Journal of King Saud University Computer and Information Sciences (2025) 37:72
https://doi.org/10.1007/s44443-025-00045-y

ORIGINAL PAPER

A novel transformer-based dual attention architecture for the prediction of financial time series

Anita Hadizadeh[1] · **Mohammad Jafar Tarokh**[1] · **Majid Mirzaee Ghazani**[1]

Received: 30 January 2025 / Accepted: 22 April 2025 / Published online: 16 June 2025

Abstract

Financial prediction has gained significant attention due to the complex and non-linear dynamics of the market. A promising approach for generating accurate predictions is Transformers. Encoder-decoder structures efficiently capture complex temporal dependencies and patterns within large-scale data. However, relying on a single attention mechanism may limit the model's ability to capture more intricate relationships. This paper proposes a dual attention architecture to improve the encoder-decoder framework for financial forecasting. First, the Price Attention Network (PAN) extracts complex features from price data and forecasts future prices using historical price inputs. Two key improvements are introduced to enhance self-attention: a Masked Self-Attention module focusing on the most relevant information and Multi-head Attention facilitating more profound insights into the data. Second, the Nonprice Attention Network (NAN) is proposed as a parallel network that processes related financial features. This network utilizes ConvLSTM, BiGRU, and Self-Attention to dynamically weigh and extract meaningful information from nonprice data. Finally, the PAN and NAN networks are integrated, enhancing prediction accuracy. The proposed approach outperforms five state-of-the-art models. Moreover, qualitative assessments of over 26 financial datasets, spanning large and small datasets with short and long histories, further validate the proposed model's ability. Evaluations using seven metrics show the model's superiority, achieving a Mean Absolute Error (MAE) of 0.01991, Mean Squared Error (MSE) of 0.00084, Mean Pinball Loss (MPL) of 0.00996, Symmetric Mean Absolute Percentage Error (SMAPE) of 3.03324, and Mean Absolute Scaled Error (MASE) of 1.85436. This framework represents a significant advancement in financial prediction, offering accurate and interpretable forecasts across various time series tasks.

Keywords Financial market prediction · Transformer · Bidirectional Gated Recurrent Units · ConvLSTM

Abbreviations

ADA	Cardano
Aragon	Aragon (Decentralized Governance platform)
ATT	Attention
BAT	Basic Attention Token
BCH	Bitcoin Cash
BiGRU	Bidirectional Gated Recurrent Unit
BiLSTM	Bidirectional LSTM
BN	Batch Normalization
BNB	Binance Coin
BO	Bayesian optimization
BTC	Bitcoin
CNN	Convolutional Neural Networks
Conv1D	1×1 D Convolutional Layer
ConvLSTM	Convolutional Long-Short-Term Memory Networks
CSI 300	China Securities Index 300
DA	Direction Accuracy
DNN	Deep Neural Networks
DOT1	Polkadot
DRAGAN	Deep Regret Analytic Generative Adversarial Network
ETH	Ethereum
FTS	Fuzzy Time Series
GA	Genetic Algorithm
GRU	Gated Recurrent Unit
HSI	Hang Seng Index
KOSPI200	Korea Composite Stock Price Index 200

✉ Mohammad Jafar Tarokh
mjtarokh@kntu.ac.ir

Anita Hadizadeh
anita.hadizadeh@email.kntu.ac.ir

Majid Mirzaee Ghazani
majidmirzaee@kntu.ac.ir

[1] Department of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran

LSTM	Long Short-Term Memory Neural Networks
LTC	Litecoin
MA	Multi-Head Attention
MAE	Mean Absolute Error
MASE	Mean Absolute Scaled Error
MCA	Multi-Head Cross Attention
MCO	Monaco
MLP	Multilayer Perceptron
MMSA	Multi-Head Masked Self-Attention
MonaCoin	MonaCoin
MPL	Mean Pinball Loss
MSE	Mean Squared Error
N225	Nikkei 225
NAN	Nonprice Attention Network
NEM	New Economy Movement
NLP	Natural Language Processing
NYSE	New York Stock Exchange Composite Index
PAN	Price Attention Network
PE	Positional Encoding
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
S&P500	Standard & Poor's 500 Index
SA	Self Attention
SDTP	Series Decomposition Transformer with Period-correlation
SH	Shanghai
SMAPE	Symmetric Mean Absolute Percentage Error
SSE	Shanghai Stock Exchange
SVM	Support Vector Machine
SVR	Support Vector Regression
SZSE	Shenzhen Stock Exchange
Tanh	Hyperbolic tangent
TST	Time-Series Transformer
XLM	Stellar
XRP	Ripple

1 Introduction

Financial prediction is a challenging but attractive field that creates opportunities for investors and market researchers to predict future profit. Fluctuations in financial markets increase investment risks. Various types of information, including technical factors, fundamental factors, quantitative data, and crowd-sourced data (Bustos and Pomares-Quimbaya 2020), and different data science techniques, including statistical theory, machine learning, and deep learning, have been utilized to predict financial market trends for accurate predictions (Thakkar and Chaudhari 2021a).

Recent studies concentrate on deep learning methods with big historical data in different data science techniques. Most studies are based on recurrent neural networks (RNN), exceptionally long short-term memory neural networks (LSTM), deep multi-layer perceptron (DMLP), and convolutional neural networks (CNN) (Sezer et al. 2020). Although these methods have been widely used in financial market prediction, they have some limitations. For example, the lack of rotational invariance in CNNs may cause the network to assign a different label to an object incorrectly, and the pooling layers in CNNs may lead to the loss of valuable information (Xi et al. 2017; Wang et al. 2022). RNNs have long training times, require significant memory resources, are prone to overfitting, and are highly sensitive to the initialization of network weights (Wang et al. 2022). LSTMs face limitations in accurately predicting because they struggle to capture the complexity and volatility inherent in financial markets. Additionally, LSTM models are prone to overfitting, resulting in poor generalization and limited applicability to real-world financial market scenarios (Bao et al. 2017).

Transformers utilize their self-attention mechanism, outperforming hybrid CNN- LSTMs in capturing long-range dependencies. This mechanism enables them to model relationships across an entire input sequence without the limitations of fixed-size receptive fields or sequential processing (Vaswani 2017). While the application of transformer-based architectures in financial market forecasting remains relatively underexplored, recent empirical evidence (Xu et al. 2023) demonstrates their superior performance in capturing long-range temporal dependencies compared to hybrid CNN- LSTM and attention-based GRU models.

Given the limitations, further research could be pursued in several areas. First, exploring alternative architectures or modifications to existing models could help address issues like overfitting. Second, conducting comparative studies with other machine learning algorithms could provide a comprehensive understanding of their strengths and weaknesses. Additionally, investigating the integration of different machine learning techniques, such as attention mechanisms or hybrid methods, could improve predictive performance. Finally, integrating data types, such as technical and fundamental indicators and news, into algorithms may lead to more accurate predictions. By incorporating a wider range of inputs, algorithms have the potential to capture more comprehensive market signals and improve prediction accuracy. However, challenges associated with data integration, preprocessing, and feature engineering must be carefully addressed to utilize additional data sources effectively.

This study addresses the shortcomings of conventional models by employing data sources, proposing a hybrid dual attention-based model, and conducting comparative analyses to augment the prediction skills in financial market analysis. Depending on the type of financial data, it utilizes daily price and related features data. This research acknowledges the necessity for additional investigations on transformer-based models and enhances the accuracy of predictions by integrating diverse modules. Furthermore, it introduces a

unique methodology for forecasting movements in the financial market. The contributions of this paper are summarized as follows:

- Proposing a dual attention framework in a dual architecture enables parallel training, overcoming the limitations of traditional recurrent networks. This approach facilitates more efficient learning from long sequences, mitigates the vanishing gradient problem, and incorporates attention mechanisms to enhance learning from time-series data. The Price Attention Network (PAN) extracts complex features from price data and forecasts future prices using historical price inputs. The Nonprice Attention Network (NAN) is a parallel network that processes related financial features. The model dynamically adjusts attention weights for both price-based and non-price-based features, enabling the system to focus more on relevant inputs, thereby improving prediction quality.
- The application of developing a Multi-head Masked Self-Attention (MMSA) and Multi-head Cross-Attention (MCA) mechanism reduces unnecessary computations by avoiding the recalculation of Attention scores for all pairs of data points. This mechanism enables the model to focus on the most critical relationships, improving efficiency in capturing complex relationships and enabling the model to derive valuable insights from short and long-term dependencies. It also dynamically adjusts the contribution of various features during the training process, promoting deeper internal correlations and ensuring that more essential features are given higher priority. This approach aligns with specialized feature embeddings, effectively extracting feature representations tailored to the specific forecasting task.
- Comprehensive price, news, fundamental, and technical data integration using a parallel dual transformer-based model. This model leverages specialized feature embeddings, creating custom feature representations for price and related financial features, which are processed through specialized attention networks (PAN and NAN). The model applies different weightings to each data source to more accurately predict financial prices, ensuring a balanced and holistic forecasting approach.
- A new news impact indicator is designed to capture market sensitivity to news events due to the scarcity of news data. By integrating this indicator, the model can weigh the impact of news more effectively, allowing it to make more informed predictions in situations where news data is sparse.
- By combining Conv1D and Attention layers, the model prioritizes the most pertinent sections of the financial feature data. It leverages temporal convolutions to effectively capture local and long-range dependencies within the data.
- Positional Encoding (PE) layers define the order of daily price data, ensuring the temporal sequence is respected in the model. This allows the model to leverage the sequential nature of time-series data for more accurate predictions while maintaining the significance of time-dependent relationships.

This study proposes a dual attention-based deep-learning method to make predictions with over 26 big financial datasets. We aim to improve the generality of the proposed model by using financial data sets and comparing results with other methods. The critical question is whether attention-based deep learning can give an appropriate answer for time series forecasting, especially with financial historical data. In a different approach to financial predicting, we use various data, i.e., news, fundamental and technical features, and historical price information. Since fundamental analysis is related to financial fluctuations in a complex way, it is less common in the literature (Thakkar and Chaudhari 2021b; Leippold et al. 2022).

In the evaluation phase, we utilize Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Pinball Loss (MPL), Symmetric Mean Absolute Percentage Error (SMAPE), Mean Absolute Scaled Error (MASE) and Direction Accuracy (DA) metrics on the chosen datasets, contrasting the outcomes with those of other models.

As shown in Fig. 1, we first compile an initial set of potential candidate predictors. We then address various real financial time series data to demonstrate the proposed model's performance. We also perform comparative experiments with five state-of-the-art models and compare the results using the seven evaluation metrics mentioned.

Our study demonstrates significant improvements over state-of-the-art models across key performance metrics. Compared to existing approaches, the model achieves an average MAE reduction of 79.7%, an MSE improvement of 38.9%, and an RMSE enhancement of 74.7%, indicating superior predictive accuracy on various datasets. These results highlight the effectiveness of our approach in addressing the limitations of existing models, especially in economic crisis situations, and providing a robust foundation for further research and application in this domain.

The structure of this paper is as follows: Sectio 2 offers a literature review. Section 3 outlines the proposed method. Section 4 presents the empirical results. Finally, Sects 5 and 6 provide the discussion and conclusions.

2 Literature review

Financial markets experience significant fluctuations, making estimating whether price trends will move upward or downward challenging. Early studies suggest that financial

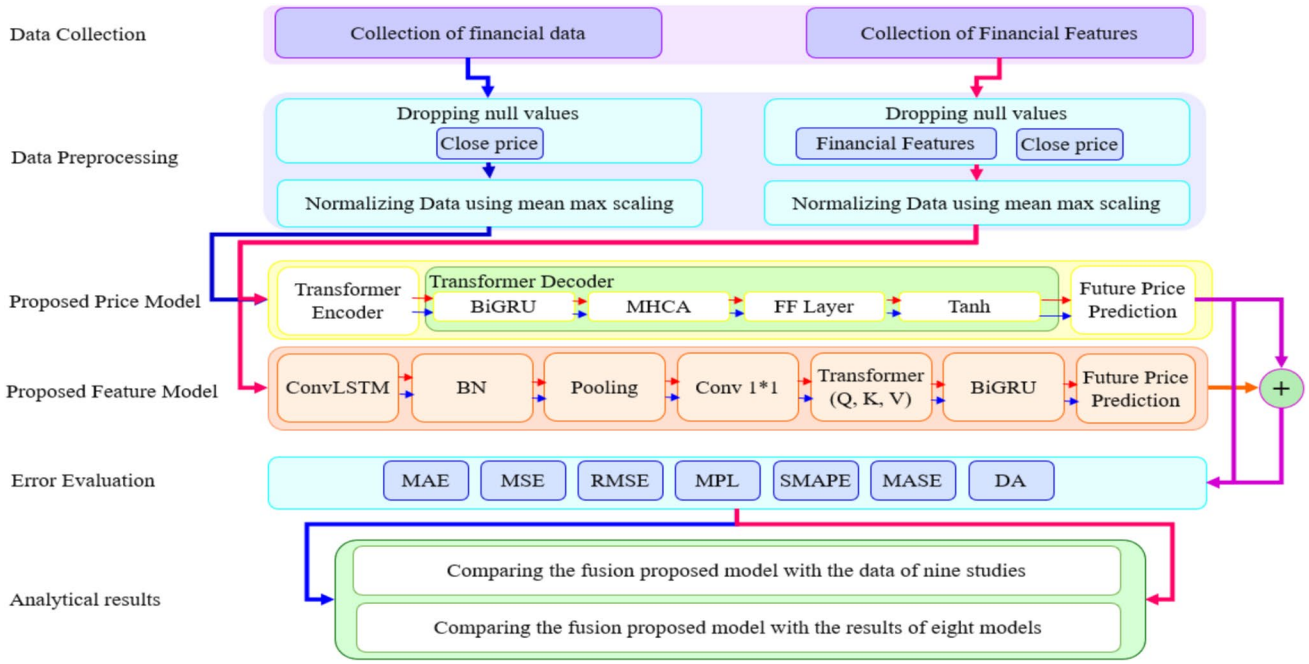


Fig. 1 Methodology flowchart

market performance resembles a random walk unless the analyst possesses new information or insights that reveal the impact of existing information on current prices (Fama 1995). Consequently, predicting financial market trends is a dynamic and complex research area increasingly reliant on big data, machine learning, and high-frequency data. Traditional models have proven useful but are often considered oversimplifying. Dichtl (Dichtl et al. 2023) highlighted the significant potential of machine learning in forecasting the financial market, noting its ability to develop predictive models that excel in feature selection and handling large volumes of data. Therefore, the key opportunities lie in applying AI and ML models to handle high-frequency data (Idrees et al. 2019).

Figure 2 illustrates the time series forecasting process, encompassing predictions for financial data. Researchers employ various machine-learning techniques to forecast financial price trends and discuss recent advances in machine-learning techniques in the financial market (Bustos and Pomares-Quimbaya 2020; Gao, et al. 2022). Abe and Nakayama (Abe and Nakayama 2018), comparing deep neural networks (DNN), concluded that DNN with more

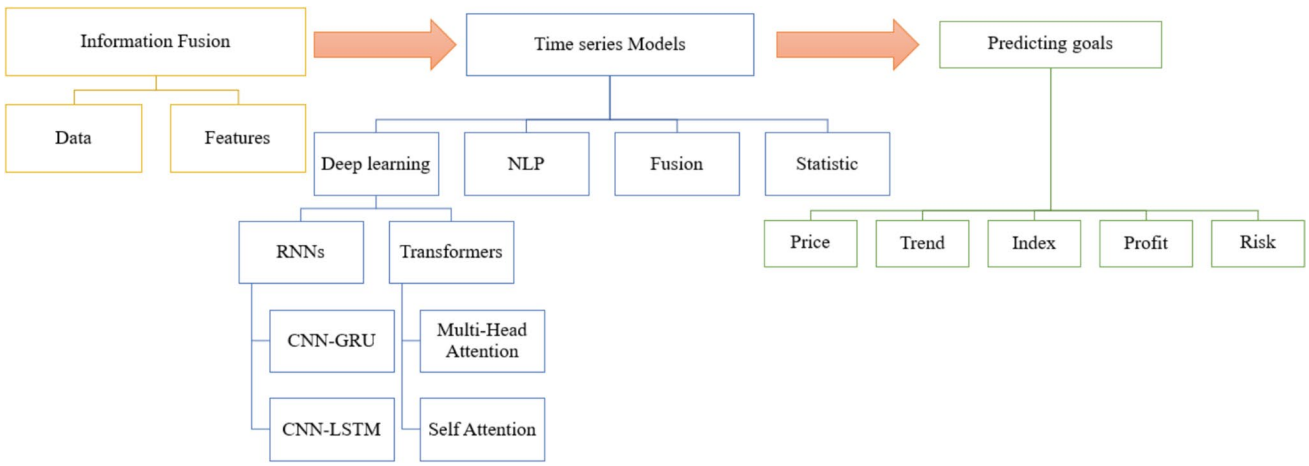


Fig. 2 A visualization of time series predicting models

layers could increase representational power by repeating nonlinear transformations that cause improvement in the prediction accuracy of the cross-sectional stock returns. DNNs provide a powerful tool for forecasting financial returns due to the ability to model complex, non-linear relationships in high-dimensional data. However, they are still under continuous enhancement due to issues like overfitting, computational intensity, and the need for adaptability to changes. Chen et al. (Chen et al. 2018) emphasized the superiority of deep learning-based models, finding that increasing the amount of data enhances predictive performance. However, the most suitable model depends on specific circumstances and requirements. Machine learning methods and hybrid models each demonstrate unique advantages and potential challenges; thus, selecting an algorithm requires careful and comprehensive consideration. There is no one-size-fits-all solution, and ongoing exploration of advanced AI methods remains essential in the rapidly evolving financial landscape.

Several studies (Leippold et al. 2022; Zhou 2019; Livieris et al. 2020) have explored various machine learning models, particularly LSTMs, highlighting the effectiveness of deep learning in predicting stock returns. These approaches shed light on risk factors, market dynamics, and the development of trading strategies, marking a paradigm shift in financial forecasting. Md, A.Q. et al. (Md et al. 2023) compared a Multi-Layer Sequential LSTM model with RNN, LSTM, CNN, MLP, and SVM models. This study indicates an encouraging perspective toward using LSTM models for financial price prediction. However, RNNs are more popular than CNNs. Shejul et al. (Shejul et al. 2023) recommended LSTM as the most effective model for financial price prediction compared to gated recurrent units (GRU) and simple RNN, aligning with the findings of Lin et al. (Lin et al. 2022). However, despite the potential, challenges such as overfitting, selection of input features, and practical implementation, such as computational complexities, must be considered.

Recent deep learning frameworks, including generative transformers inspired by optimal attention mechanisms used in natural language processing and image captioning, are also being explored for financial prediction issues. Attention-based approaches, which involve transforming the input sequence into another sequence using an encoder-decoder structure, enable the model to selectively focus on specific parts of the input sequence, thereby enhancing prediction accuracy.

The dual Transformer branches and multi-head self-attention enable the efficient processing of entire sequences in parallel, capturing long-range dependencies. The Dual Transformer's strength lies in its efficiency and direct modeling of data relationships, making it highly effective for structured prediction tasks without extensive exploration (Vaswani 2017). The Dual Transformer's strength lies in its scalability and ability to model complex, non-local relationships. It is ideal for tasks such as time-series forecasting with ample data (Devlin, et al. 2019), although it requires significant computational resources. In contrast, convolutional operations are employed in an LSTM framework to handle spatiotemporal data, preserving local spatial features and temporal dynamics (Shi 2015). Attention-based GRUs are more computationally lightweight and practical for shorter sequences or resource-constrained environments. They are limited by their sequential processing and localized attention scope, which struggle to match the Transformer's robustness over very long sequences (Cho et al. 2020). Recent studies indicate that Dual Transformers outperform Attention-based GRUs in tasks that require extensive context, although GRUs retain an edge in efficiency for simpler, structured data (Lin 2024). RL focuses on decision-making by optimizing policies through trial-and-error exploration of state-action spaces and thrives in dynamic environments that require adaptive strategies, such as those found in robotics or game playing. However, it incurs high computational costs due to its reliance on reward-driven learning. Analyses have highlighted that dual transformers excel in predictive efficiency. At the same time, RL approaches are less suited for pure sequence modeling and often underperform in representation learning compared to attention-based models. They remain unmatched for sequential decision-making tasks (Naeem et al. 2020).

Abbassimehr and Paki (Abbasimehr and Paki 2022) indicate that integrating multiplex attention and linear transformer structures is a promising approach to improving the accuracy of financial time-series prediction models. Transformer-based attention mechanisms significantly outperform traditional models in financial market prediction (Zhang et al. 2022). Wang and Zhu (Wang et al. 2022) present the deep transformer model as a highly efficient and accurate method for predicting the stock market index. Xu et al. (Xu et al. 2023) demonstrate that integrating multiplex attention and linear transformer structures is a promising approach to improving the accuracy of financial time-series prediction models.

However, the literature on the main limitations of employing transformer-based attention mechanisms for financial market prediction is still in its early stages. The challenges lie in the high computational demand and complexities in data pre-processing (Wang et al. 2022). Enhancing the reliability and generalizability of transformer models, exploring alternative feature engineering techniques, addressing data quality and quantity challenges, and improving interpretability are vital areas for future investigations of transformers (Zhang et al. 2022).

Investigating different kinds of datasets due to their features as historical trends and volume indicates that most of the conducted researches are based on the studies

of developed countries with a long history in the field of financial exchange, like the U.S. (Zhou 2019; Green et al. 2017), China (Leippold et al. 2022; Chen et al. 2018; Liu, et al. 2023; Liu, et al. 2022), and Japan (Abe and Nakayama 2018). Despite their profitability, fewer studies have been conducted in less developed countries like Mexico, Latin America, and Hongkong (Ng and Shen 2016; Dosamantes 2013; Dong et al. 2021; Nabipour et al. 2020). The point about countries with less history is that there may be challenges due to the less historical data, such as over-fitting in designing predictive machines. So, in the selected datasets, we choose high-occurrence datasets within technical features used in previous studies to demonstrate the comprehensiveness of the proposed model and a data set as a developing country that includes news, technical, fundamental, and price data history.

Several challenges have been identified based on the reviewed studies: (1) They often focus on markets with large datasets, with few studies analyzing markets in developing countries with shorter but profitable histories (Zhang et al. 2022). (2) There is a lack of comparative studies examining data with small and large histories to evaluate the results of deep attention techniques. (3) Few studies are based on combining technical, fundamental, and new datasets (Jiang 2021). (4) Transformer approaches are frequently used in natural language processing (NLP) and deep models, but there is limited experience in the financial market and time-series prediction (Wolf 2019). (5) Challenges persist, particularly around overfitting and handling non-stationary data (Sezer et al. 2020). Moreover, real-world implementation challenges and regulatory considerations require careful management. However, further research is necessary to maximize their full potential (Vaswani 2017).

The following section introduces the proposed method to achieve the desired outcome. Subsequently, we compare the proposed model with other selected models to predict financial market trends and prices. Finally, we describe the evaluation methodology. Therefore, the proposed model is designed to perform effectively on big datasets and datasets with limited historical data.

3 Proposed method

This section proposes the transformer approach for financial market prediction, a specific fusion architecture under the encoder-decoder framework, reflecting recent advancements in deep learning methodologies. Figure 3 depicts the comprehensive structure of the proposed framework. The proposed framework tackles financial prediction by leveraging a dual transformer-based architecture to forecast the future price of the financial market using two parallel modules in three steps.

In the proposed architecture, two attention-based networks, the Price-based Attention Network (**PAN**) and the Nonprice-based Attention Network (**NAN**), are integrated to create a fusion model. The PAN employs a transformer-based network to process price data as input and forecast future prices (y_{PAN}). The NAN, in contrast, leverages another attention-based network to incorporate diverse financial features and nonprice-based features, where price data functions exclusively as the label and forecasts future prices (y_{NAN}). In the final step, the outputs from the PAN and NAN are combined to enhance the accuracy of the overall Price Prediction (y_P) Fig. 4.

3.1 Price-based attention network

This section introduces the **PAN** module to predict the future price, as illustrated in Fig. 5. The proposed approach leverages an attention mechanism that dynamically adapts to fluctuations in financial data. Transformers encode the features of an object, convert them into a vectorized form, and enable them to focus on the most pertinent information in parallel. They exhibit minimal inductive biases and act scalable for complex tasks and big datasets (Parvin et al. 2023). This architecture comprises encoders, decoders, and multiple transformer modules. The encoder generates encodings of the input data information, while the decoder utilizes these encodings along with their respective contexts to produce the output sequence.

3.1.1 The encoder structure of PAN

Figure 5 presents the price history encoder with several modules. The data are collected and normalized using the maximum mean scaling function, which ensures that the features are scaled to a range between 0 and 1.

Next, the dataset is partitioned into a 70 : 30 split, with 70% allocated for training and the remaining 30% further divided into 20% for testing and 10% for validation. The encoder plays a crucial role in compressing the key information of the input sequence into a fixed-length vector across different layers. Price vectors transform each price data point into a fixed-dimensional vector representation.

PE, a technique commonly used in attention-based models like transformers, provides essential information about the positions of tokens in a sequence. In the context of price history, PE aids the model in understanding the chronological order of historical prices. Adding PE to the input embeddings allows the attention mechanism to consider both the value and the temporal relationship of historical prices.

Given a sequence of historical price data $P = \{p_1, p_2, \ldots, p_N\}$, where p_i represents the price data at the timestep i, the PE for each time step i can be computed as follows (Morita 2024):

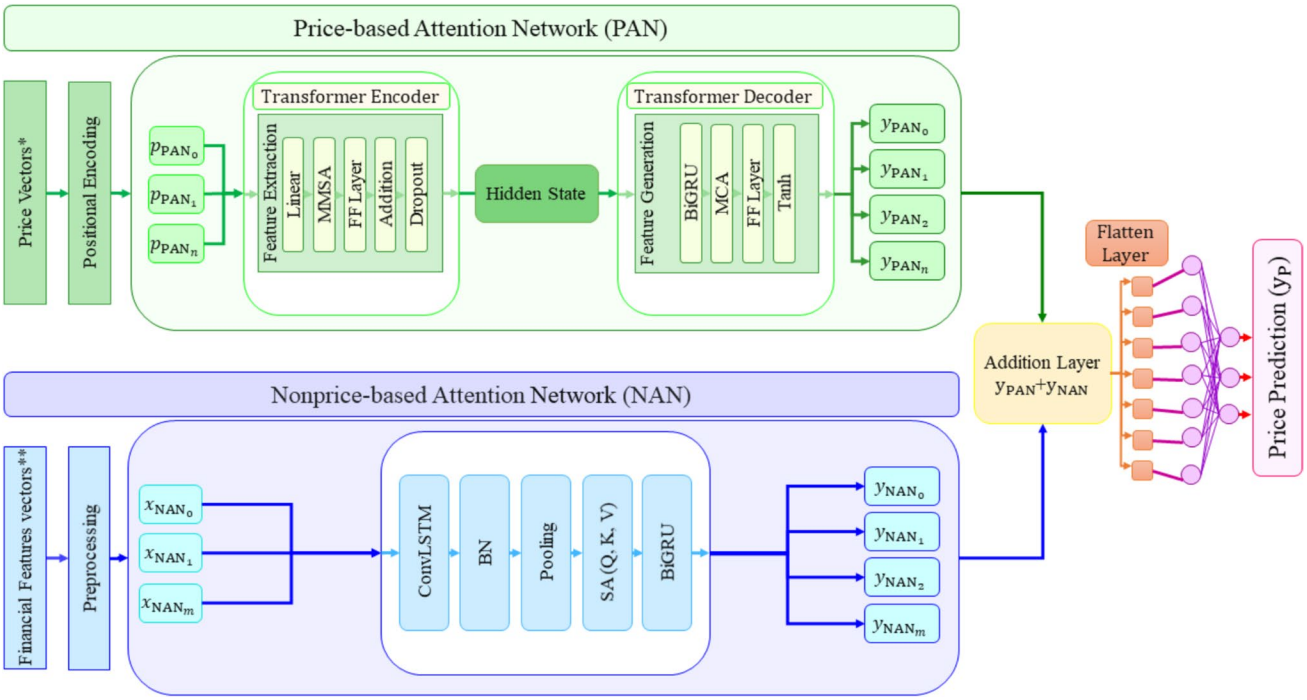


Fig. 3 The overall schema of the proposed framework.*Price Vectors: 26 datasets spanning a variety of financial indices and cryptocurrencies. **Financial Features Vectores: Related financial features, including news and technical and fundamental factors

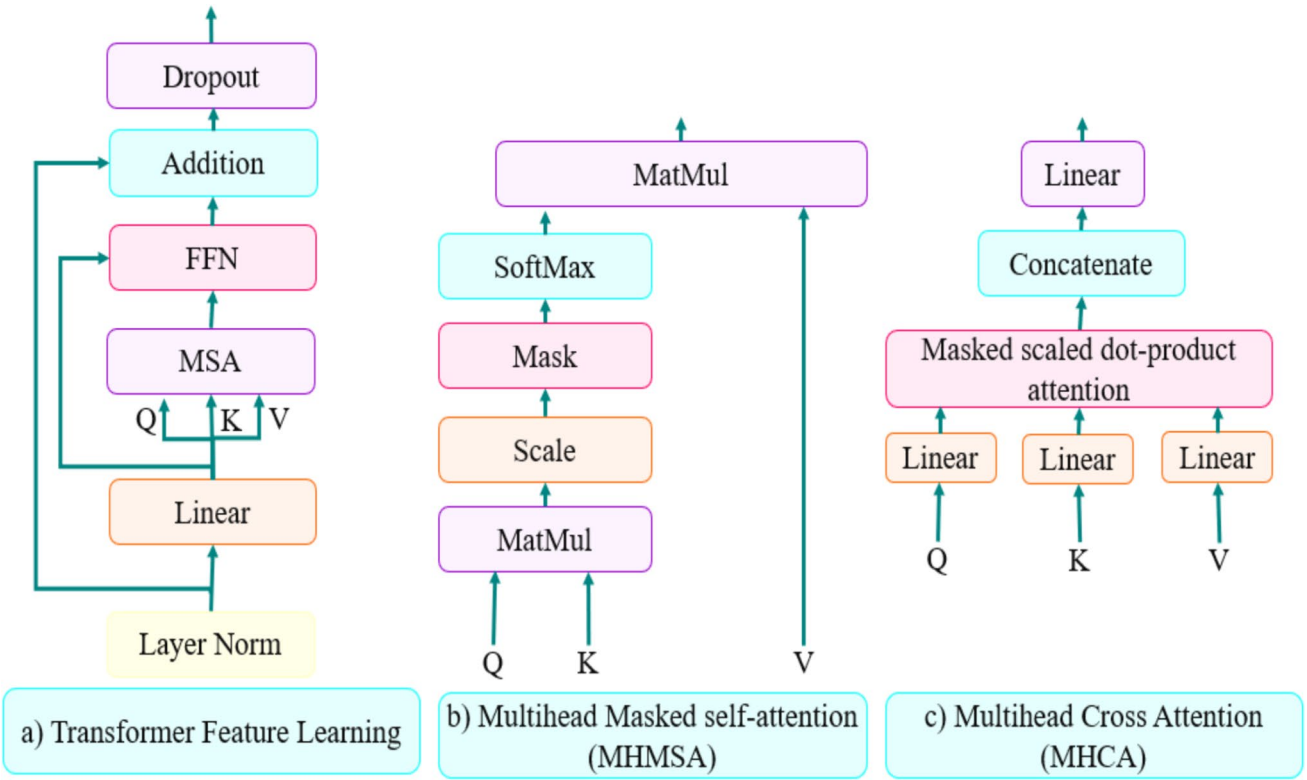


Fig. 4 The transformer structures

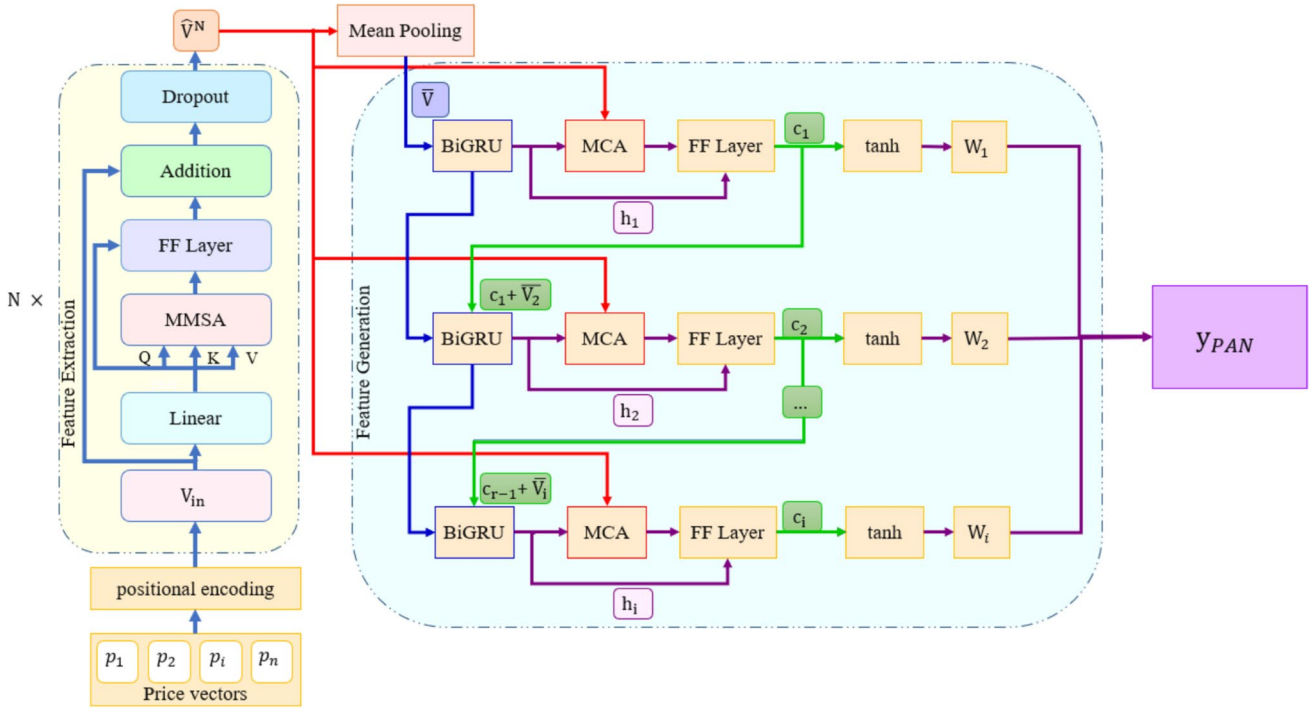


Fig. 5 Proposed model for the price history analysis

$$PE_{(pos,2i)} = sin(\ ^{pos}\Big/_{10000^{\ 2i/d_{model}}}) \tag{1}$$

$$PE_{(pos,2i+1)} = cos(\ ^{pos}\Big/_{10000^{\ 2i/d_{model}}}) \tag{2}$$

where $PE_{(pos,2i)}$ represents the sine PE for the $2i - th$ dimension, while $PE_{(pos,2i+1)}$ represents the cosine PE for the $(2i + 1)$-th dimension. pos denotes the token's position within the sequence, ranging from 1 to N, where N is the sequence length, d is the dimensionality of the model.i is the dimension's index within the positional encoding.

Concerning the transfer network structure for learning time series features, the process unfolds: Initially, normalization is applied to the time series to derive the transformer. Then, via a linear projection module through various Self-Attention (SA) blocks, matrices Q, K, and V are obtained by performing three distinct linear projections on the input features. The linear block consists of fully connected layers that transform the input data into a higher-level representation, facilitating extracting relevant features from the input sequence. Following the computation of similarity values between Q and K, the SoftMax function calculates attention weights denoted as W. Finally, V is weighted by these attention weights using the equation (Vaswani 2017):

$$Attention(K,\ Q,\ V) = Softmax\ (QK^T/\sqrt{d})V \tag{3}$$

where K, Q and V represent the matrix of keys, queries, and values, respectively, and d is the dimensionality of the query/key vectors. The proposed model's structure is presented in the following subsections.

The SA mechanism enables the model to focus on relevant parts of an input sequence by computing a weighted sum of the input elements, where the weights are based on the similarity between each component and a query vector. Figure 4 depicts the transformer architecture and diverse SA blocks.

An advanced technique of SA is multi-head attention (MA), which allows a neural network to learn multiple representations of an input sequence, such as prices or indicators. Among the transformer model variations, MA performs best, underscoring the importance of the attention mechanism and positional information in capturing financial market patterns (Devlin et al. 2019). This technique facilitates extracting and combining relevant information from numeric data features, such as historical prices, to make better predictions. Additionally, it can handle variable-length input sequences and missing or noisy data by computing several attention functions in parallel, each with different projections of the query, key, and value matrices.

Masked attention is a crucial technique that ensures the model does not consider future information unavailable during prediction by assigning zero attention weights to future positions. MMSA combines the principles of MA and masked attention, enhancing the effectiveness of financial

market prediction models. MMSA creates a versatile model capable of capturing intricate patterns and dependencies in financial data (Wang 2021). This approach facilitates robust learning from data features while mitigating the risks of overfitting or underfitting. Illustrated in Fig. 4. b, the MMSA mechanism for head h processes a set of L queries, keys, and values as detailed by Nicolson and Paliwal (2020):

$$\text{Attention}\left(\mathcal{Q}_{\text{h}}, \text{K}_{\text{h}}, \text{V}_{\text{h}}\right) = \text{softmax}\left(\text{M} + \frac{\mathcal{Q}_{\text{h}}\text{K}_{\text{h}}^{\text{T}}}{\sqrt{\text{d}_{\text{k}}}}\right)\text{V}_{\text{h}} \quad (4)$$

where the dot product of $\mathcal{Q}_h$ and the transposition of K_h presents the unnormalized weights of the attention mechanism obtained from the similarity matrix. These weights are then scaled by $\frac{1}{\sqrt{d_k}}$, and a mask $M \in R^{L,L}$ is applied to filter out similarities involving future frames, preserving causality.

In this process, the queries, keys, and values are projected h times via linear transformations to dimensions d_Q, d_K and d_V, respectively. Subsequently, attention functions are computed in parallel, yielding the next d_V output values as described as follows (Vaswani 2017; Abbasimehr and Paki 2022):

$$\begin{aligned} &Multihead\ (Q,K,V) = Concat\left(head_1, \ldots, head_h\right) \\ &W_o head_i = Attention\ (QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \quad (5)$$

where $i = 1, \ldots, h$ and QW_i^Q, KW_i^K, VW_i^V presents weights.

Next, the softmax activation function normalizes each row of the sequence similarity matrix into a probability distribution.

Accordingly, the MMSA block enables the model to attend to different positions within the input sequence and capture dependencies over varying ranges, focusing on relevant information while preventing information leakage from future timestamps through masking.

Each architecture layer integrates multiple MMSA mechanisms alongside a Feed-Forward (FF) module, facilitating adaptation to various multi-modal representations. FF layers apply fully connected layers with non-linear activation functions to capture complex relationships between historical prices. The FF layer modules simplify the structure for prediction (Liu et al. 2022). Then, the addition layer combines the outputs of the MMSA layer with the FF layer output, $FF(Q, \widehat{V})$. This mitigates significant changes in the data distribution caused by transformations applied in the FF layer.

To address overfitting and computational complexity, the outputs of the normalized and dropout modules are employed to enhance the generalization of the proposed architecture. Dropout randomly drops units from the network during training, preventing the model from relying too much on specific features or relationships. This entire process is repeated N times for each layer of the neural network, as shown $\widehat{\text{V}}^{\text{N}}$. The addition layer ensures the model learns meaningful representations while mitigating drastic changes in the data distribution. Additionally, incorporating dropout helps regularize the model, reducing overfitting and complexity. The time series encoder process is formulated as follows:

$$PO = PE\ (PriceVector) \quad (6)$$

$$V_{In} = Layer\ Norm(O) \quad (7)$$

$$Q, K, V = Linears(V_{In}) \quad (8)$$

$$\widehat{V} = MMSA(Q, K, V) \quad (9)$$

$$\widehat{V}^N = Drop\ Out\ (V_{In} + FF(Q, \widehat{V})) \quad (10)$$

where PO represents the positional encoding of price vectors. V_{In} represents the normalized form of PO. $\widehat{V}$ captures meaningful relationships within the sequence through the MMSA mechanism. $FF(Q, \widehat{V})$ depicts the FF layer output. $\widehat{V}^N$ combines and provides a robust representation of the prediction task.

The mean pooling layer takes the mean of n representations,$\widehat{V}^N$, to transform n matrices into vectors, resulting in the generated $\overline{v}$, which is then fed into a decoder Structure.

This adaptive mechanism significantly improves the model's ability to discern essential patterns, leading to more accurate predictions.

3.1.2 The decoder structure of PAN

The proposed decoder module takes the vectors of the encoder layer. At each timestep, the decoder computes a weighted sum of the encoder output and represents relevant historical data information. Then, it combines the attention output with its own hidden state and input feature to produce the predicted financial price. The decoder can have multiple layers and attention heads, allowing for learning different aspects of the input and output sequences and capturing more complex patterns and dependencies (Vaswani 2017). The proposed decoder layers include BiGRU, Multi-head Cross-Attention (MCA), FF layer, and the hyperbolic tangent (Tanh).

GRUs, a specialized variant of LSTM, have garnered increasing popularity for tasks involving sequential learning. Unlike LSTMs, GRUs employ a dual-gate mechanism—update and reset—to regulate information flow without requiring a separate memory cell for internal data manipulations (Chung, et al. 2015). Moreover, GRUs boast more computational efficiency and less complexity and possess bidirectional capabilities, enabling them to assimilate information from past and future contexts (Cho et al. 2020;

Schuster and Paliwal 1997). The bidirectional aspect is further enhanced with BiGRU, offering versatility in capturing intricate relationships and discerning complex patterns and dependencies inherent in sequential data, resulting in more precise predictions. BiGRU exhibits enhanced adaptability to fluctuations and uncertainties inherent in input sequences. Leveraging bidirectional information enables it to adjust to diverse data distributions and input sequences, thereby fostering the development of more resilient and generalizable models. This holistic comprehension proves advantageous in scenarios where contextual nuances significantly impact outcomes, such as discerning trends and patterns within historical financial price data (Chung, et al. 2015).

At step i, BiGRU takes the mean-pooled feature and the price vector $(\overline{v}_i + ci)$, as following functions:

$$h_i^f = GRU^F(\overline{v}_i + ci, h_{i-1}^f) \tag{11}$$

$$h_i^b = GRU^b(\overline{v}_i + ci, h_{i-1}^b) \tag{12}$$

$$h_i = concat\ (h_i^f, h_i^b) \tag{13}$$

where h_i^f and h_i^b are the hidden states at the i-th time step of the forward and backward BiGRU. c_i is the context vector at the time step i, which is combined with $\overline{v}_i$. $\mathrm{h_i}$ denotes combining the forward and backward hidden states, allowing the model to capture information from past and future contexts, enhancing its ability to predict effectively, and maintaining continuity and coherence in the decoding process (Schuster and Paliwal 1997).

The decoder architecture utilizes BiGRU with MCA (Fig. 4. c), followed by an FF layer with a tanh activation function and linear transformation tailored for decoding price data.

MCA is a mechanism used in neural network architectures, particularly transformers, for processing sequential data and effectively capturing complex relationships, dependencies, and aspects of the input data. MCA comprises four key components: the attention mechanism, MA, and Cross-Attention. Cross-attention extends the idea of attention from SA to attending across different sequences. Cross-attention allows the model to take information from multiple input sequences or modalities simultaneously, facilitates richer interactions, and improves task performance that requires understanding relationships between different inputs (Vaswani 2017).

The decoder employs an FF layer to transform the concatenated input and context vector. A Tanh activation function is applied to introduce non-linearity to the output of the FF layer. Finally, a linear transformation layer is utilized to map the output to the desired dimension suitable for predicting financial market data as the following equations:

$$\overline{v}_i = MCA(h_i, \overline{v}_n(i)) \tag{14}$$

$$c_i = FF(h_i + \overline{v}_i) \tag{15}$$

$$w_i = Tanh(c_i) \tag{16}$$

$$y_{PAN} = softmax(w_i) \tag{17}$$

where $\overline{v}_i$ presents the output of the MCA mechanism for the time step i. c_i is the context vector at the time step i, combined with $\overline{v}_i$. w_i applies the hyperbolic tangent ($Tanh$) activation function to the context vector and represents the output of the $Tanh$ activation function. y_{PAN} presents the predicted price at the time step i.

3.2 Nonprice-based Attention Network (NAN)

The proposed hybrid architecture predicts financial prices by processing feature vectors, as depicted in Fig. 6, through a series of layers, including ConvLSTM, attention mechanisms, BiGRU, and fully connected layers. Due to the financial features, this network represents a sequential learning mechanism for price prediction Table 1.

The financial features vector, comprising prices and the financial attributes outlined in Table 13, is initially transformed into a normalized vector through preprocessing and normalization. The model accepts a sequence of financial feature vectors:

$$X = \{x_0, x_1, \ldots, x_N\}, x_n \in \mathbb{R}^d \tag{18}$$

where x_n represents the feature vector at time n and d is the dimension of the feature vector.

This preprocessing stage, detailed in Sect. 3–1-1, improves the quality and utility of the feature vector while ensuring its compatibility with subsequent analysis. Specifically, it prepares the feature vector for representation learning within the ConvLSTM block, which performs the feature extraction in the next step.

The ability of LSTMs to process long-term dependencies, combined with the attention mechanism's capacity to focus on essential aspects, enhances the effectiveness of ConvLSTM) Bahdanau et al. 2014). As a robust tool for sequential data analysis, ConvLSTM increases the number of features in sequential data, expanding the feature space to specified dimensions such as 32, 64, or 128. This facilitates the creation of more informative and comprehensive data representations (Shi 2015). Unlike traditional LSTM layers, ConvLSTM incorporates convolutional operations instead of internal matrix multiplications, enabling the model to capture intricate spatiotemporal patterns inherent in large sequential datasets, such as time-series data (Lee and Kim

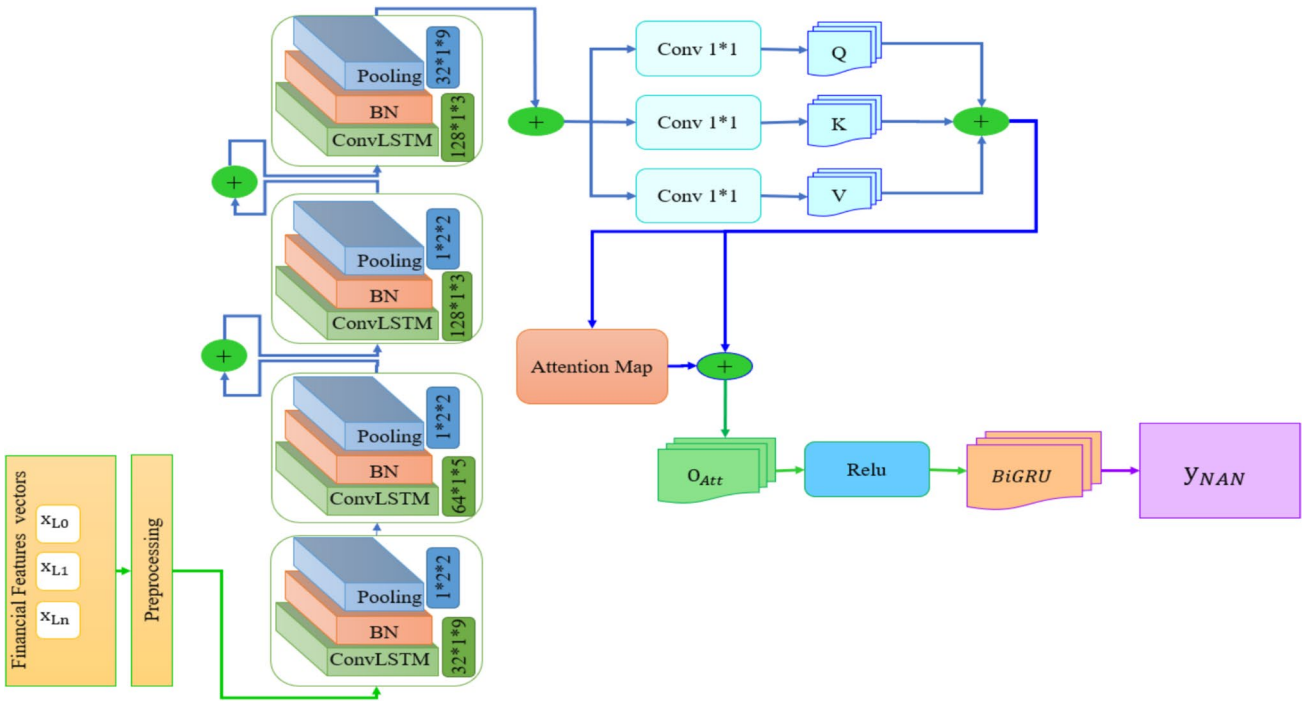


Fig. 6 Proposed architecture for the financial features analysis

2020). By combining the spatial learning capabilities of convolutional layers with the temporal modeling prowess of LSTM layers, ConvLSTM is particularly well-suited for processing spatiotemporal data like financial market sequences.

The proposed architecture utilizes four ConvLSTM layers, each followed by Batch Normalization (BN) layers to stabilize training and a pooling layer to refine features for final prediction. The first layer encodes temporal information, whereas the second and third layers capture spatial features during convolution operations. Thus, ConvLSTM integrating convolutional operations enables the extraction of features with both spatial and temporal dimensions (Mustaqeem and Kwon 2020). The ConvLSTM layer is utilized to optimize the sequence and to keep the sequential information in the internal state.

Following the ConvLSTM output, the next block, BN, prevents changes in the data distribution (Ioffe and Szegedy 2015). BN ensures that the data within each batch is normalized, thereby reducing variance in the data distribution. Normalizing the data within each batch stabilizes the learning process and facilitates better convergence during training.

Next, the output enters the pooling layer to reduce the size of the representation and the number of parameters and computations, decreasing computational complexity, which helps prevent overfitting and reduces computational costs (Krizhevsky et al. 2017). Each pooling layer is connected to the output of the next layer so that past information can be used for the future. The output of the pooling layer is then fed into an SA with an activation function, which further processes the features extracted by the ConvLSTM. Therefore, network optimization is facilitated, and data distribution does not change.

$$H_i, C_i = ConvLSTM(X_i, H_{i-1}, C_{i-1}) \tag{19}$$

where H_i is the hidden state, C_i is the cell state and C_i applies convolutional operations on temporal sequences.

$$H_{BN} = BN(H_i) \tag{20}$$

where H_{BN} normalizes the output.

$$H_{Po} = Pooling(H_{BN}) \tag{21}$$

where H_{Po} reduces spatial dimension and is the output of the ConvLSTM layer after BN and pooling.

$$F = \sum_{i=1}^{L=4} H_{Po} \tag{22}$$

where L is the number of ConvLSTM layers. F aggregates the outputs from multiple ConvLSTM layers to combine multi-scale spatiotemporal features.

Concatenating the outputs of different layers creates a more significant feature representation, preventing data alteration within each block or layer of the neural network. Since convolutional operations can alter the data, concatenating the outputs from different layers at the beginning and

Table 1 The details of the selected financial data from the previous studies

	Reference	Datasets	Data History	Evaluation Metrics
1	Wang and Zhu (2023)	SZSE Composite Index SSE Composite Index NYSE Composite Index	January 4, 2011 to August 31, 2021	RMSE MAPE MAE MSPE
2	Quek et al. (2022)	Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), Cardano (ADA), Polkadot (DOT1), Bitcoin Cash (BCH), Stellar (XLM), Binance Coin (BNB)	the earliest obtainable date to October 31, 2021	MAE RMSE
3	Rathee et al. (2023)	BTC, ETH, LTC	2010 to 2021	MSE RMSE
4	Lahmiri et al. (2022)	Aragon, Basic Attention Token, Bitcoin Cash, Blocknet, Binance Coin, Bitcoin, Civic, DigixDAO, district0x, Dogecoin, EOS, Ethereum, Gnosis, ICON, Lisk, Litecoin, MaidSafeCoin, MCO, MonaCoin, Nano, NavCoin, Neblio, OmiseGO, Stratis, Substratum, Tether, NEM, and XRP	November 16, 2018 to November 16, 2019	RMSE MAE
5	Tao et al. (2024)	Shanghai Composite Index (SH) SZSE Hang Seng Index (HSI)	July 1, 1991 to June 30, 2020 July 31, 1991 to May 31, 2022	MAE RMSE MAPE R2-score
6	Lee and Kim (2020)	Standard & Poor's 500 Index (S&P500) KOSPI200	Oct. 24, 2002, to Feb.10, 2017 Jul. 22, 2002 to Jan.24, 2017	MSE MAPE MAE
7	Wang et al. (2022)	Shanghai and Shenzhen 300 Index (CSI 300) (S&P 500) Nikkei 225 Index (N225)	Jan 1, 2010 to Dec 31, 2020	MAE MSE MAPE
8	Nejad and Ebadzadeh (2024)	APPLE, Microsoft, Amazon, Nvidia, Google, Tesla	2010 to 2020	MAE RMSE R2-score
9	Cheng et al. (2024)	BTC	2017 to 2022	MSE MAE
10	Proposed by writers for short-term analysis	TSM (covers 350 share stocks traded)	February 2021 to 9 November 2021	MAE, MSE, RMSE, MPL, SMAPE, MASE, and DA

propagating them to the end helps maintain consistency and stability in the data distribution throughout the network. By utilizing an architecture featuring branches extending from the beginning to the end, we ensure minimal alteration of the data, maintaining consistency across various processing stages. This preserves crucial information and prevents substantial changes in the data distribution, thereby bolstering the effectiveness and robustness of the model. In the following, the network's output as a vector will enter the SA transformer block to be converted into three matrices of K, Q, and V as mentioned in 3.1.1.

A 1×1 convolutional layer is utilized to diminish the computational complexity upon entry into the attention block as a pre-processing step. The purpose of using a 1×1 convolution at this stage is to reduce the dimensionality of the input data, enabling more efficient processing within the subsequent attention block while capturing essential patterns and relationships in the data.

$$Q = Conv_{1\times1}(F, W_q) \tag{23}$$

$$K = Conv_{1\times1}(F, W_k) \tag{24}$$

$$V = Conv_{1\times1}(F, W_v) \tag{25}$$

where W_q, W_k, and W_v are the learnable weight matrices of the 1×1 convolutional layer for a given aggregated feature map F.

$$O_{ATT} = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{26}$$

where O_{ATT} generate the attention-weighted output for the BiGRU blocks to capture temporal dependencies.

A stacked BiGRU network, a streamlined variant of sequence learning modules, dynamically adjusts the weights in response to the extracted information and enables the detection of long-term and bidirectional dependencies and the temporal dynamics inherent within data segments (Zhou et al. 2022).

$$H_{BiGRU} = BiGRU(O_{ATT}) \tag{27}$$

where H_{BiGRU} represents the concatenated hidden states from both forward and backward GRU passes.

The BiGRU output is passed through a *ReLU* activation function and a fully connected layer to generate the final prediction:

$$y_{NAN} = ReLU(W_{OUT}.H_{BiGRU}+b_{out}) \tag{28}$$

where W_{OUT} and b_{out} are the weights and biases of the fully connected layer. y_{NAN} is the predicted output of NAN.

As a result, a network combining ConvLSTM, SA, and BiGRUs analyzes historical financial feature data and predicts prices. The resulting output is utilized for further analysis and integrated with the attention-driven model described in Section 3.1.

3.3 Combining local and global modules for final prediction

The first model proposes a transformer encoder-decoder architecture for price data, capturing temporal patterns in historical price data. The second architecture employs a ConvLSTM and transformer-BiGRU for financial features. In the combination layer, the outputs from the two architectures are concatenated to combine the information learned from both approaches. Additional fully connected layers can be added to the concatenated outputs to process the combined information further and extract relevant features.

The flattening layer is applied to flatten the output from the additional layers before feeding it into the final prediction layer. The final prediction layer predicts based on the combined information learned from both models.

$$c = Add(\left[y_{PAN}, y_{NAN}\right]) \tag{29}$$

$$a = FullyConectec(c) \tag{30}$$

$$f = Flatten(a) \tag{31}$$

$$y_P = Softmax(f) \tag{32}$$

where y_{PAN} is the output from the transformer encoder-decoder model applied to the price data. y_{NAN} is the output from the ConvLSTM and transformer-BiGRU model applied to the financial features. *c* concatenate the outputs from both models. a is the additional fully connected layer to process the concatenated outputs. f flatten the output from the additional layers. The final prediction layer, as y_P, uses the flattened output to predict the future of the pointed financial dataset.

3.4 Architectural distinctions from existing attention models

Recent studies have highlighted the applicability and strengths of transformer-based architectures in financial time series prediction and their potential to become indispensable tools in financial forecasting (Souto and Moradi 2024). Unlike standard dual-attention frameworks applied in domains such as natural language processing or computer vision, where parallel attention mechanisms are typically used to focus on different spatial or contextual representations(Parvin et al. 2023), the proposed model introduces a domain-specific dual-attention structure designed to address the unique complexities of financial time series prediction. Notably, transformer variations such as the TST (Wang et al. 2022) and Informer (Ren et al. 2023) do not explicitly differentiate attention roles across fundamentally heterogeneous data streams.

In contrast, the proposed architecture combines MHCA and MHMSA to model inter-series relationships and temporal dependencies jointly at multiple levels of resolution. The MHCA module enables the model to capture interactions between distinct input representations, such as trend-based versus volatility features. In contrast, the MHMSA module is specifically designed to extract patterns across various temporal scales, enabling the model to detect both abrupt changes and persistent trends.

Moreover, while TST and Informer rely on a global attention mechanism to process univariate or multivariate inputs uniformly, our model is architecturally modular and designed to adapt to financial data's non-stationary, high-variance nature. This architectural distinction provides an edge in handling structural breaks and market volatility, improving generalizability across asset classes (e.g., equities versus cryptocurrencies).

3.5 Training strategy

The training process integrates key practices to ensure optimal performance and generalization. The model is compiled using the Adam optimizer, which is well-suited

for handling sparse gradients and large-scale data (Kingma et al. 2014). The Adam optimizer offers efficient gradient optimization by adaptively adjusting learning rates for each parameter. The loss function selected is the **MSE**, which is commonly used for regression problems. MSE minimizes the squared difference between the actual and predicted price values.

The Early Stopping technique prevents overfitting and enhances generalization (Prechelt 2002). This method monitored the validation MSE during training and stopped the process if the metric does not improve for consecutive epochs. By doing so, computational efficiency is improved, and the risk of overfitting is mitigated. The model is trained using two types of input data: price and financial features. The price is expressed as an optimization problem as follows (Bishop and Nasrabadi 2006):

$$\theta = -\arg min_{\theta} \frac{1}{n} \sum_{t=1}^{T} (y_i - f_{\theta}(X, P))^2 \tag{33}$$

where f_{θ} represents the hybrid model parameterized by θ. P denotes a set of price data. X is a collection of time-series features derived from financial indicators.

3.6 Measures of forecasting performance

We employ seven error metrics to assess the forecasting performance of the proposed models for predicting financial historical data, since the focus of the analysis is regression (Hyndman and Koehler 2006; Moosa 2014; Armstrong and Collopy 1992; Koenker 2005). Lower values of MAE, MSE, RMSE, MPL, SMAPE, and MASE, along with higher values of DA, signify superior performance.

1) MAE measures the average absolute difference between the predicted values and the actual values:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \left| Y_i - \hat{Y}_i \right| \tag{34}$$

2) MSE measures the average of the squares of the differences between predicted and actual values.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{35}$$

3) RMSE is the square root of the MSE and provides the standard deviation of the prediction errors.

$$RMSE = \sqrt{MSE} \tag{36}$$

4) MPL measures the accuracy of quantile regression models by penalizing over-predictions and under-predictions asymmetrically based on a specified quantile.

$$MPL = \frac{1}{n} \sum_{i=1}^{n} [0.9\left(Y_i - \hat{Y}_i\right) + 0.1(\hat{Y}_i - Y_i)] \tag{37}$$

5) SMAPE, a symmetric variant of MAPE, exhibits reduced sensitivity to outliers.

$$SMAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{\left| Y_i - \hat{Y}_i \right|}{(|Y_i| + \left|\hat{Y}_i\right|)/2} \tag{38}$$

6) MASE evaluates the accuracy of a forecasting model. It compares the mean absolute error (MAE) of a given model with the MAE of a forecast model (Hyndman and Koehler 2006).

$$MASE = \frac{1}{n} \sum_{i=1}^{n} \frac{\left| Y_i - \hat{Y}_i \right|}{\frac{1}{n-1} \sum_{i=2}^{n} \left| Y_i - \hat{Y}_{i-1} \right|} \tag{39}$$

7) DA measures the proportion of correctly predicted directional movements (e.g., increase or decrease) compared to the actual movements.

$$DA = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} * 100 \tag{40}$$

where Y_i represents the actual value, $\hat{Y}_i$ represents the predicted value and n is the total number of data points.

These error metrics provide comprehensive insights into the forecasting model's accuracy and reliability, enabling robust evaluation and comparison.

3.7 Algorithm design

Algorithm 1, presents the pseudocode for implementing a model tailored to predict financial markets by leveraging price data and financial features. A step-by-step explanation of the algorithm is as follows:

In step 1, the algorithm begins with initializing parameters and matrices, as detailed in line 1. These include the input price data (P), financial features (X), and other hyperparameters listed in Table 2. This initialization sets the foundation for the data flow and processing within the model.
In Step 2, the price and financial features are normalized and segmented into multi-step sequences in lines 2 to 4. This step prepares the input data for subsequent feature extraction and analysis.

In step 3, the PE function is applied to the price vectors to incorporate positional information into the data. Lines 5 to 8 depict extracting relevant features and patterns, capturing temporal dependencies and complex relationships. Based on the processed price data, this step predicts the future price for a predefined number of time steps.
In Step 4, from lines 9 to 11, a linear transformation maps the extracted features from the financial features. This network focuses on learning patterns, leveraging attention mechanisms and sequential layers to enhance prediction accuracy.
In Step 5, the predictions generated from the PAN and NAN phases are integrated to produce a combined output. This step ensures that price data and financial features contribute holistically to the final forecast.
Finally, in step 6, the model completes the price prediction process, generating comprehensive forecasts for the financial market's future behavior. This step ensures that the predictions are aligned with the defined objective of accurately forecasting market trends.

Algorithm 1 Pseudocode for the proposed framework

	Inputs: Price data, non-price data, news **Outputs**: Predicted Future Price
Start:	
	# ***Step 1:*** *Initialize hyperparameters:*
1:	Initializing the hyperparameters of the proposed method according to Table 2
	# ***Step 2:*** *Preparing Data*
2:	Normalization of price and non-price data
3:	Multi-step division over price data
4:	Multi-step division over the non-price data
	# ***Step 3****: PAN prediction according to Fig.5*
5:	Positional-Encoding over the price data using Eq. (1), Eq. (2) and Eq. (6)
6:	Feature extraction from price data using Eq. (7) to Eq. (10)
7:	Feature generation from price data using Eq. (11) to Eq. (13)
8:	Price prediction using Eq. (14) to Eq. (17)
	# ***Step 4:*** NAN *prediction according to Fig.6*
9:	Feature extraction from non-price data using Eq. (18) to Eq. (22), and Table 13
10:	Apply self-attention block using Eq. (23) to Eq. (26)
11:	Sequential prediction using Eq. (27) and Eq. (28)
	# ***Step 5:*** *Combined prediction according to Fig.3*
12:	Combining the outputs of the PAN and NAN networks using Eq. (29) to Eq. (32)
	# ***Step 6:*** *Prediction*
13:	Training the proposed model
14:	Optimizing using Eq. (33)
15:	Output: Future Price Prediction using Eq. (32)
End	

Table 2 Constants and hyperparameters and their values and definitions

	Constants & hyperparameters	Value	Definition
1	Epochs	120	The number of training epochs
2	timestep	25	The number of time steps considered for each input sequence
3	n_future	3	The number of future time steps to predict
4	d_model	10	The dimensionality of the model's output space
5	numheads	6	The number of attention heads in the MA mechanism
6	keydim	5	The dimensionality of the key vectors in the attention mechanism
7	valuedim	5	The dimensionality of the value vectors in the attention mechanism
8	N	4	The number of encoder and decoder layers in the transformer model
9	maximum_position_encoding	100	The maximum position index in the positional encoding
10	batchsize	5	The number of samples processed in each batch during training
11	trainsize	0.7	The proportion of the dataset used for training
12	h	4	The number of projections of the queries, keys, and values in the Feature Generation process of the encoder's structure

4 Experimental results and discussion

This section introduces the price datasets, related features, and a series of experiments conducted to demonstrate the effectiveness of the proposed model. As the CRISP model (Wirth et al. 2000), first, we outline the experimental setup. Next, we assess the framework's performance by comparing it to state-of-the-art methods in financial prediction. Finally, we comprehensively analyze the proposed model to gain deeper insights into its behavior and effectiveness.

4.1 Datasets and related features

The selected financial datasets include stock symbols, indexes, and digital currencies based on their historical data. To prove the proposed model, the experiments encompass 26 datasets spanning a variety of financial indices and cryptocurrencies, including SZSE, SSE, NYSE Composite Index, BTC, ETH, LTC, ADA, DOT1, BCH, XLM, BNB, BAT, DigixDAO, Dogecoin, Ethereum, MonaCoin, Nano, NavCoin, Tether, XRP, SH, HSI, S&P500, KOSPI200, CSI 300, N225, and TSM, as shown in Table 1 and also their proposed features in Table 13. These datasets include long-term and short-term datasets.

Data sets comprise extensive financial datasets referenced in nine references for analyzing and predicting future prices. They are chosen due to their shared comparative criteria for evaluation. The datasets are sourced for download from finance.yahoo.com and investing.com. For discussion on the short-term data and characteristics, we chose TSM data. We obtained daily liquid stock symbols from the bourseview.com database.

We choose some indicators, with their related functions shown in Table 13. There are many indicators in the literature review and business experiences. For instance, Light et al. (Light et al. 2017) use 26 variables associated with classified prominent asset pricing anomalies. Zhong et al. (Zhong and Enke 2017) use 60 financial and economic indicators. Abe et al. (Abe and Nakayama 2018) use 25 indicators. Zhou (Zhou 2019) utilizes 15 indicators sourced from Light et al. (2017). Leippold et al. (Leippold et al. 2022) use 94 features in 3 daily, monthly, and yearly groups. Our collection includes 34 characteristics compressed into 12 groups: momentum, profitability, performance, financial risk, market risk, relative valuation, momentum, sales, shareholders'performance, liquidity, exchange rate, index, and news.

Features were selected for each dataset for two key reasons. First, they were chosen based on availability, as they are commonly used in financial datasets referenced in Table 1. Second, given that data from emerging markets and less developed countries tend to be short-term and limited in volume, it is essential to define additional features to compensate for this limitation. For example, indicators like profitability are utilized for all datasets, while news, namely news impact, is employed for the TSM dataset.

The news impact has been defined based on daily news. During the specified period, news is reviewed, classified, and scored based on the presence of keywords as follows:

$$NE(t) = \left\{ne_0, ne_1, \ldots, ne_N\right\}, ne_n \in NE(t) \tag{41}$$

where $NE(t)$ represents the set of news items during the specified period t. Each ne_n is a value associated with a specific news item and is evaluated for the presence of positive, negative, or neutral keywords.

For keyword scoring, two sets of keywords are defined as follows:

$$\begin{aligned} &Positive\ keywords: K^+ = \\ &\{\varepsilon increase, \varepsilon\varepsilon up, \varepsilon\varepsilon buy, \varepsilon\varepsilon support, \varepsilon\varepsilon most, \varepsilon \\ &\varepsilon growth, \varepsilon\varepsilon positive, \varepsilon\varepsilon good\varepsilon\} \end{aligned} \tag{42}$$

$$\begin{aligned} &Negative\ keywords: K^- = \\ &\{\varepsilon decrease, \varepsilon\varepsilon descent, \varepsilon\varepsilon sell, \varepsilon\varepsilon lowest, \varepsilon\varepsilon less, \varepsilon \\ &\varepsilon negative, \varepsilon\varepsilon drop, \varepsilon\varepsilon bad\varepsilon\} \end{aligned} \tag{43}$$

The score S_n for each news item ne is computed as:

$$S_n = \begin{cases} +1\ if\ ne\ contains\ any\ k \in K^+ \\ -1\ if\ ne\ contains\ any\ k \in K^- \\ 0\ otherwise\ (neutral\ score) \end{cases} \tag{44}$$

For a given symbol s, the aggregated news impact score $I_S(t)$ during the period t is the algebraic sum of scores for all related news items $NE_S(t)$:

$$I_S(t) = \sum_{ne \in NE_S(t)} S_n \tag{45}$$

where $NE_S(t) \subseteq NE(t)$ represents news items specific to the symbol s.

This approach allows the impact of news to be featured in predictions alongside other features.

4.2 Experimental settings

We utilized the scikit-learn Python library, supplemented by additional supportive libraries, to develop and optimize our proposed framework for predictive analysis. The experimental dataset comprises historical financial data. The financial data was segmented into distinct time intervals, with each segment sequentially inputted into the model to extract discriminative features and facilitate accurate recognition. We utilized a Google Cloud Platform, Google Colab, equipped with a Tesla V100-SXM2 16 GB GPU to execute the model,

complemented by 25 GB of RAM and 100 GB of storage space. The proposed model was executed on this system, which has 2 CPU cores, resulting in an average CPU usage of 3.5% and a RAM usage of 18.1%. During runtime, 10.38 GB of RAM was available. The GPU used for training was an NVIDIA Tesla T4, which had a total memory of 15.36 GB. Of these, only 5.2% were used.

We employed a trial-and-error approach for hyperparameter tuning, adjusting one parameter at a time while keeping the others fixed. This enabled the model to identify the optimal configuration and achieve the best performance in the final test. For instance, we varied the D-model value from 1 to 50 in increments of 5, keeping all other parameters constant, and recorded the optimal result corresponding to the D-model setting. This method enabled the identification of optimal settings, resulting in the best outcomes in the final evaluation. The optimal parameter values were determined and documented in Table 2.

4.3 Quantitative Evaluation

The experimental evaluation is listed in Tables 3, 4, 5, and 6 against state-of-the-art models for large and small datasets. Table 3 displays the evaluation metrics for the proposed model, alongside the data from the referenced studies. It is important to note that we have selected the optimal results from each referenced study to ensure meaningful comparisons. Table 3 also emphasizes the improvements of the proposed framework in contrast to nine other studies across more than 26 datasets. On average, the proposed model achieves approximately 79.7% improvement in MAE, 74.7% in RMSE, and 38.9% in MSE compared to the referenced baseline models.

Tables 4 and 5 provide experimental evaluations of the proposed fusion framework on BTC and TSM using a sample of large and small datasets. The best results for each evaluation metric are highlighted in bold, indicating that the proposed model outperforms most other studies across the seven metrics considered. Experimental evaluations demonstrate improvements of 0.052, 0.066, 0.073, 0.026, 11.347, 3.764, and 1.393, compared to the average of five state-of-the-art models, based on the following evaluation measures: MAE, MSE, RMSE, MPL, MAPE, MASE, and DA, respectively, on the BTC dataset, which is a large dataset. Similarly, on the small dataset, the improvements are 0.24, 0.14, 0.33, 0.12, 107.4, 10.55, and 1.90 for the same evaluation measures.

Table 6 compares the computational efficiency of the proposed fusion model against five state-of-the-art models. Specifically, the table reports each method's runtime (in seconds), CPU usage, number of CPU cores utilized, RAM consumption, and GPU memory usage. The proposed model demonstrates the lowest runtime and RAM consumption while maintaining minimal GPU memory usage. These results suggest that the proposed architecture is effective in performance and computationally efficient, making it suitable for practical deployment in resource-constrained environments.

The proposed fusion model consistently exhibits substantial performance improvements across various financial datasets, encompassing large-scale (BTC) and small-scale (TSM) scenarios. Specifically, for the BTC dataset, the model achieves reductions of 72.2% in MAE, 92.5% in MSE, 71.5% in RMSE, 72.1% in MPL, 78.9% in SMAPE, and 67.0% in MASE, along with a 2.88% increase in DA. On the TSM dataset, the model further underscores its robustness, yielding reductions of 98.6% in MAE, 99.95% in MSE, 97.9% in RMSE, 98.6% in MPL, 59.9% in SMAPE, and 74.7% in MASE, with an additional 5.2% improvement in DA. These results underscore the model's strong generalization ability and predictive effectiveness across market conditions. The consistent enhancements observed across evaluation metrics and datasets indicate that the proposed architecture outperforms existing models in most scenarios. Notably, the proposed framework surpasses a wide range of previously published models, including LSTM-based architectures (Livieris et al. 2020; Lee and Kim 2020; Wang and Zhu 2023; Rathee et al. 2023; Cheng et al. 2024; Semenoglou et al. 2023; Aldhyani and Alzahrani 2022), transformer-based and MA models (Wang et al. 2022; Xu et al. 2023; Abbasimehr and Paki 2022; Tao et al. 2024; Li et al. 2023; Yue and Ma 2023), fuzzy time series, genetic algorithm-based models (Quek et al. 2022), SVR (Lahmiri et al. 2022), and generative adversarial networks (Nejad and Ebadzadeh 2024).

4.4 Convergence behavior

As illustrated in Fig. 7, the proposed architecture is compared with five competing models to evaluate the convergence behavior in the BTC dataset. Learning curves are plotted based on the number of epochs where the learning process achieved improved results.

The convergence study confirms that the proposed architecture exhibits stable convergence behavior, effectively learning underlying patterns in the financial data. Furthermore, the evaluation of the validation dataset yielded promising results, demonstrating the model's capability to make accurate predictions on real-world data. This observation denotes that the proposed model requires fewer iterations to achieve stable outputs, underscoring its efficiency in training.

4.5 Ablation study

Tables 7 and 8 present a systematic analysis of the impact of the price and feature networks within the proposed

Table 3 Addressing the analysis of the proposed model based on over 26 financial data sets in the period of each reference

	Method	Datasets	Model	MAE	MSE	RMSE	MPL	SMAPE	MASE	DA
1	GA-LSTM (Wang and Zhu 2023)	SZSE	P.M	0.03160	0.00161	0.04016	0.01580	17.40331	2.77590	50.972
			REF	0.16721	-	0.22414	-	-	-	-
		SSE	P.M	0.02299	0.00079	0.02819	0.01149	26.08176	3.06496	49.855
			REF	0.3357	-	0.47240	-	-	-	-
		NYSE	P.M	0.04451	0.00320	0.05653	0.02225	28.12475	6.43942	47.996
			REF	0.15200	-	0.22371	-	-	-	-
2	FTS (Quek et al. 2022)	BTC	P.M	0.00163	0.00001	0.00340	0.00081	40.93076	7.92346	49.505
		ETH LTC ADA DOT1 BCH XLM BNB	REF	0.039	-	0.042	-	-	-	-
3	CNN-BiLSTM (Rathee et al. 2023)	BTC	P.M	0.12777	0.05554	0.23567	0.06389	34.12100	13.28743	51.026
			REF	-	0.033	0.181	-	-	-	-
		ETH	P.M	0.23905	0.10054	0.31708	0.11952	58.61785	9.94758	48.204
			REF	-	0.042	0.200	-	-	-	-
		LTC	P.M	0.06033	0.00902	0.09499	0.03017	17.12360	3.30568	50.776
			REF	-	0.0304	0.174	-	-	-	-
4	SVR-BO (Lahmiri et al. 2022) 29 cryptocurrencies[a]	Daily-RBF kernel	P.M	**0.00055**	**0.00001**	**0.00067**	**0.00027**	40.73262	11.89807	34.331
			REF	0.2013	-	0.2111	-	-	-	-
		Weekly- polynomial kernel	P.M	0.00071	0.00001	0.00083	0.00035	50.19614	12.53305	38.028
			REF	0.1009	-	0.1045	-	-	-	-
5	SDTP (Tao et al. 2024)	SH	P.M	0.03284	0.00440	0.06636	0.01642	22.14310	2.32754	48.397
			REF	0.21731	-	0.31604	-	-	-	-
		SZSE	P.M	0.03075	0.00275	0.052498	0.01537	25.27902	3.08500	3.085
			REF	0.12904	-	0.175059	-	-	-	-
		HSI	P.M	0.02973	0.00157	0.03966	0.01487	24.34735	5.61298	49.958
			REF	0.25602	-	0.345411	-	-	-	-
6	ConvLSTM (Lee and Kim 2020)	S&P500	P.M	0.05979	0.00503	0.07095	0.02990	8.00424	8.75305	51.394
			REF	0.10329	0.31965	-	-	-	-	-
		KOSPI200	P.M	0.11209	0.01863	0.13651	0.05605	53.40271	15.94960	48.820
			REF	0.1735	0.0706	-	-	-	-	-
7	TST (Wang et al. 2022)	CSI 300	P.M	0.02862	0.00138	0.03717	0.01431	17.62333	2.59829	50.098
			REF	0.0641	0.0079	-	-	-	-	-
		S&P 500	P.M	0.03998	0.00217	0.04658	0.01999	42.39668	9.65025	**52.579**
			REF	0.0814	0.0145	-	-	-	-	-
		Nikkei 225	P.M	0.03302	0.00170	0.04118	0.01651	45.98903	5.88521	49.955
			REF	0.0471	0.0043	-	-	-	-	-
		Hang Seng	P.M	0.03147	0.00168	0.04093	0.01573	14.48316	2.25748	51.249
			REF	0.0881	0.0138	-	-	-	-	-

Table 3 (continued)

	Method	Datasets	Model	MAE	MSE	RMSE	MPL	SMAPE	MASE	DA
8	DRAGAN (Nejad and Ebadzadeh 2024)	APPLE	P.M	0.01242	0.00023	0.01513	0.00621	31.22428	6.95586	49.906
			REF	0.68	-	0.92	-	-	-	-
		Microsoft	P.M	0.00675	0.00007	0.00841	0.00337	34.12501	4.19282	49.580
			REF	1.21	-	1.80	-	-	-	-
		Amazon	P.M	0.00697	0.00007	0.00850	0.00348	38.18333	6.88476	49.668
			REF	1.10	-	1.71	-	-	-	-
		Nvidia	P.M	0.00250	0.00001	0.00343	0.00125	29.12304	4.00353	48.952
			REF	1.21	-	1.68	-	-	-	-
		Google	P.M	0.03361	0.00164	0.04045	0.01681	48.64976	13.42229	51.048
			REF	0.81	-	1.09	-	-	-	-
		Tesla	P.M	0.01339	0.00029	0.01702	0.00669	51.40414	42.92406	50.547
			REF	0.57	-	0.80	-	-	-	-
9	Cheng et al. 2024)	BTC	P.M	0.01991	0.00084	0.02903	0.00996	**3.03324**	**1.85436**	50.253
			REF	0.61709	-	0.38	-	-	-	-
	Improvement %		P.M	79.7%	38.9%	74.7%				

[a]29 cryptocurrencies including Aragon, Basic Attention Token, Bitcoin Cash, Blocknet, Binance Coin, Bitcoin, Civic, DigixDAO, district0x, Dogecoin, EOS, Ethereum, Gnosis, ICON, Lisk, Litecoin, MaidSafeCoin, MCO, MonaCoin, Nano, NavCoin, Neblio, OmiseGO, Stratis, Substratum, Tether, NEM, and XRP

Table 4 Comparative analysis on large dataset-BTC

	References	MAE	MSE	RMSE	MPL	SMAPE	MASE	DA
1	CNN–LSTM (Livieris et al. 2020)	0.04644	0.00446	0.06681	0.02322	8.75327	3.64838	50.20957
2	Multiplex Attention and Transformer (Xu et al. 2023)	0.08284	0.00915	0.09568	0.04142	17.65432	6.50765	49.96588
3	CNN (Semenoglou et al. 2023)	0.08401	0.02345	0.15314	0.04201	16.85837	6.59991	49.58098
4	TST(Wang et al. 2022)	0.07947	0.00982	0.09912	0.03973	16.46095	6.24300	50.02437
5	LSTM (Aldhyani and Alzahrani 2022)	0.06484	0.00891	0.09442	0.03242	12.17489	5.09401	49.30305
6	Proposed Fusion Model	**0.01991**	**0.00084**	**0.02903**	**0.00996**	**3.03324**	**1.85436**	**51.25332**
7	Improvement %	72.2%	92.5%	71.5%	72.1%	78.9%	67.0%	2.88%

Table 5 Comparative analysis of the small dataset- TSM

	Method	MAE	MSE	RMSE	MPL	SMAPE	MASE	DA
1	CNN–LSTM (Livieris et al. 2020)	0.30910	0.17193	0.41464	0.15454	61.08895	18.70394	45.62876
2	Multiplex Attention and Transformer (Xu et al. 2023)	0.28485	0.12392	0.35203	0.14243	59.84300	17.23723	46.436453
3	CNN (Semenoglou et al. 2023)	0.31216	0.17177	0.41445	0.15608	62.80954	18.88950	43.73465
4	TST (Wang et al. 2022)	0.41456	0.34081	0.58379	0.20728	99.99999	25.08591	46.61018
5	LSTM (Aldhyani and Alzahrani 2022)	0.30709	0.16992	0.41221	0.15354	63.37483	18.58299	43.79532
6	Proposed Fusion Model	**0.00443***	**0.00009**	**0.00932**	**0.00226**	**27.84679**	**4.984789**	**47.61253**
7	Improvement %	98.6%	99.95%	97.86%	98.61%	59.89%	74.70%	5.24%

* Bold values represent the best performances

Table 6 Comparison of Computational Resource Utilization and Runtime

	References	Run Time (Second)	CPU Usage%	CPU Count	Used RAM (GB)	GPU Total Memory (GB)	GPU Used Memory (MB)
1	CNN–LSTM (Livieris et al. 2020)	100.27	35.8	2	5.69	15.36	404.0
2	Multiplex Attention and Transformer (Xu et al. 2023)	61.15	3.0	2	6.13	15.36	404.0
3	CNN (Semenoglou et al. 2023)	39.59	3.5	2	4.73	15.36	392.0
4	TST (Wang et al. 2022)	100.19	52.0	2	10.60	15.36	430.0
5	LSTM (Aldhyani and Alzahrani 2022)	78.13	5.0	2	5.89	15.36	406.0
6	Proposed Fusion Model	32.54	3.5	2	2.29	15.36	378.0

architecture for large and small datasets. The proposed architecture addresses the challenge of financial prediction by employing a dual transformer-based architecture, which enables the forecasting of future prices in the financial market through two parallel networks. These networks are specifically designed to capture distinct aspects of the data, focusing on price and feature data. We aim to discern the contributions of these networks to the model's overall predictive performance through an ablation study.

Experiment 1 concentrates on the price model. The results demonstrated superior predictive performance compared to nine other models across various financial datasets, as shown in Table 3**.** Specifically, the results are presented in the first row of Tables 7 and 8. Experiment 2, in the second row, shifted the focus to the feature phase. Results show few improvements in prediction accuracy.

Experiment 3 involves integrating two networks to enhance the accuracy of price prediction. Therefore, the proposed fusion framework outperforms other models by leveraging both modules for financial price prediction, as evidenced by the results presented in the third row of Tables 7 and 8. Integrating price and feature models enhances

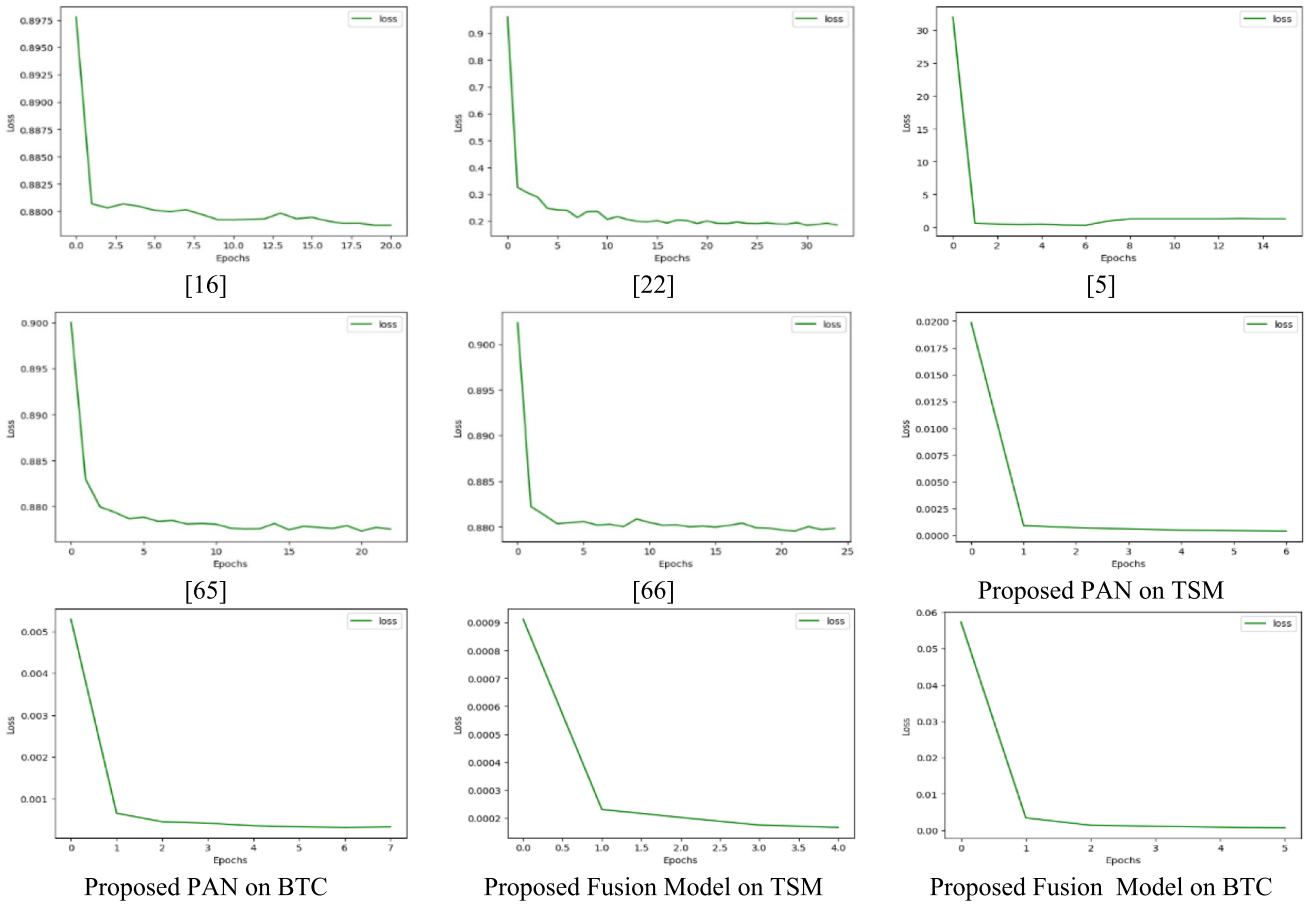


Fig. 7 Convergence behavior of the proposed model across diverse financial datasets compared to five related models

Table 7 Ablation study regarding the application of PAN and NAN for large data set BTC

Models	MAE	MSE	RMSE	MPL	SMAPE	MASE	DA
Proposed Feature Model	0.043871	0.00364	0.05950	0.02043	3.41776	3.29640	50.02627
Proposed Price Model	0.025631	0.00112	0.12100	0.01570	3.12100	3.71702	50.02536
Proposed Fusion Model	**0.01991**	**0.00084**	**0.02903**	**0.00996**	**3.03324**	**1.85436**	**50.25332**

Table 8 Ablation study regarding the application of PAN and NAN for large data set TSM

Models	MAE	MSE	RMSE	MPL	SMAPE	MASE	DA
Proposed Feature Model	0.00859	0.00022	0.01506	0.00429	45.04097	9.65984	43.85289
Proposed Price Model	0.00986	0.00019	0.01384	0.00493	96.38827	11.09258	44.78192
Proposed Fusion Model	**0.00443**	**0.00009**	**0.00932**	**0.00226**	**27.84679**	**4.98478**	**47.61253**

prediction performance by leveraging two parallel attention-based networks.

The ablation study highlights the significance of PAN and NAN in improving predictive performance. Integrating these networks in the fusion framework presents superior results, underscoring the complementary nature of price data and related financial features in prediction tasks.

4.6 Sensitivity analysis

To evaluate the sensitivity of the time step, n_future, and d_model parameters, varying value effects are examined on the model's performance across BTC datasets. Figure 8 illustrates the results, covering the seven evaluation metrics

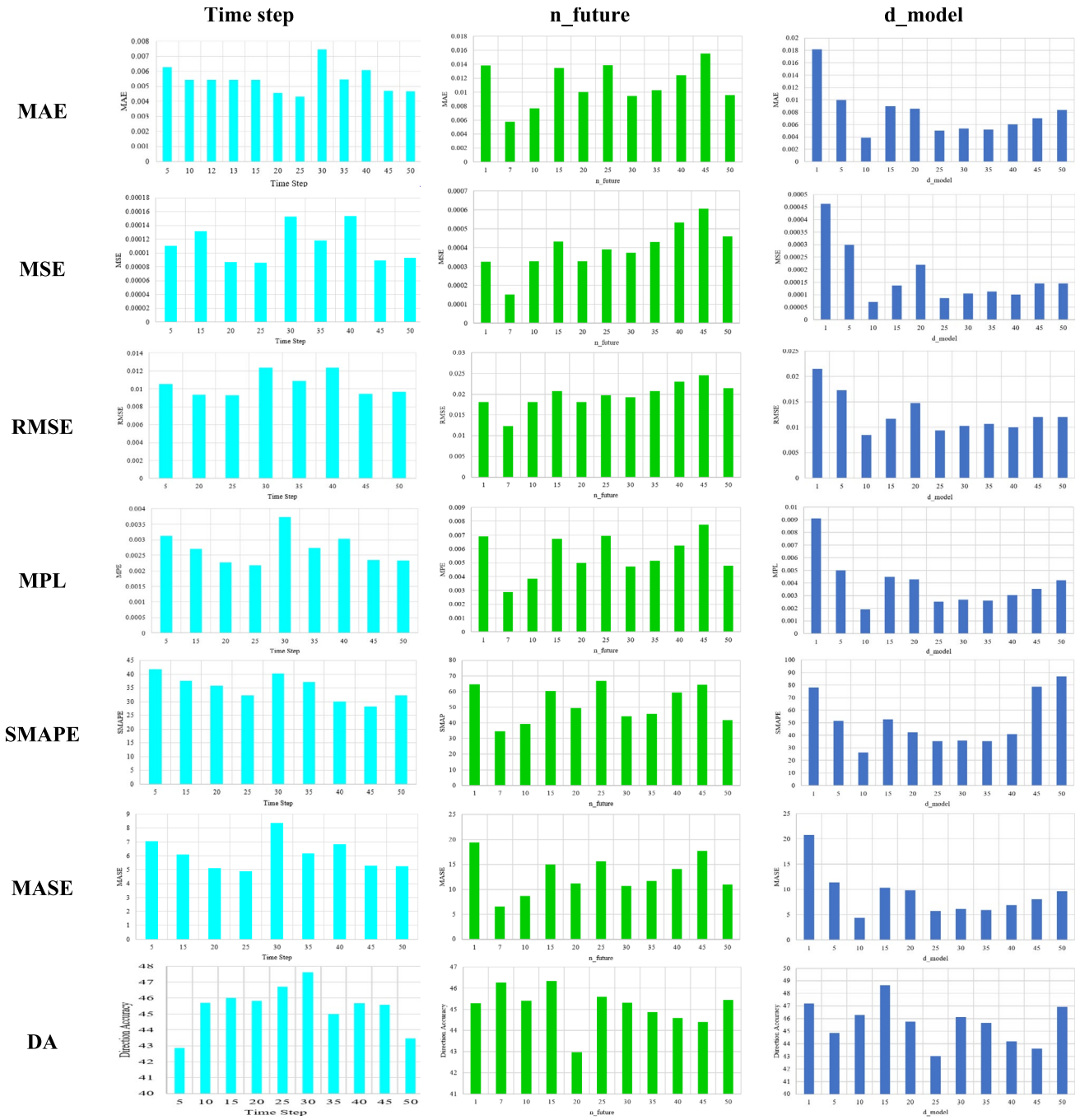


Fig. 8 Addressing the sensitivity of Time step, n_future, and d_model parameters to optimize the performance of the financial data prediction due to the seven evaluation metrics

mentioned in Sections 3–5. Performance metrics vary across different variable values as follows:

The timesteps 20 and 25 yield lower error metrics (MAE, MSE, RMSE, and MPL), indicating improved model performance. However, there is a slight increase in SMAPE. Timestep 20 exhibits higher metrics regarding DA, suggesting a more favorable model performance.

The analysis of the n_future parameter reveals insights into the model's performance across various evaluation metrics. As n_future increases, the error metrics tend to rise, indicating a decrease in accuracy for longer prediction horizons. SMAPE and MASE show fluctuations across different n_future values, with SMAPE increasing for larger values, indicating higher relative errors for longer horizons. MASE

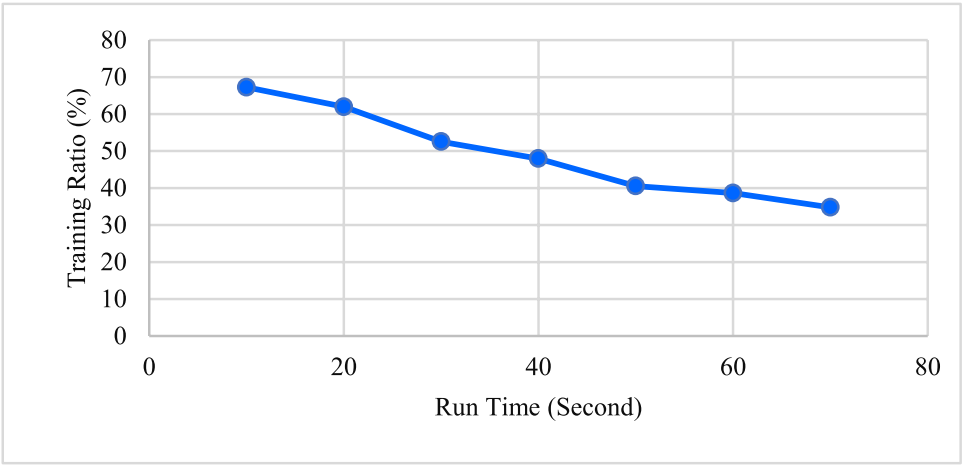
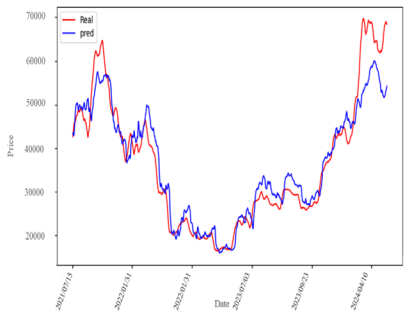
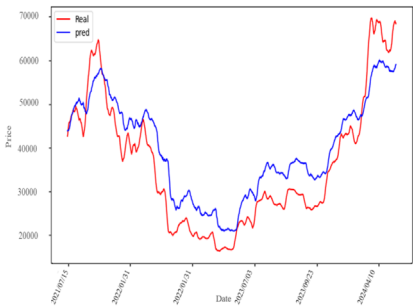


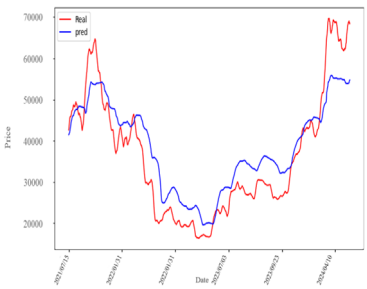
Fig. 9 Scalability of the proposed model by varying the training size and measuring the training duration over the BTC dataset



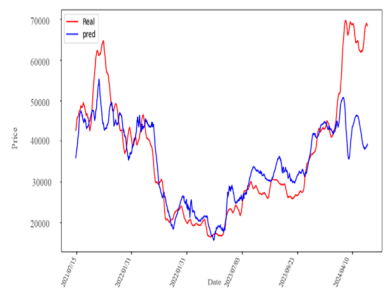
BTC Prediction based on model in [16]



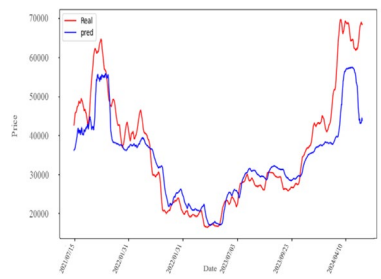
BTC Prediction based on model in [22]



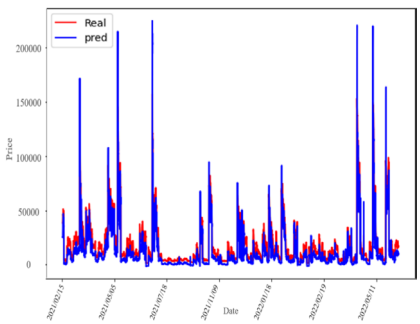
BTC Prediction based on model in [5]



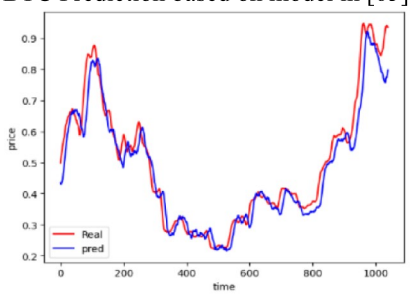
BTC Prediction based on model in [65]



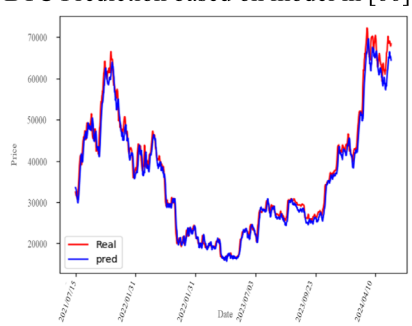
BTC Prediction based on model in [66]



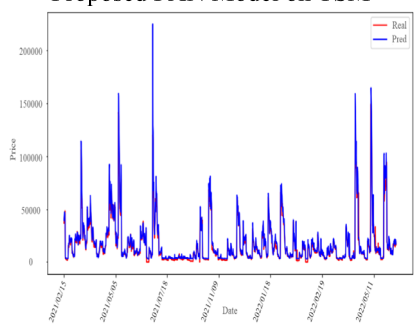
Proposed PAN Model on TSM



Proposed PAN Model on BTC



Proposed Fusion Model on BTC



Proposed Fusion Model on TSM

Fig. 10 Qualitative performance to evaluate the performance of the financial data prediction

exhibits a mixed trend. DA demonstrates consistent performance across n_future values.

Lower values of d_model, such as 7 and 15, demonstrate relatively lower MAE and RMSE, indicating better prediction accuracy. However, as d_model increases, specific metrics like SMAPE and MPL also increase, suggesting a potential trade-off. These values demonstrate low prediction errors while maintaining stability in other evaluation criteria, such as SMAPE and Direction Accuracy Fig. 9.

4.7 Qualitative performance

Figure 10 illustrates a visual comparison of the qualitative performance of the proposed models with those of the other five state-of-the-art models mentioned in the BTC dataset (Table 3). The model thoroughly understands temporal dependencies and market dynamics by leveraging transformer-based architectures with attention mechanisms. It maintains consistent performance levels, highlighting its adaptability and generalization capabilities. This versatility makes it a valuable tool for various financial forecasting tasks. Despite the complexity of the financial data and the large-scale datasets used for training, the model remains computationally efficient and scalable. This scalability ensures that the model can handle increasing volumes of data without compromising performance or requiring significant computational resources.

4.8 Robustness analysis

To evaluate the proposed model's robustness against real-world uncertainties, we employ Gaussian noise with a standard deviation of 0.1 to the datasets and analyze its impact on performance. Table 8 presents the results under noisy conditions, which are compared with the original (noise-free) results from Tables 4 and 5 and the best-performing baseline models from previous studies.

In the large dataset (BTC, for example), the proposed model's MAE increased from 0.01991 (without noise) to 0.02740 (with noise), indicating a slight degradation in performance. However, the RMSE remained low at 0.03512, confirming the model's stability. In the small dataset (TSM), the MAE increased slightly from 0.00443 to 0.00590, demonstrating that the model maintains high accuracy even in noise.

Despite the introduction of noise, the proposed fusion model continues to outperform the best baseline models. In the BTC dataset, the best baseline model (Livieris et al. 2020) achieves an MAE of 0.04644, which is 69% higher than the noisy version of the proposed model (0.02740). Similarly, the RMSE of the proposed model under noisy conditions (0.03512) remains significantly lower than the best baseline result (0.06681). In the TSM dataset, the best baseline model (Xu et al. 2023) has an MAE of 0.28485, over 48 times higher than the proposed model under noise (0.00590). This demonstrates the superior generalization capability of the proposed approach.

The overall robustness analysis indicates that the proposed model remains the best-performing approach across all metrics, even in noise. The minimal error increase confirms the model's robustness and resistance to input perturbations. Unlike traditional models, which suffer significant accuracy drops under noisy conditions, the proposed hybrid Transformer-based model demonstrates consistent predictive stability.

The results highlight the proposed fusion model's strong adaptability and reliability. Despite market noise, the model yields superior forecasting accuracy, outperforming both baseline methods and its noise-free version. This robustness is crucial for real-world financial applications, where market fluctuations and data imperfections are inevitable Table 9.

4.9 Scalability analysis

In this section, we address the proposed model's scalability from two perspectives: increasing the size of various financial datasets, varying the training set size, and measuring the training duration.

In the first step, we investigate the scalability of the proposed model by expanding the size of various financial datasets, using data from January 2011 to March 2025. Since some of the datasets include cryptocurrencies with different starting dates, the start date for each selected dataset is specified in Table 10.

In the long-term evaluation, the proposed model achieves an average MAE of 0.01923, MSE of 0.00076, RMSE of 0.02623, MPE of 0.00973, SMAPE of 6.67937, MASE of 2.5700, and DA of 50.4651 across the 26 datasets. The proposed model performs well on low-volatility datasets, such as Tether (MAE of 0.00648, SMAPE of 1.05079), as well as on small-scale datasets, like MonaCoin (MAE of 0.00259, MPE of 0.00130). It also performs well on major indices,

Table 9 Results over the noisy data

Data sets	MAE	MSE	RMSE	MPL	SMAPE	MASE	DA
on large dataset-BTC	0.02740	0.00123	0.03512	0.01370	6.65272	2.97676	50.29625
on small dataset-TSM	0.00590	0.00014	0.01224	0.00295	37.83134	8.28271	46.64656

Table 10 Results for large-scale financial datasets

Data Sets	Start Date	MAE	MSE	RMSE	MPL	SMAPE	MASE	DA
ADA	2018/04/26	0.01154	0.00037	0.01947	0.00577	6.81941	2.02133	50.28748
Amazon	2011/01/03	0.02462	0.00090	0.03008	0.01231	4.14839	2.48703	50.40548
APPLE	2011/01/03	0.01652	0.00045	0.02129	0.00826	2.56753	1.94756	50.24953
BNB	2017/11/09	0.02290	0.00112	0.03358	0.01145	3.84262	1.98601	51.30651
BTC	2012/01/01	0.01809	0.00059	0.02439	0.00904	4.91886	2.28483	50.44125
CSI 300	2011/01/04	0.02332	0.00096	0.03099	0.01166	4.43303	2.45782	50.87153
Dogecoin	2017/06/03	0.01634	0.00068	0.02607	0.00817	8.36338	2.32767	50.37182
ETH	2016/03/10	0.02080	0.00090	0.03016	0.01040	4.32113	1.88200	50.27027
Gnosis	2018/02/02	0.02052	0.00088	0.02979	0.01026	5.78862	2.07394	50.20613
Google	2011/01/03	0.01716	0.00051	0.02270	0.00858	2.81996	1.81208	52.08983
HSI	2011/01/03	0.02887	0.00134	0.03662	0.01443	13.2863	2.14768	50.06369
KOSPI	2011/01/03	0.02477	0.00111	0.03332	0.01239	3.81440	1.90476	50.43034
LTC	2017/12/15	0.01525	0.00041	0.02037	0.00762	9.49300	2.40844	53.18671
Microsoft	2011/01/03	0.02791	0.00095	0.03270	0.01395	4.17063	3.26641	51.65315
MonaCoin	2017/12/08	0.00259	0.00001	0.00377	0.00130	26.7046	3.65052	51.51007
Nano	2018/01/18	0.00864	0.00011	0.01093	0.00432	20.9861	4.58556	50.02137
Nikkei 225	2011/01/04	0.02674	0.00102	0.03200	0.01337	4.03423	3.11251	50.69065
Nvidia	2011/01/03	0.01767	0.00090	0.03014	0.00884	5.23162	2.10913	51.40361
NYSE	2011/01/03	0.03749	0.00180	0.04245	0.01874	5.03219	4.55444	48.4375
S&P500	2011/01/03	0.01502	0.00033	0.01821	0.00751	2.25540	2.20832	53.24392
SSE	2011/01/04	0.01675	0.00049	0.02206	0.00838	4.19269	2.30787	50.35576
SZSE	2011/01/04	0.01662	0.00050	0.02257	0.00831	3.16300	1.97986	50.80853
Tesla	2011/01/03	0.02481	0.00106	0.03267	0.01540	5.84892	2.10859	51.00244
Tether	2017/04/14	0.00648	0.00006	0.00769	0.00324	1.05079	3.98328	43.01116
XLM	2018/06/13	0.02372	0.00133	0.03651	0.01186	10.4552	3.14901	50.0
XRP	2012/01/22	0.01462	0.00099	0.03154	0.00731	5.92154	2.06330	49.77471

such as the S&P 500 (SMAPE of 2.25540, DA of 53.24392) and high-volatility datasets, including the NYSE (MAE of 0.03749, MSE of 0.00180).

The results demonstrate the scalability and robustness of the proposed model, as its performance consistently improves with the expansion of the dataset. Specifically, the model exhibits enhanced predictive accuracy by effectively leveraging larger volumes of historical data, which results in a notable reduction in key error metrics such as MAE, MSE, and RMSE, alongside improvements in SMAPE and MASE. Furthermore, the model achieves a significant gain of 7.26% in DA, indicating its capability to make more reliable directional forecasts. Importantly, the absence of overfitting across experiments suggests that the model maintains strong generalization ability, learning meaningful patterns from the data rather than merely memorizing historical trends.

In the second step, we address the scalability of the proposed model by varying the training size and measuring the training duration over the BTC dataset, as shown in Fig. 9. The chart illustrates that the run time for processing a large dataset decreases as the training ratio increases. This downward trend suggests that more training data improves the model's efficiency, reducing computational time. Notably, a significant drop in run time is observed between lower training ratios (10% to 50%), while the reduction becomes less pronounced at higher ratios (50% to 70%). These findings suggest that selecting an optimal training ratio can significantly improve the computational efficiency of the model.

4.10 Computational complexity

Table 11 presents the computational complexity analysis of the proposed model, broken down into different components.

where N represents the number of input samples and refers to the total number of data points or instances being processed. d stands for the feature dimension, the number of features associated with each input sample. T is the time sequence length in recurrent layers, denoting the number of time steps or the length of the sequences processed by RNN, such as GRU and LSTM. Lastly, K represents the kernel size in convolutional layers, which defines the size of the filters used to perform the convolution operation.

The proposed model comprises two main components: the PAN and the NAN. The computational complexity is primarily influenced by MA and Recurrent Layers (GRU/LSTM). The overall complexity of the model is driven

Table 11 The overall Computational complexity of the proposed model

Model Component	Main Operations	Approximate Computational Complexity
Input Layer (PAN & NAN)	Receiving input data	$O(1)$
Multi-Head Attention	Computing Q, K, and V matrices and attention	$O(N \times d^2)$
Dropout & Layer Norm	Normalization and random dropout	$O(N \times d)$
AveragePooling1D	Feature dimension reduction	$O(N)$
Dense in Encoder	Fully connected layer	$O(N \times d)$
Bidirectional GRU	Processing sequential data in both directions	$O(T \times d^2)$
Flatten & reshape	Reshaping data	$O(1)$
Conv1D	1D convolution operation	$O(N \times K \times d)$
LSTM in the Fundamental Model	Processing time sequences	$O(T \times d^2)$
Batch Normalization	Normalizing batch data	$O(N)$
MaxPooling1D	Reducing feature dimensions	$O(N)$
Dense for Q, K, V	Computing attention query, key, and value matrices	$O(N \times d^2)$
Attention Mechanism	Computing attention scores	$O(N^2 \times d)$
Dense Output	Final output layer	$O(d)$
Model Aggregation (Average)	Merging outputs of both models	$O(N)$
Overall Model Complexity	Combined complexity of all components	$O(N^2 \times d + T \times d^2)$

Table 12 Statistical analysis on p-value

References	MAE	MSE	RMSE	MPL	SMAPE	MASE	DA
ANOVA	0.0071	0.0314	0.0244	0.0071	0.0155	0.0133	0.0317
T-test	0.0017	0.0314	0.0054	0.0017	0.0031	0.0022	0.0385

mainly by the most computationally expensive components, namely the MA and Recurrent Layers (BiGRU and LSTM). The terms $O(N^2 \times d)$ and $O(T \times d^2)$ represent the dominant contributions to the overall computational complexity.

4.11 Statistical analysis

We conducted a rigorous analysis using one-way ANOVA and independent sample t-tests to validate the proposed fusion model's predictive performance statistically, as shown in Table 12. The model is benchmarked against several advanced models, including CNN- LSTM, Multiple Attention, Deep CNN, TST, and LSTM.

The statistical test results, summarized comprehensively in Table 12, clearly confirm the superiority of the proposed model. The ANOVA results reveal statistically significant differences (p-values < 0.05) for key performance metrics.

Further supporting the findings, independent t-tests reinforce the proposed model's significant improvements, reporting substantial differences across all evaluated error metrics.

The robust statistical validations underscore that the observed performance improvements are not coincidental, confirming the proposed fusion model's efficacy and reliability in forecasting cryptocurrency markets.

5 Discussion

The proposed model enables parallel training and addresses the limitations of traditional recurrent neural networks when applied to time series data. Its effectiveness has been evaluated across over 26 financial datasets encompassing diverse asset classes, including stock prices, financial indices, and cryptocurrencies. These datasets span large and small markets with varying historical lengths and include SZSE, SSE, NYSE Composite Index, CSI 300, KOSPI200, S&P 500, N225, BTC, ETH, LTC, ADA, BCH, XLM, BNB, MonaCoin, Aragon, EOS, ICON, Tether, Basic Attention Token, Blocknet, Civic, DigixDAO, district0x, Dogecoin, Gnosis, Lisk, MaidSafeCoin, MCO, Nano, NavCoin, Neblio, OmiseGO, Stratis, Substratum, NEM, XRP, SH, HSI, and TSM.

In real-world financial markets, predictive models frequently encounter significant challenges due to sudden macroeconomic shifts, geopolitical tensions, and systemic shocks. Such disruptions can degrade the performance

of deep learning models trained under stable conditions. Therefore, evaluating model performance within volatile or crisis-prone contexts is crucial. This study addresses this need by utilizing diverse datasets that cover multiple regions from 2021 to early 2025, capturing several major financial disruptions. Notable examples include trade tensions and tariff policies in the United States, particularly in March 2025; geopolitical instabilities such as the ongoing Russia–Ukraine conflict and rising tensions between the United States and China; inflation and interest rate volatility, especially throughout 2022 and 2023; the failure of major financial institutions, which contributed to heightened market uncertainty and liquidity issues; post-pandemic economic effects stemming from the COVID-19 crisis, disrupting global supply chains and labor markets; and heightened cryptocurrency market volatility, particularly in assets like Bitcoin, Ethereum, and Litecoin—driven by regulatory changes, increased institutional involvement, and technological advancements. These events, occurring across diverse timeframes and financial systems, significantly influenced the behavior of stock indices (e.g., S&P 500, Hang Seng, Nikkei 225), cryptocurrencies (e.g., BTC, ETH, LTC), and significant equities (e.g., Apple, Microsoft, Tesla).

Improvements in seven performance metrics on the various financial datasets demonstrate the effectiveness of the proposed attention-based method for financial prediction. The proposed fusion model consistently exhibits substantial performance improvements across various financial datasets, encompassing large-scale (BTC) and small-scale (TSM) scenarios. Specifically, for the BTC dataset, the model achieves reductions of 72.2% in MAE, 92.5% in MSE, 71.5% in RMSE, 72.1% in MPL, 78.9% in SMAPE, and 67.0% in MASE, along with a 2.88% increase in DA, compared with five state-of-the-art models.

The model's architecture is well-suited for handling time-sensitive applications in real-world financial trading environments due to its computational efficiency and ability to learn temporal dependencies from recent price trends. Since the model is designed to predict the next three days, it can be easily integrated into live decision-support systems with minimal data latency and processing delay. Furthermore, the modular design allows for batch updates, making it feasible to operate in semi-real-time settings where predictions are generated daily.

6 Conclusion

This study proposes a forecasting method using various financial market datasets. We propose a novel dual-attention prediction model combining price and financial features. Two parallel networks, PAN and NAN, are proposed to enhance the accuracy of overall price prediction. Our approach addresses several key challenges in financial forecasting, including integrating diverse data sources such as prices, news, and technical features, effectively learning long sequences, mitigating the vanishing gradient problem, integrating attention mechanisms for financial data and related features, and capturing complex and non-linear dependencies.

In the initial step, we constructed a transformer-based network, PAN, to input various historical data while preserving the chronological order of daily data. We designed an encoder to develop and force MMSA to discern the relationship between information, preventing the calculation of attention weights for all value vectors under the assumption of their interrelation.

In the next step, leveraging ConvLSTM, SA, and BiGRU, we proposed NAN to capture spatial–temporal dependencies in financial feature data. The semantic relations between financial features and prices are essential, whereas existing models focus primarily on short- and long-term data history. The third step involves integrating the two mentioned models in parallel to enhance the accuracy of the prediction, leveraging various data sources.

The PAN–LAN attention model efficiently predicts financial data across all evaluation measures. Experimental evaluations of various datasets prove the proposed model's superiority over the state-of-the-art frameworks. However, the model's DA remains near random, suggesting that future improvements could enhance directional predictions, possibly by incorporating additional features (e.g., market sentiment) or hybrid loss functions that balance numerical and directional accuracy. Additionally, the higher SMAPE and MASE highlight the challenges of forecasting over shorter horizons, where noise and volatility are more pronounced. Future work could explore adaptive attention mechanisms to handle such scenarios better.

The model's effectiveness and reliability make it a valuable tool for time series analysts, financial professionals, investors, and policymakers seeking accurate predictions in dynamic and complex financial markets. From a stakeholder perspective, the model promotes interpretability through its attention layers, highlighting significant historical patterns contributing to each forecast. This enhances transparency and supports trust, particularly in high-risk decision-making scenarios. Financial analysts can better understand the rationale behind model outputs, increasing confidence in its practical deployment. In further study, we plan to develop the PAN–NAN framework for other time series forecasting and real-time prediction requiring detailed information.

Appendix 1- The Details of the Financial Features

Table 13.

Table 13 The details of the Financial Features (Jansen 2020)

	Group	Indicators	Functions	Financial Features
1	Return	Daily Return	$\text{Return} = \frac{\text{closingprice}}{\text{previous day closing price}} - 1$ Return on investment compared to the previous day	G
2	Momentum	MACD[a] AVG value	MACD = EMA12 – EMA 9 MACD subtracts the long-term EMA from the short-term. An EMA is a moving average with greater weight and significance on the most recent data points	G
3		RSI[b] value	$\text{RSI} = 100 - \frac{100}{1+\frac{\text{Average gain}}{\text{Average loss}}}$ The average gain or loss used in this calculation is the average percentage gain or loss during a look-back period	G
4		ULTOSC[c] value	$ULTOSC_t = 100 * \frac{4\text{Avg}_t(4)+2\text{Avg}_t(7)+\text{Avg}_t(28)}{4+2+1}$ $\text{Avg}_t(T) = \frac{\sum_{i=0}^{T-1} BP_{t-i}}{\sum_{i=0}^{T-1} TR_{t-i}}$ $BP_{t-i} = P_t^{Close} - \min(P_t^{Close}, P_t^{Low})$ $TR_t = \max(P_{t-1}^{Close}, P_t^{High}) - \min(P_{t-1}^{Close}, P_t^{Low})$ ULTOSC: Measures the average difference between the current close and the previous lowest price over three timeframes BP_t: The buying pressures TR_t: The true range	
5		K% value	Stochastic Oscillator	G
6		Momentum (M)	$\text{M} = \frac{\text{Today's closing price}}{\text{Final price of 10 days ago}}$	S
7	Profitability	Return On Assets (ROA)	$\text{ROA} = \frac{\text{Net income}}{\text{Total Assets}}$	S
8		Return on Equity (RoE)	$\text{ROE} = \frac{\text{Net income}}{\text{Shareholder's Equity}}$	S
9		Net Profit Margin (NPM)	$\text{NPM} = \frac{\text{Net income}}{\text{Revenue}} * 100$	S
10		Gross Profit Margin	$\text{GPM} = \frac{\text{Revenue}-\text{CoGS}}{\text{Revenue}}$ COGS: Cost of goods sold	S
11		Return on Sales (ROS)	$\text{ROS} = \frac{\text{Operating Earnings}}{\text{Revenue}}$	S
12	Performance	Inventory Turnover (IT)	$\text{IT} = \frac{\text{COGS}}{\text{Average value of inventory}}$ COGS: Cost Of Goods Sold	S
13		Accounts Receivable Turnover (ART)	$\text{ART} = \frac{\text{Net credit sales}}{\text{Average accounts receivable}}$	S
14		Asset Turnover Ratio (ATR)	$\text{ATR} = \frac{\text{Total Sales}}{\frac{\text{Beginning Assets + Ending Assets}}{2}}$	S
15		Fixed Asset Turnover (FAT)	$\text{FAT} = \frac{\text{Net Sales}}{\text{Average Fixed Asset}}$	S
16	Financial Risk	Debt Ratio (DR)	$\text{DR} = \frac{\text{Total Depts}}{\text{Total Assets}}$	S
17		Debt-to-Equity Ratio ($^{D}/_{E}$)	$^{D}/_{E} = \frac{\text{Total liabailities}}{\text{Stakeholder's Equity}}$	S
18		Times interest earned (TIE)	$\text{TIE} = \frac{\text{EBITD}}{\text{Interest Expense}}$ EBITD: Earnings before interest, taxes, Depreciation	S
19		Retained Earnings to Asset Ratio (REA)	$\text{REA} = \frac{\text{RetainedEarnings}}{\text{Asset}}$	S
20		Debt Ratio (DR)	$\text{DR} = \frac{\text{Debt}}{\text{Asset}}$	S
21	Market Risk	Standard Deviation of Returns (s)	$s = \sqrt{\left(\left(\frac{1}{n-1}\right)\sum_{i=1}^{n} u_i^2 - \frac{1}{n(n-1)}\left(\sum_{i=1}^{n} u_i^2\right)^2\right)}$ $u_i = \ln\frac{s_i}{s_{i-1}}$ n = 60 weekly, 36 monthly, Daily	S

Table 13 (continued)

	Group	Indicators	Functions	Financial Features
22	Relative valuation	Price-to-Earnings Ratio (P/E)	$P/E = \frac{\text{Price}}{\text{Earinig}}$	S
23		Price-to-Equity Ratio (P/B)	$P/B = \frac{\text{Price}}{\text{Book Value}}$	S
24		Free Cash Flow (FCF)	$FCF = \text{Operating Cash Flow} - \text{Capital Expenditures}$	S
25		Earnings before Interest and taxes (EBIT)	$EBIT = \text{Revenue} - COGS - \text{Operating Expenses}$ $COGS$: Cost of goods sold	S
26		Free Cash Flow to Equity (FCFE)	$FCFE = \text{Cash from operations} - \text{Capex} + \text{Net debt issued}$ Capex: Capital Expenditure	S
27	Sales	Monthly Sales	$\text{Growth rate} = \frac{\text{past value} - \text{current value}}{\text{past value}}$ The Past value of one, three, five, and nine-months price	S
28	Performance of shareholders	The institutional deals percent	$\text{Sales per capita} = \frac{\text{Sales volume}}{N}$	S
29		The individual deals percent	$\text{Sales per capita} = \frac{\text{Sales volume}}{N}$	S
30	Liquidity	Portfolio Turnover (PT)	$PT = \frac{\max\begin{cases} \textit{Fund purchases} \\ \textit{Fund sales} \end{cases}}{\text{average assets}}$	S
31		Ratio of Trading Days (RTD)	$RTD = \frac{\text{trading days in the past year}}{365}$	S
32		Average value of 30-day trades (Value_m)	$\text{Value}_m = \frac{\text{Volume} * \text{Price}}{30}$	S
33	Exchange Rate	Dollar value	Dollar value	S
34	Index	Stock market index	Iranian stock market index	S
35		Impact News (Proposed by writers)	The algebraic sum of the news scores for each symbol	S

Note 1: The mean average of each group is used as a financial feature for the proposed financial features model

Note 2:'G'denotes global and large datasets, whereas'S'refers to small and short-term datasets for price prediction based on financial features in the NA

Note 3: These 12 groups are the mean average of their subgroups. We chose this combination of the mentioned indicators according to fewer correlations with each other and different points of view on the financial market

[a]MACD: Moving Average Convergence Divergence

[b]Relative Strength Index

[c]Ultimate Oscillator

Acknowledgements Not applicable.

Author contributions All authors contributed equally to the manuscript's conception, design, data collection, analysis, and writing. All authors reviewed and approved the final version of the manuscript.

Funding The authors declare that this research received no specific financial support from any public, commercial, or not-for-profit funding agency.

Data availability The datasets used and/or analyzed during the current study are available upon request.

Declarations

Competing interests The authors declare that they have no competing interests.

References

Abbasimehr H, Paki R (2022) Improving time series forecasting using LSTM and attention models. J Ambient Intell Humaniz Comput 13:673–691. https://doi.org/10.1007/s12652-020-02761-x

Abe M, Nakayama H (2018) Deep learning for forecasting stock returns in the cross-section. In Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD

2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part I 22. Springer International Publishing, pp 273–284. https://doi.org/10.1007/978-3-319-93034-3_22
Aldhyani TH, Alzahrani A (2022) Framework for predicting and modeling stock market prices based on deep learning algorithms. Electronics 11(19):3149
Armstrong JS, Collopy F (1992) Error measures for generalizing about forecasting methods: Empirical comparisons. Int J Forecast 8(1):69–80
Bahdanau D, Cho K, Bengio Y (2014) *Neural machine translation by jointly learning to align and translate.* arXiv preprint arXiv:1409.0473
Bao W, Yue J, Rao Y (2017) A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS ONE 12(7):e0180944
Bishop CM, Nasrabadi NM (2006) Pattern recognition and machine learning, Vol 4, No 4, Springer, New York, p 738
Bustos O, Pomares-Quimbaya A (2020) Stock market movement forecast: A Systematic review. Expert Syst Appl 156:113464
Chen L et al (2018) Which artificial intelligence algorithm better predicts the Chinese stock market? IEEE Access 6:48625–48633
Cheng J et al (2024) Forecasting Bitcoin prices using artificial intelligence: Combination of ML, SARIMA, and Facebook Prophet models. Technol Forecast Soc Chang 198:122938
Cho K, et al. (2020) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv 2014. arXiv preprint arXiv:1406.1078
Chung J, Gulcehre C, Cho K, Bengio Y (2015) Gated feedback recurrent neural networks. The 32nd International Conference on machine learning, vol 37, PMLR, pp 2067–2075
Devlin J, et al. (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding/24.05. arXiv preprint arXiv:1810.04805
Devlin J, Chang MW, Lee K, Toutanova K (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the Association for Computational Linguistics: human language technologies, vol 1, pp 4171-4186. https://arxiv.org/abs/1810.04805
Dichtl H, Drobetz W, Otto T (2023) Forecasting stock market crashes via machine learning. J Financ Stab 65:101099
Dong S et al (2021) A dynamic predictor selection algorithm for predicting stock market movement. Expert Syst Appl 186:115836
Dosamantes CAD (2013) The relevance of using accounting fundamentals in the Mexican stock market. J Econ Finance Administrative Sci 18:2–10
Fama EF (1995) Random walks in stock market prices. Financ Anal J 51(1):75–80
Gao R et al (2022) Forecasting the overnight return direction of stock market index combining global market indices: A multiple-branch deep learning approach. Expert Syst Appl 194:116506
Green J, Hand JR, Zhang XF (2017) The characteristics that provide independent information about average US monthly stock returns. Rev Financial Stud 30(12):4389–4436
Hyndman RJ, Koehler AB (2006) Another look at measures of forecast accuracy. Int J Forecast 22(4):679–688
Idrees SM, Alam MA, Agarwal P (2019) A prediction approach for stock market volatility based on time series data. IEEE Access 7:17287–17298
Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. The 32nd International Conference on Machine Learning, PMLR 37, pp 448-456. https://doi.org/10.48550/arXiv.1502.03167
Jansen S (2020) Machine Learning for algorithmic trading: predictive models to extract signals from market and alternative data for systematic trading strategies with Python. Packt Publishing Ltd
Jiang W (2021) Applications of deep learning in stock market prediction: recent progress. Expert Syst Appl 184:115537
Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
Koenker R (2005) Quantile regression, vol 38. Cambridge University Press. https://doi.org/10.1017/CBO9780511754098
Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
Lahmiri S, Bekiros S, Bezzina F (2022) Complexity analysis and forecasting of variations in cryptocurrency trading volume with support vector regression tuned by Bayesian optimization under different kernels: An empirical comparison from a large dataset. Expert Syst Appl 209:118349
Lee SW Kim HY (2020) Stock market forecasting with super-high-dimensional time-series data using ConvLSTM, trend sampling, and specialized data augmentation. Expert Syst Appl 161:113704. https://doi.org/10.1016/j.eswa.2020.113704
Leippold M, Wang Q, Zhou W (2022) Machine learning in the Chinese stock market. J Financ Econ 145(2):64–82
Li Z, Zhang X, Dong Z (2023) TSF-transformer: a time series forecasting model for exhaust gas emission using transformer. Appl Intell 53(13):17211–17225
Light N, Maslov D, Rytchkov O (2017) Aggregation of information about the cross section of stock returns: A latent variable approach. Rev Financial Stud 30(4):1339–1381
Lin J (2024) A novel and effective model for automatic modulation classification prediction based on Multi-BIGRU, Multi-Encoder, and Hyper-Cross. IEEE Access 13:20260–20277. https://doi.org/10.1109/ACCESS.2024.3514078
Lin H, Chen X, Chui SY (2023) Stock prediction and analysis using LSTM network. In International Conference on Statistics, Data Science, and Computational Intelligence (CSDSCI 2022), vol 12510, SPIE 12510, pp 151–155. https://doi.org/10.1117/12.2656805
Liu Q et al (2022) Stock market prediction with deep learning: The case of China. Financ Res Lett 46:102209
Liu T, et al (2023) Herding in Chinese stock markets: Evidence from the dual-investor-group. Pacific-Basin Finance J p. 101992.
Livieris IE, Pintelas E, Pintelas P (2020) A CNN–LSTM model for gold price time-series forecasting. Neural Comput Appl 32:17351–17360
Md AQ et al (2023) Novel optimization approach for stock price forecasting using multi-layered sequential LSTM. Appl Soft Comput 134:109830
Moosa I (2014) Direction Accuracy, Forecasting Error and the Profitability of Currency Trading: Simulation-Based Evidence-Accuratezza direzionale, errore previsionale e convenienza del currency trading: evidenze dalle simulazioni. Economia Internazionale/Int Econ 67(3):413–423
Morita T (2024) Positional Encoding Helps Recurrent Neural Networks Handle a Large Vocabulary. arXiv preprint arXiv:2402.00236
Mustaqeem Kwon S (2020) CLSTM: Deep feature-based speech emotion recognition using the hierarchical ConvLSTM network. Mathematics 8(12):2133
Nabipour M et al (2020) Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. IEEE Access 8:150199–150212

Naeem M, Rizvi STH, Coronato A (2020) A gentle introduction to reinforcement learning and its application in different fields. IEEE Access 8:209320–209344

Nejad FS, Ebadzadeh MM (2024) Stock market forecasting using DRAGAN and feature matching. Expert Syst Appl 244:122952

Ng CCA, Shen J (2016) Screen winners from losers using simple fundamental analysis in the Pacific-Basin stock markets. Pac Basin Financ J 39:159–177

Nicolson A, Paliwal KK (2020) Masked multi-head self-attention for causal speech enhancement. Speech Commun 125:80–96

Parvin H, Naghsh-Nilchi AR, Mohammadi HM (2023) Image captioning using transformer-based double attention network. Eng Appl Artif Intell 125:106545

Prechelt L (2002) Early stopping-but when? Neural Networks: Tricks of the trade. Springer, pp 55–69

Quek SG et al (2022) A new hybrid model of fuzzy time series and genetic algorithm based machine learning algorithm: a case study of forecasting prices of nine types of major cryptocurrencies. Big Data Res 28:100315

Rathee N et al (2023) Analysis and price prediction of cryptocurrencies for historical and live data using ensemble-based neural networks. Knowl Inf Syst 65(10):4055–4084

Ren S et al (2023) A novel hybrid model for stock price forecasting integrating Encoder Forest and Informer. Expert Syst Appl 234:121080

Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45(11):2673–2681

Semenoglou A-A, Spiliotis E, Assimakopoulos V (2023) Image-based time series forecasting: A deep convolutional neural network approach. Neural Netw 157:39–53

Sezer OB, Gudelek MU, Ozbayoglu AM (2020) Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. Appl Soft Comput 90:106181

Shejul AA, Chaudhari A, Dixit BA, Lavanya BM (2023) Stock price prediction using GRU, SimpleRNN, and LSTM. In intelligent systems and applications: select proceedings of ICISA 2022, Vol 959, Springer Nature Singapore, Singapore, pp 529-535. https://doi.org/10.1007/978-981-19-6581-4_42

Shi X, Chen Z, Wang H, Yeung DY, Wong WK, Woo WC (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. Part of Advances in Neural Information Processing Systems (NIPS 2015), vol 28, pp 802–810. https://arxiv.org/abs/1506.04214

Souto HG, Moradi A (2024) Can transformers transform financial forecasting? China Finance Rev Int. https://doi.org/10.1108/CFRI-01-2024-0032

Tao Z, Wu W, Wang J (2024) Series decomposition Transformer with period-correlation for stock market index prediction. Expert Syst Appl 237:121424

Thakkar A, Chaudhari K (2021a) A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. Expert Syst Appl 177:114800

Thakkar A, Chaudhari K (2021b) Fusion in stock market prediction: a decade survey on the necessity, recent developments, and potential future directions. Information Fusion 65:95–107

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. 31st Conference on Neural Information Processing Systems (NIPS 2017), vol 30, Long Beach, CA, USA, pp 5998–6008. https://arxiv.org/abs/1706.03762

Wang D, et al (2021) Multi-Head Self-Attention with Role-Guided Masks. in Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43. 2021. Springer.

Wang J, Zhu S (2023) A multi-factor two-stage deep integration model for stock price prediction based on intelligent optimization and feature clustering. Artif Intell Rev 56(7):7237–7262

Wang C et al (2022) Stock market index prediction using deep Transformer model. Expert Syst Appl 208:118128

Wirth R, Hipp J (2000) CRISP-DM: Towards a standard process model for data mining. In Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining , Vol 1, pp 29-39

Wolf T (2019) Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771

Xi E, Bing S, Jin Y (2017) Capsule network performance on complex data. arXiv preprint arXiv:1712.03480.

Xu C et al (2023) A Financial Time-Series Prediction Model Based on Multiplex Attention and Linear Transformer Structure. Appl Sci 13(8):5175

Yue M, Ma S (2023) LSTM-Based Transformer for Transfer Passenger Flow Forecasting between Transportation Integrated Hubs in Urban Agglomeration. Appl Sci 13(1):637

Zhang Q et al (2022) Transformer-based attention network for stock movement prediction. Expert Syst Appl 202:117239

Zhong X, Enke D (2017) A comprehensive cluster and classification mining procedure for daily stock market return forecasting. Neurocomputing 267:152–168

Zhou Bo (2019) Deep learning and the cross-section of stock returns: neural networks Combining price and fundamental information, p 91. Available at SSRN: https://ssrn.com/abstract=3179281 or https://doi.org/10.2139/ssrn.3179281

Zhou Q, Zhou C, Wang X (2022) Stock prediction based on bidirectional gated recurrent unit with convolutional neural network and feature selection. PLoS ONE 17(2):e0262501