Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/29/20230874

Comparison between transformer, informer, autoformer and non-stationary transformer in financial market

Yumin Wu

Finance Management School, Shanghai University of International Business and Economics, Shanghai 201620, China

21078009@suibe.edu.cn

Abstract. This paper delves into the significance of predicting stock prices and carries out comparative experiments using a variety of models, including Support Vector Regression, Long Short-Term Memory model, Transformer, Informer, Autoformer, and Non-Stationary Transformer. These models are used to train and forecast the China Securities Index, Hang Seng Index, and S&P 500 Index. The results of the experiments are measured using indicators such as Mean Absolute Error and Root Mean Square Error. The findings show that the Non-Stationary Transformer model has the highest prediction accuracy. Additionally, a simple trading strategy is designed for each model and their Sharpe and Calmar ratios are compared. Since Autoformer has the highest Sharpe and Calmar, it can be concluded that Autoformer is the most practical in financial market among the four models. This research contributes to the field of stock price prediction by providing an empirical study on the application of Transformer and its derivative models which have been less explored in this domain. In conclusion, this paper offers valuable insights and recommendations for data scientists and financial engineer and introduces new methods for predicting stock prices.

Keywords: transformer, informer, autoformer, non-sationary transformer, stock price prediction.

1. Introduction

With the rapid development of artificial intelligence technology, an increasing number of application scenarios have begun to involve machine learning and deep learning algorithms. In the field of finance, time series prediction has always been a research direction of great interest. Market index prediction, as an important application scenario, has long been the focus of attention for market practitioners and the academic community. In recent years, with the rise of Transformer models, more and more research has begun to explore how to apply Transformer models to market index prediction, achieving certain results.

A market index is a representation of stocks, which are weighted by stocks. Therefore, predicting a particular market index can well reflect the ability of the model for predicting stocks in the specific market to some extent. However, Market index forecasting is a challenging problem because it is influenced by many complex factors such as economic policies, international situations, natural disasters and so on. Over the past few decades, academics and practitioners have attempted to apply various time series forecasting models to solve the market index forecasting problem.

AutoRegressive Integrated Moving Average(ARIMA) is a commonly used time series forecasting model, which is often applied in stock price prediction [1]. In 2018, Lahmiri applied the ARIMA model

DOI: 10.54254/2755-2721/29/20230874

to forecast the stock price of Apple Inc. in the US stock market and analyzed the performance and reliability of the model [2]. The results showed that the ARIMA model could predict the trend of stock prices with reasonable accuracy, but its performance would decline in highly volatile market conditions. It shares a common weakness with other statistical models in that its forecasting robustness is not high, and therefore ARIMA is often not used in practical work.

To address the limitations of the ARIMA model, academics and practitioners began exploring other time series models such as machine learning-based models and deep learning-based models. Taking machine learning models as an example, Henrique, Sobreiro, and Kimura found that Support Vector Regression (SVR) can be used to predict stock prices [3]. However, support vector regression may have certain limitations when dealing with long sequence data and requires manual setting of some hyperparameters which increases the difficulty of model tuning.

Taking deep learning models as an example, Liu et al. used a Long Short-Term Memory (LSTM) network to predict China's stock market index and achieved good results [4]. However, LSTM has certain limitations when dealing with long sequence data which may lead to overfitting and information loss. In addition, LSTM requires extensive hyperparameter tuning which increases training time and difficulty.

In addition to ARIMA, support vector regression and LSTM models there are other time series models such as Autoregressive Neural Network (ANN), Convolutional Neural Network (CNN) and Spatio-Temporal Graph Neural Network (STGNN). Hoseinzade et al. redesigned the CNN network structure to be more suitable for extracting intra-day and inter-day features at different levels for the stock market [5]. However, the computer mechanism of CNNs that extracts information by continuously stacking convolutional layers may cause a significant increase in computation and weak network convergence. These models also have their own strengths and weaknesses and have achieved different results in different markets and forecasting problems. Therefore, in practical applications choosing an appropriate time series model is a matter that requires careful consideration and comparison.

Data scientists are dedicated to inventing more advanced models for predicting market indices. In 2017, Ashish Vaswani introduced the Transformer model [6]. Originally proposed as a model for natural language processing, the Transformer model has been widely applied in fields such as time series prediction due to its excellent performance. In addition to this, data scientists have researched various variant models such as Informer, Autoformer and Non-stationary Transformer (NS Transformer) specifically for time series data prediction work based on it. These models have not only achieved significant improvements in prediction accuracy but also have obvious advantages in processing long sequence data and parallel training.

However, few financial engineers have applied these models to financial data at present. Therefore, based on the above problems, this paper will compare the results of Transformer and its derivative models in market index prediction to explore whether these models can effectively predict in the financial market. In addition, this paper designs a simple trading strategy to trade according to the prediction results and further evaluates the practicality of the model.

The structure of this paper is as follows. Chapter 2 provides an introduction to these models, while Chapter 3 describes the design of the prediction scheme. Chapter 4 presents the results of statistical prediction and analysis, and Chapter 5 demonstrates the economic forecast results. Finally, the findings are summarized, and potential future research directions are suggested in Chapter 6.

2. Model introduction

2.1. Transformer

The Transformer is a deep learning model based on the attention mechanism that is utilized for various natural language processing tasks, such as machine translation, text summarization, and semantic analysis [6]. Unlike the recurrent neural network (RNN), the Transformer completely abandons its structure and adopts an architecture with multiple layers of encoders and decoders, with self-attention and feed-forward neural networks as the basic units. This design enables parallel computation, which

DOI: 10.54254/2755-2721/29/20230874

enhances the speed and performance of model training. The Transformer model comprises multiple layers of encoders and decoders. Each encoder and decoder is composed of two sub-layers: a self-attention layer and a feed-forward neural network layer. The self-attention layer calculates the correlation between each word in the input sequence and other words, assigning varying weights to them. In contrast, the feed-forward neural network layer introduces a non-linear transformation to the output of the self-attention layer. To avoid the problem of gradient vanishing or exploding, residual connections and layer normalization are incorporated after each sub-layer in the Transformer neural network model. This approach helps to stabilize the training process and improve the overall performance of the model. The decoder also has an additional sub-layer, the encoder-decoder attention layer, which computes the correlation between each word in the decoder output sequence and all words in the encoder output sequence, assigning different weights to them. As the Transformer does not use the RNN structure, it cannot capture the sequential information of words in the input sequence. To address this issue, the Transformer adds positional encoding to the input sequence, which involves adding a vector that represents its positional information to each word vector. Typically, the positional encoding vector is generated using sine and cosine functions. (See Figure 1.)

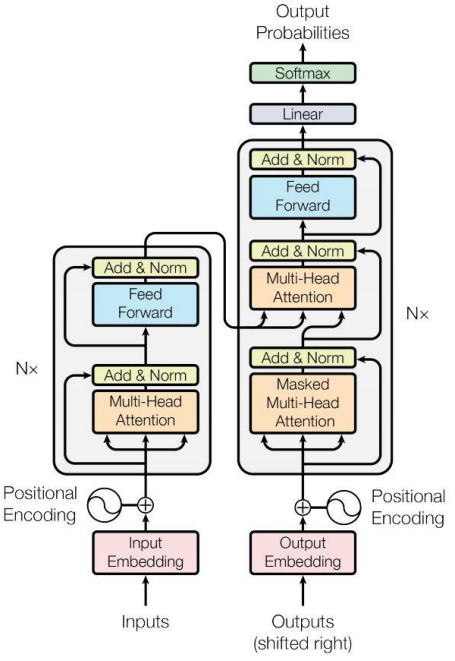

Figure 1. Transformer model architecture.

2.2. Informer

The Informer model is a deep learning framework for long sequence time series prediction [7]. Informer and Transformer models are both sequence models based on self-attention mechanisms, but they differ in structure. The Transformer model adopts an encoder-decoder structure, while the Informer model uses a self-attention structure. The Transformer model uses fixed sine and cosine functions as time encodings, while the Informer model uses learnable time encodings to encode the sequence. Moreover, the multi-head attention mechanism in the Informer model is more complex, with some heads used for inter-temporal information interaction and some heads used for intra-temporal information interaction. Finally, in terms of inter-layer connections, the Informer model adopts more complex methods such as gating and residual connections. (See Figure 2.) Based on the Transformer structure, it improves efficiency and performance in three ways:

1. The ProbSparse self-attention mechanism utilizes the long-tail property of self-attention distribution to only calculate dot products between selected queries and keys, using probability methods

DOI: 10.54254/2755-2721/29/20230874

to choose the most influential queries. This reduces time and space complexity to $O(L * log\ L)$ where L is the length of the sequence.

2. The Self-attention distilling operation reduces memory consumption and enhances feature representation by extracting dominant scores from each layer of self-attention, having the length of the input sequence. This lowers space complexity to $O\big((2-\epsilon)L\ *\ logL\big)$ where ϵ is a small constant.

3. The Generative decoder structure outputs the entire prediction sequence in one forward pass, without the need for step-by-step decoding. This improves inference speed and avoids query-key mismatch between the encoder and decoder.

Several experiments demonstrate that the Informer model significantly outperforms existing methods in prediction accuracy, inference speed, and memory usage for long sequence time series prediction [7]. The Informer model provides a novel and effective solution to this problem.

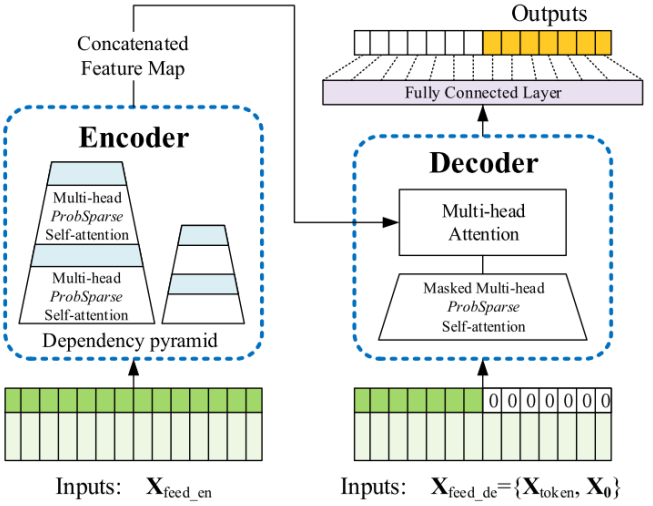

Figure 2. Informer model architecture.

2.3. Autoformer

Autoformer is a model for long-term sequence prediction that is based on the encoding-decoding architecture of Transformer but introduces two innovative modules: the Series Decomp Block and the Auto-Correlation Block [8]. The Series Decomp Block decomposes a sequence into two parts: trend-periodic and seasonal, which reflect the long-term evolution and periodic changes of the sequence. (See figure 3.) The Series Decomp Block can not only decompose the input sequence but also decompose the hidden variables in the model, thus extracting stronger predictability. The Auto-Correlation Block can replace the self-attention mechanism, using the inherent periodicity of the sequence to calculate the correlation between sub-sequences and aggregate information based on the correlation. The Auto-Correlation Block can achieve sequence-level connections and lower complexity, breaking through the bottleneck of information utilization.

DOI: 10.54254/2755-2721/29/20230874

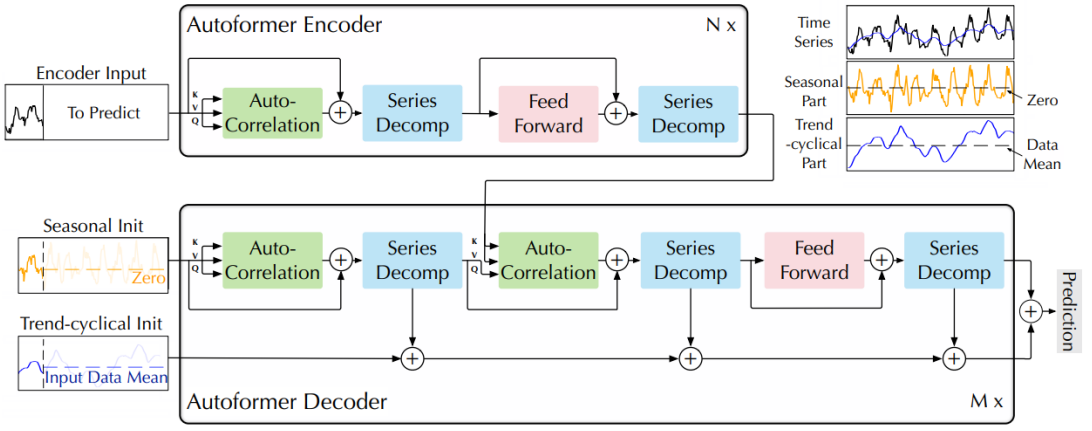

Figure 3. Autoformer model architecture.

2.4. Non-stationary transformer

Non-stationary Transformer is a deep learning model for time series prediction that builds upon the structure of the Transformer but is specifically designed to handle non-stationary time series data. Non-stationary time series data refers to data whose joint distribution changes over time, such as stock prices and temperatures [9]. This type of data is difficult for traditional Transformer models to handle because they assume that the input data is stationary, meaning it has the same or similar statistical characteristics.

To address this issue, the Non-stationary Transformer proposes a general framework consisting of two main modules: Series Stationarization and De-stationary Attention [9]. (See figure 4.) The Series Stationarization module normalizes and denormalizes the input data, ensuring that each input sequence has similar mean and variance, thereby enhancing its predictability. The De-stationary Attention module reintegrates the non-stationary information lost in the original data into the model's internal time-dependence modeling, thereby alleviating the information loss and performance degradation caused by over-stationarization.

Through these two modules, the Non-stationary Transformer not only ensures the predictability of input data but also preserves its non-stationarity feature, thus improving performance on time series prediction tasks. This model can be widely applied to the Transformer and its variants, achieving significantly better results than baseline and existing methods on multiple public datasets.

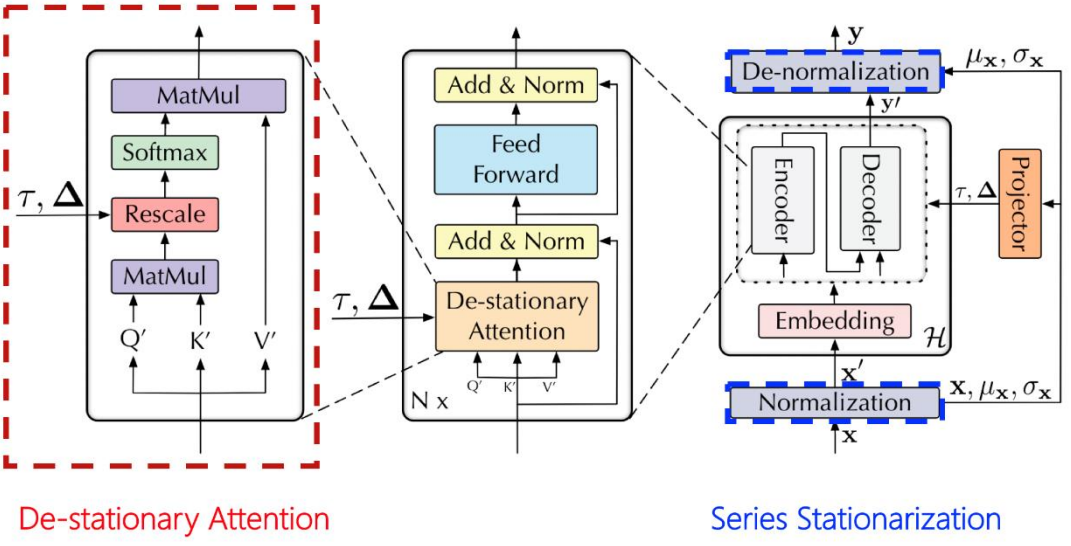

Figure 4. Non-stationary former model architecture.

DOI: 10.54254/2755-2721/29/20230874

3. Predictive scheme design

3.1. Problem restatement

Stock price prediction belongs to the time series prediction problem. By giving a set of time series observations from moment t_1 to t_L, for a total length of T time series data to a function f that satisfies the following mapping, it can be described as:

$$f(x_{t_1}, \dots, x_{t_L}) = [\hat{y}_{t_{L+1}}, \dots, \hat{y}_{t_{L+T}}] \tag{1}$$

where L represents observation length, and T represents prediction length. Simultaneously, this function also minimizes the disparity between the predicted and true values to a significant extent. From the data format point of view, the input of the stock price prediction model is a sequence of stock price historical data with a length of L and other $N - dimensional$ feature factor data (i.e. $X \in R^{(N+1)\times L}$) the output dimension of the model is a sequence of price prediction values with a length of T (i.e. $\hat{Y} \in R^{1\times T}$).

3.2. Implementation of prediction

In time series prediction, the most important issues to avoid are "using future data," "model overfitting," and "excessive model training time." Therefore, this paper will perform the following pre-processing steps on the data before training:

First, to better evaluate predictive ability, three indices were selected: the S&P 500, the Hang Seng Index and the China Securities Index(CSI 300). They represent indices of mature markets, markets between mature and immature, and immature markets respectively. The above data sources are all from the Wind database. Second, Z-score normalization of the data to improve model training accuracy and efficiency. Third, splitting the data into training, validation, and testing sets in a 7:1:2 ratio to prevent overfitting and using future data. The dataset comprises of 3005 training data, 430 validation data, and 860 test data. Fourth, using validation and early stopping methods in deep learning to avoid the risk of overfitting while also improving experimental efficiency by avoiding unnecessary training time.

For the model, the opening price, highest price, lowest price, closing price, and trading volume of the previous five days are used as input features for the encoder, and the same features of the previous two days are used as input features for the decoder to assist in prediction. The output is the closing price of the sixth day.

The model training uses an Adam optimizer with a learning rate of 0.0001, and the loss function is Mean Squared Error (MSE) [10]. The batch size is set to 32. The training set is iterated for 20 epochs, and at the end of each epoch, the current model's Mean Absolute Error (MAE) is evaluated on the validation set. If the MAE is smaller than the current best MAE, the model is saved, and the best MAE is updated. If the model's MAE is larger than the best MAE for 3 consecutive epochs, it is determined that the model has reached the learning limit, and training is stopped. This strategy is called Early Stopping, which can effectively prevent overfitting and im-prove training efficiency. Finally, the best-performing model on the validation set is chosen as the complete model trained, and its performance is evaluated on the test set.

To quantitatively evaluate the predictive accuracy of the model, the Mean Absolute Error (MAE) is used to measure the difference between predicted and actual values [10]. The Root Mean Square Error (RMSE) is used to increase the sensitivity of evaluation indicators to very large or very small errors. The smaller the values of MAE and RMSE, the closer the predicted values are to the actual values and the more accurate the prediction results. They can be described as:

$$MAE(y, \hat{y}) = (\sum_{i=1}^{n} |y_i - \hat{y}_i|) \times n^{-1} \tag{2}$$

$$RMSE = (\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \times n^{-1})^{1/2} \tag{3}$$

DOI: 10.54254/2755-2721/29/20230874

Afterwards, a simple long-only trading strategy will be designed to further validate the practicality of the model. If the predicted value on day_n is greater than the actual closing price on day_{n-1}, the strategy will choose to buy the index. Conversely, given the immature short-selling mechanism in the Chinese financial market, the strategy will not open a position if the predicted value on day_n is less than the actual closing price on day_{n+1}. The initial capital is ten million Chinese yuan, and transaction costs are considered, with a commission fee of 0.015%.

4. Statistical prediction

Based on the results presented in Table 1, the four models exhibit strong performance across most of the evaluated scenarios than the commonly used SVR and LSTM models. Notably, the NSTransformer model emerges as the top performer due to its superior handling of non-stationary sequences, closely followed by the Auto-former model. The NSTransformer model demonstrates a significant improvement in CSI prediction when compared to the Transformer model. Specifically, the MAE was reduced by 57.53%, and the RMSE was reduced by 54.76%. In addition, the Autoformer model also shows promising results, with a reduction in MAE by 51.10% and RMSE by 48.60%. Findings on Table 1 all suggest that the NSTransformer and Autoformer models may have practical applications in improving the accuracy of prediction tasks. While the Informer model, which is built on the Transformer architecture and intended for time-series prediction, was evaluated in this paper, it did not surpass the Transform-er model's performance under the established parameters. It is important to note that both the Informer and Transformer models exhibit poor performance when applied to the S&P 500 dataset, with significantly higher MAE and RMSE values than the other two models. Furthermore, as illustrated in Figure 3, these two models demonstrate an ability to identify trends, but are unable to make precise predictions of index due to significant differences in magnitude.

Table 1. The performance of models.

		SVR	LSTM	Transformer	Informer	Autoformer	NSTransformer
MAE	CSI	3.8263	0.07274	0.10135	0.11287	0.04956	0.04304
	HSI	2.6543	0.08371	0.08731	0.09187	0.08082	0.07662
	SPX	4.1038	0.10360	2.44143	2.53959	0.09039	0.08563
RMSE	CSI	4.1369	0.10248	0.13023	0.14408	0.06693	0.05891
	HSI	2.7698	0.10244	0.11431	0.12001	0.10598	0.10282
	SPX	5.1368	0.14357	2.72962	2.82438	0.12536	0.12087

5. Economic forecast

For financial engineers, the most important task is to apply predictive models to actual trades and determine if profits can be made in financial markets. In this context, three predicted indices are the underlying assets and long positions are constructed based on the prediction results of the test set for trading. It is assumed that there is a frictionless market, which means there are no transaction costs or slippage. If the closing price of day n+1 is greater than that of day n, the signal will equal to 1 and an order will be placed, executing it at the closing price of day n. Otherwise, signal will equal to 0, indicating no position is taken. The position size is 100% and the initial capital is approximately $100,000 USD or ¥700,000 RMB.

From Table 2, the overall performance of the strategies based on the four models is poor, and none of the strategies have a Sharpe ratio or a Calmar ratio greater than 1. This indicates that the practicality of the four models is average, despite their good predictive capabilities. Among the four models, Autoformer has the best performance relatively.

Table 2. The performance of strategies.

		SVR	LSTM	Transformer	Informer	Autoformer	NSTransformer
Sharpe	CSI	-1.56	-0.07	-0.01	-0.49	0.09	-0.25
	HSI	-1.02	-0.78	-0.35	-0.15	-0.56	-0.88

DOI: 10.54254/2755-2721/29/20230874

Table 2. (continued).

	SPX	**-1.87**	**-0.37**	**0.08**	**-0.54**	**0.47**	**-0.61**
Calmar	CSI	0.03	0.04	0.1	0.16	0.23	0.03
	HSI	0.02	0.09	0.15	0.03	0.21	-0.33
	SPX	0.03	0.11	0.32	0.21	0.43	0.26

The paper takes SPX as an example and draws Profit and Loss(PnL) graphs and other related curve graphs. In Figures 5 to 9, we can see that the buy signals generated by the Autoformer-based strategy are quite accurate. Despite the significant drawdown and left-skewed returns of the strategy, the net asset value curve is tilted upwards, indicating that the Autoformer's economic forecasting ability is good. In other words, it shows that the Autoformer model, which takes into account the autocorrelation of financial data, handles financial data well, and its practicality is stronger than that of other models. Autoformer is more suitable for further research on its effectiveness in predicting stock prices and applying it in such fields. However, directly applying the model to financial trading is often not feasible. A more reliable approach is to do more in-depth feature engineering for the model, debug the hyperparameters of the model, and use the model together with other timing methods (See Figures 5-9).

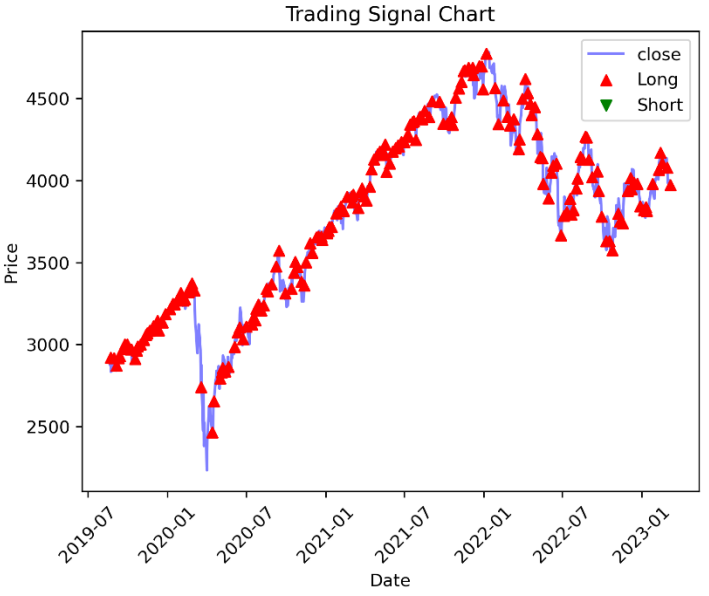

Figure 5. Backtesting result by autoformer on SPX——trading signal chart.

DOI: 10.54254/2755-2721/29/20230874

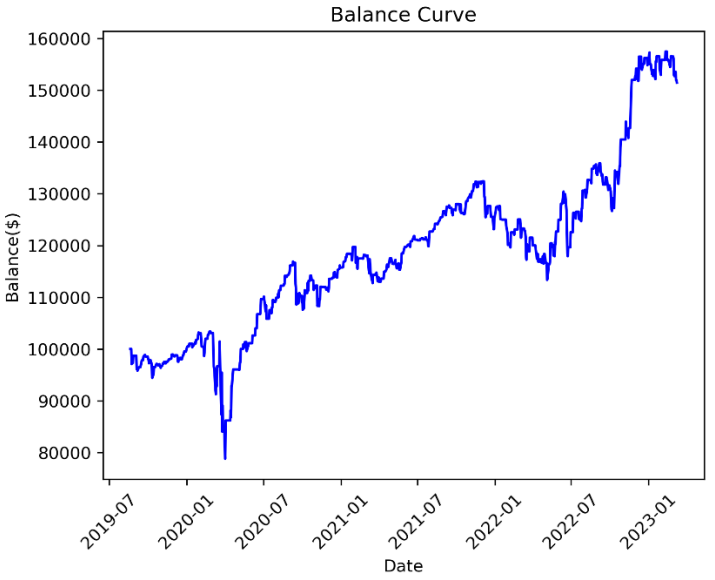

Figure 6. Backtesting result by autoformer on SPX--balance curve.

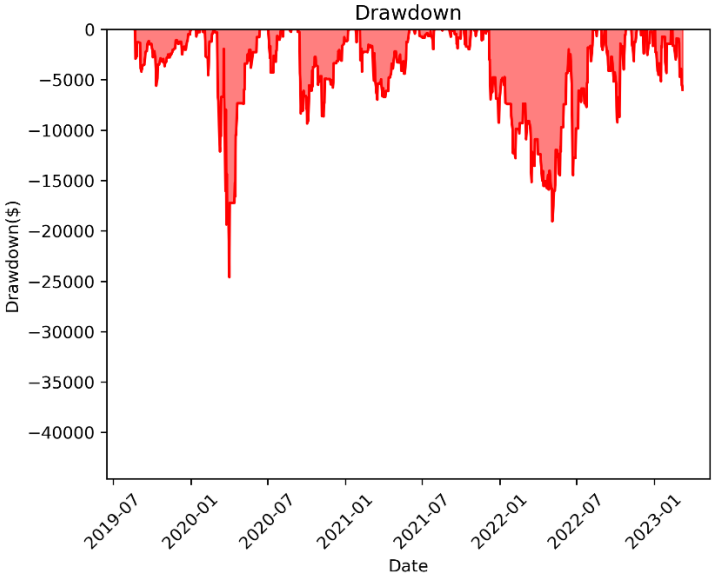

Figure 7. Backtesting result by autoformer on SPX--drawdown curve.

DOI: 10.54254/2755-2721/29/20230874

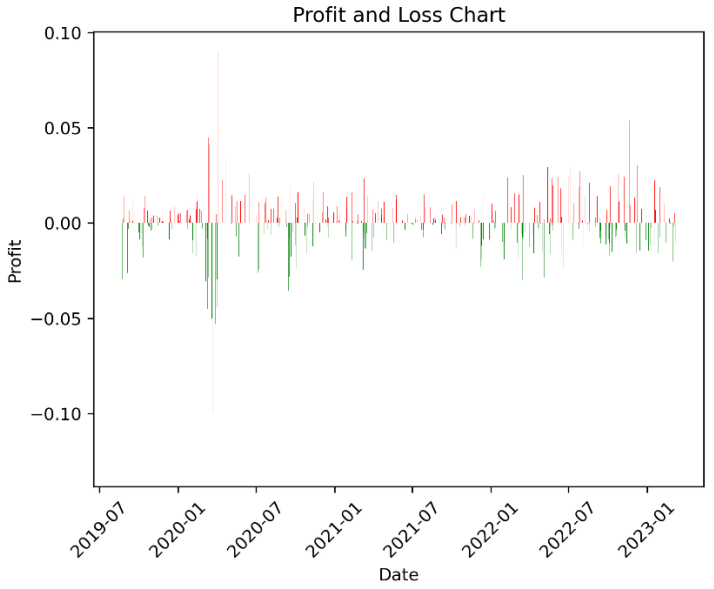

Figure 8. Backtesting result by autoformer on SPX-profit and loss chart.

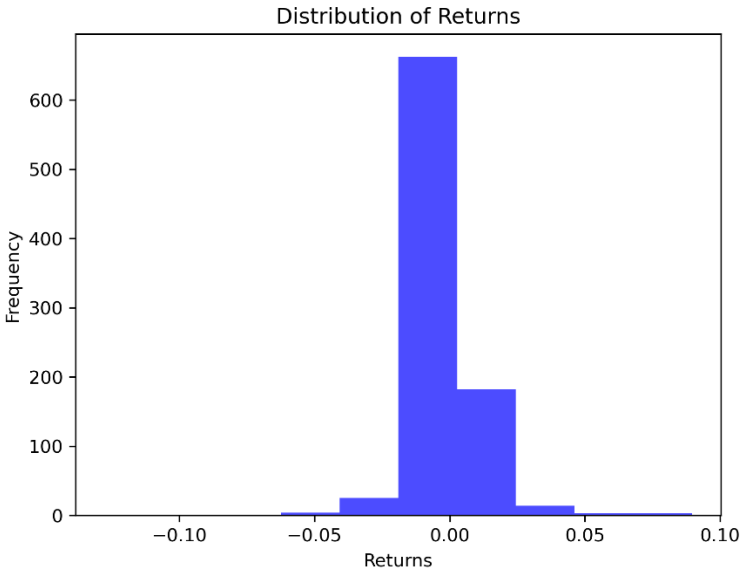

Figure 9. Backtesting result by autoformer on SPX-distribution of returns.

6. Conclusion

In conclusion, this paper has explored both the stock price prediction capabilities and practicability of the four models. Among the four models, Non-Stationary Transformer has the best predictive ability, while Autoformer has the best practicality. However, since stock prices are difficult to predict due to multiple factors such as politics, economy, society, and psychology, future researchers need to further study and consider sentiment analysis theories that integrate financial news, stock reviews, and investor sentiment into stock price prediction models. In addition, more in-depth feature engineering is required. In this paper, it only used the opening price, highest price, lowest price, closing price, and trading volume as features, but in actual quantitative investments, more complex and diverse feature factors are used. Therefore, data scientists and financial engineering need to conduct more studies to optimize the predictive performance of the model and provide more accurate and practical references for investors.

Acknowledgments
The author would like to express great thanks to the editors and reviewers.

References

[1] G. Box and G. Jenkins, "Models for Forecasting Seasonal and Non-seasonal Time Series," Defense Technical Information Center (DTIC), AD0656685, 1967. DOI:10.1016/S0167-5648(08)70667-1.

[2] A. A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," in 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, pp. 106-112, 2014. DOI: 10.1109/UKSim.2014.67.

[3] B. Henrique, V. Sobreiro and H. Kimura, "Stock price prediction using support vector regression on daily and up to the minute prices," The Journal of Finance and Data Science, vol. 4, no. 3, pp. 183-201, 2018. DOI: 10.1016/j.jfds.2018.04.003.

[4] J. Liu, X. He, J. Gao and J. Han, "Stock price prediction using LSTM, RNN and CNN sl," in ICACCI, pp. 230-234, 2017. DOI: 10.1109/ICACCI.2017.8126078.

[5] E. Hoseinzade and S. Haratizadeh, "CNNpred: CNN based stock market prediction using a diverse set of variables," Expert Systems with Applications, vol. 129, pp. 273-285, 2019. DOI: 10.1016/j.eswa.2019.03.029.

[6] A.Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems 30 (NIPS), I.Guyon et al., Eds., pp.5998–6008, 2017. DOI: 10.48550/arXiv:1706:03762.

[7] H.Y.Zhou et al., "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," arXiv preprint arXiv:2012:07436, 2021. DOI:10:48550/arXiv:2012:07436.

[8] M.Chen et al., "AutoFormer: Searching Transformers for Visual Recognition," arXiv preprint arXiv:2107:00651, 2021. DOI: 10:48550/arXiv:2107:00651.

[9] Z.Wang et al., "Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting," arXiv preprint arXiv:2205:14415, 2022. DOI: 10:48550/arXiv:2205:14415.

[10] D.P.Kingma and J.Ba., "Adam:A method for stochastic optimization," in Proceedings of the 3rd International Conference for Learning Representations(ICLR), pp1-15, 2015. DOI: 10:48550/arXiv1412:6980.