.60 ± 0.06	0.57 ± 0.04
	AUC	0.52 ± 0.02	0.49 ± 0.01	0.53 ± 0.01	0.52 ± 0.01	0.46 ± 0.01	0.49 ± 0.01	0.51 ± 0.01	0.50 ± 0.01
LSTM	Acc	54.69 ± 3.18	60.94 ± 1.92	57.03 ± 1.27	55.36 ± 1.36	46.09 ± 2.41	57.03 ± 2.49	58.59 ± 3.76	55.68 ± 5.02
	F1	0.48 ± 0.04	0.59 ± 0.04	0.62 ± 0.06	0.57 ± 0.03	0.52 ± 0.02	0.62 ± 0.06	0.56 ± 0.01	0.57 ± 0.06
	AUC	0.48 ± 0.05	0.51 ± 0.02	0.52 ± 0.05	0.55 ± 0.04	0.48 ± 0.04	0.50 ± 0.06	0.54 ± 0.01	0.51 ± 0.05
Non-stationary Transformers	Acc	57.81 ± 2.00	**65.18 ± 3.46**	58.59 ± 1.92	**65.18 ± 0.91**	64.06 ± 0.91	48.44 ± 2.36	**65.62 ± 1.27**	60.70 ± 6.18
	F1	0.52 ± 0.03	0.59 ± 0.02	0.53 ± 0.03	0.64 ± 0.02	**0.69 ± 0.04**	0.40 ± 0.05	0.69 ± 0.04	0.58 ± 0.10
	AUC	0.45 ± 0.01	0.50 ± 0.04	0.49 ± 0.05	**0.60 ± 0.01**	**0.58 ± 0.01**	0.49 ± 0.02	0.56 ± 0.03	0.53 ± 0.06
DLinear	Acc	47.66 ± 1.28	62.50 ± 1.27	49.22 ± 2.56	58.93 ± 0.91	60.94 ± 4.56	49.22 ± 1.72	64.06 ± 1.27	56.08 ± 6.95
	F1	0.35 ± 0.04	0.63 ± 0.01	0.36 ± 0.03	0.56 ± 0.02	0.58 ± 0.05	0.54 ± 0.01	0.69 ± 0.04	0.53 ± 0.12
	AUC	0.51 ± 0.01	0.53 ± 0.01	0.46 ± 0.02	0.57 ± 0.01	0.55 ± 0.00	0.48 ± 0.02	0.53 ± 0.01	0.52 ± 0.06
Ours	Acc	**59.38 ± 0.97**	60.94 ± 1.61	**64.06 ± 2.41**	59.82 ± 2.34	**65.62 ± 1.91**	**64.84 ± 0.97**	63.28 ± 1.47	**62.56 ± 2.25**
	F1	**0.64 ± 0.01**	**0.66 ± 0.01**	**0.63 ± 0.02**	**0.68 ± 0.03**	0.67 ± 0.02	**0.55 ± 0.01**	**0.71 ± 0.02**	**0.65 ± 0.06**
	AUC	0.52 ± 0.01	**0.55 ± 0.02**	**0.55 ± 0.02**	0.56 ± 0.02	0.57 ± 0.02	**0.54 ± 0.01**	**0.56 ± 0.01**	**0.55 ± 0.02**

DLinear:[26] introduced by Zeng *et al.*, it decomposes time series into trend and residual sequences and uses two single-layer linear networks to model these sequences.

The results are summarized in Table V, where bold figures indicate the best performance for each metric. The proposed model achieved the highest accuracy in four of the seven stocks, the highest F1-score in six of the stocks, and the best AUC in four of the stocks. It outperformed all baseline models on average in three metrics, showcasing its significant superiority.

Overall, deep learning methods generally outperform classical machine learning and traditional ARIMA models across all three metrics. The difference in average accuracy is the most pronounced, while deep learning methods also demonstrate stronger feature extraction capabilities in terms of F1-score and AUC. For AUC, the differences between models are relatively small, but deep learning methods still demonstrate an advantage over traditional methods, highlighting their superior feature representation capabilities.

Regarding the specific performance of deep learning models, LSTM shows relatively balanced results, but its AUC value is close to that of classical machine learning models. This reflects that LSTM, due to the sequence length limitation, failed to effectively capture discriminative features when handling high-frequency minute-level time series. DLinear exhibits a clear polarization trend, with weaker performance on some stocks across all three metrics, which could be attributed to its reliance on a single-layer linear network structure that fails to adapt to the complex dynamic features in time series. Additionally, its decomposition strategy requires the effective separation of trend and residual sequences, which may not be suitable for high-frequency stock time series with significant volatility and noise, leading to unstable performance. The non-stationary transformer performs better than the proposed model on some stocks across all three metrics, further proving the potential of the transformer architecture for time series prediction. However, its performance volatility suggests that further research and optimization of this method are still necessary.

In general, stock price prediction is influenced by various factors such as macroeconomic conditions, market supply and demand, and industry trends, resulting in differences in data distribution over time that limit the generalization ability of the models. Nevertheless, the proposed model, by innovatively combining the transposed transformer and RQA for time series analysis, still performs best on average. This dual-branch approach effectively learned the intrinsic connections between time points and extracts features from multi-variate time series, enhancing the model's ability to represent and analyze time series data.

C. Ablation study

Our ablation study focuses on two main aspects: (1) analyzing the contributions of the RQA branch and the transformer branch; (2) exploring the performance of the RQA branch when using different time series.

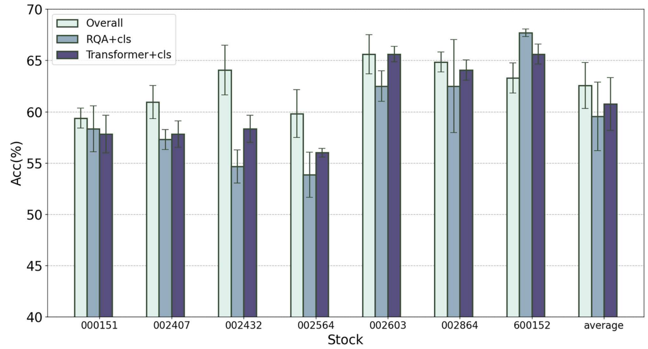

FIG. 4. Performance evaluation of dual-branch vs RQA+cls and transformer+cls architectures.

To further investigate the contributions of each branch to the model's overall performance, we conducted ablation experiments. Figure 4 presents the accuracy for each stock under three configurations: (1) the complete model framework (overall), (2) the classifier fed directly with features extracted from the RQA branch (RQA+cls), and (3) the classifier fed directly with features from the transformer branch (Transformer+cls).

The results demonstrate that the dual-branch architecture achieved optimal prediction accuracy for six out of seven stocks, with the highest average accuracy and lowest standard deviation, indicating robustness and stability. Under the RQA+cls configuration, the model attained an accuracy of 59.55% ± 3.35%. While this configuration performed relatively well for 600 152.SH, its average accuracy was 3.01% lower, and its standard deviation was 1.10% higher than the dual-branch model, reflecting reduced robustness. In the transformer+cls configuration, the model achieved an accuracy of 60.75% ± 2.57%, exhibiting slightly lower performance in accuracy compared to the dual-branch model. These findings highlight that the RQA branch provides sophisticated nonlinear and complexity analyses, capturing dynamic system features and uncovering intrinsic structures within the time series. Meanwhile, the transformer, through its attention mechanism, captures multivariate correlations and learns comprehensive nonlinear representations. By harnessing the complementary strengths of both branches, the dual-branch architecture significantly enhances the model's overall performance.

Next, we explored the impact of different time series selections on the RQA branch. We generated RP using the closing price, opening price, high price, low price, and trading volume separately, followed by RQA. The extracted features were then merged with those

TABLE VI. Comparative accuracy of RQA with various time series inputs. Boldface highlights the best performance of RQA with various time series inputs.

	TF-RQA$_{close}$	TF-RQA$_{open}$	TF-RQA$_{high}$	TF-RQA$_{low}$	TF-RQA$_{vol}$	TF-RQA$_{all}$
Average acc	**62.56 ± 2.25**	61.57 ± 2.26	60.85 ± 2.25	59.84 ± 2.45	58.97 ± 2.35	59.06 ± 2.16

from the Transformer branch for classification predictions. We also considered a combined approach: after performing RQA, we concatenated the features from these five time series and fed them into two consecutive fully connected layers, as previously mentioned. The subsequent operations were consistent with those performed using a single time series after RQA. The accuracy obtained under different conditions is shown in Table VI.

The experimental results demonstrate that using the closing price in the RQA branch achieves the best performance compared to the other four branches. Specifically, the results from the opening and closing prices are similar, while the performance with trading volume is the weakest. This could be due to minute-level trading volume being more sensitive to market fluctuations and random trading behavior, leading to greater volatility. Notably, combining all branches did not yield the expected improvement. This might be attributed to the introduction of additional noise during the input stage, increasing data complexity and exacerbating overfitting, which ultimately reduced the model's generalization ability.

V. CONCLUSION

In summary, we propose a hybrid dual-branch model that combines RQA with transposed transformer architecture for the purpose of stock trend prediction. The first branch leverages RP to uncover nonlinear interdependencies within historical stock data, followed by a detailed analysis using RQA. Meanwhile, the integration of transposed transformer in the second branch allows for intricate pattern detection in multivariate time series. By merging these two branches, we create a comprehensive framework for forecasting stock price volatility.

We collect data for the selected stocks from 2018 to 2022, with 2018 to 2021 as the training set and 2022 as the test set. The experimental results demonstrate an exceptional performance of our proposed model, with an average accuracy of 62.56% an F1-score of 0.65, and an AUC of 0.55 across a selection of seven stocks, surpassing all comparative baseline models. Going forward, we plan to enhance our analytical framework by integrating elements of fundamental analysis, incorporating diverse data sources, and exploring relational dynamics among various stocks, thereby enabling a more holistic analysis.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62373278; in part by the National Natural Science Foundation of Tianjin, China, under Grant No. 21JCJQJC00130; and in part by the Taishan Industrial Experts Program.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Jingyu Su: Methodology (equal); Writing – original draft (equal). **Haoyu Li:** Methodology (equal). **Ruiqi Wang:** Methodology (equal). **Wei Guo:** Methodology (equal). **Yushi Hao:** Methodology (equal). **Jürgen Kurths:** Writing – review & editing (equal). **Zhongke Gao:** Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] E. F. Fama, "Efficient capital markets," J. Finance **25**, 383–417 (1970).

[2] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," Expert Syst. Appl. **42**, 259–268 (2015).

[3] Z. Hu, W. Liu, J. Bian, X. Liu, and T.-Y. Liu, "Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (Association for Computing Machinery, 2018), pp. 261–269.

[4] M.-C. Lee, "Using support vector machine with a hybrid feature selection method to the stock trend prediction," Expert Syst. Appl. **36**, 10896–10904 (2009).

[5] J.-H. Wang and J.-Y. Leu, "Stock market trend prediction using ARIMA-based neural networks," in *Proceedings of International Conference on Neural Networks (ICNN'96)*, Vol. 4 (IEEE, 1996), pp. 2160–2165.

[6] N. Merh, V. Saxena, and K. R. Pardasani, "Next day stock market forecasting: An application of ANN and ARIMA," IUP J. Appl. Finance **17**, 70 (2011).

[7] N. Li, J. Yang, and X. Liang, "Mining associations between trading volume volatilities and financial information volumes based on GARCH model and neural networks," in *The First International Conference on Management Innovation*, Vol. 1 (Citeseer, 2007), pp. 388–396.

[8] C.-C. Lu and C.-H. Wu, "Support vector machine combined with GARCH models for call option price prediction," in *2009 International Conference on Artificial Intelligence and Computational Intelligence*, Vol. 1 (IEEE, 2009), pp. 35–40.

[9] C. N. Babu and B. E. Reddy, "Selected indian stock predictions using a hybrid ARIMA-GARCH model," in *2014 International Conference on Advances in Electronics Computers and Communications* (IEEE, 2014), pp. 1–6.

[10] S. Basak, S. Kar, S. Saha, L. Khaidem, and S. R. Dey, "Predicting the direction of stock market prices using tree-based classifiers," North Am. J. Econ. Finance **47**, 552–567 (2019).

[11] A. Dezhkam and M. T. Manzuri, "Forecasting stock market for an efficient portfolio by combining XGBoost and Hilbert–Huang transform," Eng. Appl. Artif. Intell. **118**, 105626 (2023).

[12] Z. Wang, Z. Hu, F. Li, S.-B. Ho, and E. Cambria, "Learning-based stock trending prediction by incorporating technical indicators and social media sentiment," Cogn. Comput. **15**, 1092–1102 (2023).

[13] L. A. Teixeira and A. L. I. De Oliveira, "A method for automatic stock trading combining technical analysis and nearest neighbor classification," Expert Syst. Appl. **37**, 6885–6890 (2010).

[14] X. Yuan, J. Yuan, T. Jiang, and Q. U. Ain, "Integrated long-term stock selection models based on feature selection and machine learning algorithms for China stock market," IEEE Access **8**, 22672–22685 (2020).

[15] Q. Ding, S. Wu, H. Sun, J. Guo, and J. Guo, "Hierarchical multi-scale Gaussian transformer for stock movement prediction," in *IJCAI* (Morgan Kaufmann, 2020), pp. 4640–4646.

[16] D. Chen, Y. Zou, K. Harimoto, R. Bao, X. Ren, and X. Sun, "Incorporating fine-grained events in stock movement prediction," arXiv:1910.05078 (2019).

[17] Q. Zhang, Y. Zhang, F. Bao, Y. Liu, C. Zhang, and P. Liu, "Incorporating stock prices and text for stock movement prediction based on information fusion," Eng. Appl. Artif. Intell. **127**, 107377 (2024).

[18] Z. Yu, L. Qin, Y. Chen, and M. D. Parmar, "Stock price forecasting based on LLE-BP neural network model," Phys. A **553**, 124197 (2020).

[19] Y. Zhao and G. Yang, "Deep learning-based integrated framework for stock price movement prediction," Appl. Soft Comput. **133**, 109921 (2023).
[20] D. M. Nelson, A. C. Pereira, and R. A. De Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *2017 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2017), pp. 1419–1426.
[21] M. Jiang, J. Liu, L. Zhang, and C. Liu, "An improved stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms," Phys. A **541**, 122272 (2020).
[22] J. Zhao, D. Zeng, S. Liang, H. Kang, and Q. Liu, "Prediction model for stock price trend based on recurrent neural network," J. Ambient Intell. Humanized Comput. **12**, 745–753 (2021).
[23] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35 (AAAI, 2021), pp. 11106–11115.
[24] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International Conference on Machine Learning* (PMLR, 2022), pp. 27268–27286.
[25] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," arXiv:2211.14730 (2022).
[26] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37 (AAAI, 2023), pp. 11121–11128.
[27] J.-P. Eckmann, S. Kamphorst, and D. Ruelle, "Recurrence plots of dynamical systems," Europhys. Lett. **4**, 973–977 (1987).
[28] J. P. Zbilut and C. L. Webber Jr, "Embeddings and delays as derived from quantification of recurrence plots," Phys. Lett. A **171**, 199–203 (1992).
[29] N. Marwan, M. C. Romano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex systems," Phys. Rep. **438**, 237–329 (2007).
[30] J. Lu and K.-Y. Tong, "Robust single accelerometer-based activity recognition using modified recurrence plot," IEEE Sens. J. **19**, 6317–6324 (2019).
[31] Y. Hirata, "Recurrence plots for characterizing random dynamical systems," Commun. Nonlinear Sci. Numer. Simul. **94**, 105552 (2021).
[32] Z. Zhao, Y. Zhang, Z. Comert, and Y. Deng, "Computer-aided diagnosis system of fetal hypoxia incorporating recurrence plot with convolutional neural network," Front. Physiol. **10**, 255 (2019).
[33] H. Zhang, C. Liu, F. Tang, M. Li, D. Zhang, L. Xia, S. Crozier, H. Gan, N. Zhao, W. Xu *et al.*, "Atrial fibrillation classification based on the 2D representation of minimal subset ECG and a non-deep neural network," Front. Physiol. **14**, 1070621 (2023).
[34] C. Hao, R. Wang, M. Li, C. Ma, Q. Cai, and Z. Gao, "Convolutional neural network based on recurrence plot for EEG recognition," Chaos **31**, 0 (2021).
[35] J. A. Bastos and J. Caiado, "Recurrence quantification analysis of global stock markets," Phys. A **390**, 1315–1325 (2011).
[36] Y. Yin and P. Shang, "Multiscale recurrence plot and recurrence quantification analysis for financial time series," Nonlinear. Dyn. **85**, 2309–2352 (2016).
[37] H. Niu and J. Wang, "Multifractal and recurrence behaviors of continuum percolation-based financial price dynamics," Nonlinear Dyn. **83**, 513–528 (2016).
[38] Y. Ma, R. Mao, Q. Lin, P. Wu, and E. Cambria, "Multi-source aggregated classification for stock price movement prediction," Inf. Fusion **91**, 515–528 (2023).
[39] Q. Li, J. Tan, J. Wang, and H. Chen, "A multimodal event-driven LSTM model for stock prediction using online news," IEEE Trans. Knowl. Data Eng. **33**, 3323–3337 (2020).
[40] J. P. Zbilut, "Use of recurrence quantification analysis in economic time series," in *Economics: Complex Windows* (Springer, 2005), pp. 91–104.
[41] J. P. Zbilut, J.-M. Zaldivar-Comenges, and F. Strozzi, "Recurrence quantification based Liapunov exponents for monitoring divergence in experimental data," Phys. Lett. A **297**, 173–181 (2002).
[42] T. Rawald, M. Sips, and N. Marwan, "Pyrqa—Conducting recurrence quantification analysis on very long time series efficiently," Comput. Geosci. **104**, 101–108 (2017).
[43] T. Rawald, M. Sips, N. Marwan, and D. Dransch, "Fast computation of recurrences in long time series," in *Translational Recurrences: From Mathematical Theory to Real-World Applications* (Springer, 2014), pp. 17–29.
[44] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control* (John Wiley & Sons, 2015).
[45] L. Breiman, "Random forests," Mach. Learn. **45**, 5–32 (2001).
[46] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, 2016), pp. 785–794.
[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput. **9**, 1735–1780 (1997).
[48] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," Adv. Neural Inform. Process. Syst. **35**, 9881–9893 (2022).