

# Tokenizing Stock Prices for Enhanced Multi-Step Forecast and Prediction

Zhuohang Zhu<sup>1[0009-0009-8362-281X]</sup>, Haodong Chen<sup>1[0000-0003-2254-5629]</sup>, Qiang Qu<sup>1[0000-0002-6648-5050]</sup>, Xiaoming Chen<sup>2[0000-0002-7503-3021]</sup>, and Vera Chung<sup>1[0000-0002-3158-9650]</sup>

<sup>1</sup> University of Sydney, Australia

[zzhu6520@uni.sydney.edu.au](mailto:zzhu6520@uni.sydney.edu.au)

{haodong.chen,vincent.qu,vera.chung}@sydney.edu.au

<sup>2</sup> Beijing Technology and Business University, China

[xiaoming.chen@btbu.edu.cn](mailto:xiaoming.chen@btbu.edu.cn)

**Abstract.** Effective stock price forecasting (estimating future prices) and prediction (estimating future price changes) are pivotal for investors, regulatory agencies, and policymakers. These tasks enable informed decision-making, risk management, strategic planning, and superior portfolio returns. Despite their importance, forecasting and prediction are challenging due to the dynamic nature of stock price data, which exhibit significant temporal variations in distribution and statistical properties. Additionally, while both forecasting and prediction targets are derived from the same dataset, their statistical characteristics differ significantly. Forecasting targets typically follow a log-normal distribution, characterized by significant shifts in mean and variance over time, whereas prediction targets adhere to a normal distribution. Furthermore, although multi-step forecasting and prediction offer a broader perspective and richer information compared to single-step approaches, it is much more challenging due to factors such as cumulative errors and long-term temporal variance. As a result, many previous works have tackled either single-step stock price forecasting or prediction instead. To address these issues, we introduce a novel model, termed Patched Channel Integration Encoder (PCIE), to tackle both stock price forecasting and prediction. In this model, we utilize multiple stock channels that cover both historical prices and price changes, and design a novel tokenization method to effectively embed these channels in a cross-channel and temporally efficient manner. Specifically, the tokenization process involves univariate patching and temporal learning with a channel-mixing encoder to reduce cumulative errors. Comprehensive experiments validate that PCIE outperforms current state-of-the-art models in forecast and prediction tasks.

**Keywords:** Financial analysis · Price forecasting · Data mining.

## 1 Introduction

As of 2023, the worldwide stock market has achieved a valuation of \$109 trillion [15], with the U.S. market contributing an average daily transaction volume of

\$110 billion. The ability to forecast and predict stock prices allows investors to make informed decisions about their portfolio positions, such as buying, holding, or selling shares. Furthermore, potential risks and volatilities can be estimated based on stock price forecasts, aiding in the mitigation of losses and minimization of risks. The challenge of stock price forecasting and prediction lies in financial markets' inherently dynamic and often unpredictable nature. The underlying distribution of stock prices is continuously influenced by a myriad of factors, both economic and geopolitical, making accurate prediction particularly difficult. As such, many researches [6,14,27] have been conducted to tackle this task.

Typically, most of the current works either perform stock price prediction [11,26] or stock price forecast [24,30]. Stock price prediction can be defined as predicting the changes of the stock price  $\Delta p$  between  $t$  and  $t + 1$ , the prediction can be approached as either a numerical estimation or a classification task. In comparison, stock price forecast explicitly aims to forecast the future price  $p$  of a stock directly. The effectiveness of each method may vary depending on the specific stock considered. Consequently, both approaches are equally important, as one may outperform the other depending on the stock in question. Although both approaches utilize the same underlying data, the transformation of stock prices from  $p$  to  $\Delta p$  significantly alters the data's distribution. As a result, models designed for stock price prediction often underperform when applied to stock price forecasting, leading to diminished generalization capabilities.

Currently, a majority of research focuses on a single-step forecast or prediction  $\{p_{t+1}\}$  for stock price, instead of making forecast or prediction for the next multiple intervals  $\{p_{t+1}, \dots, p_{t+n}\}$ . Intuitively, single-step output enables investors to decide whether to buy or sell a stock to secure profits for the following day. On the other hand, predicting stock movements over a longer term offers additional capabilities. Long-term forecast is vital for several applications, including the pricing and hedging of financial derivatives by financial institutions, as well as assessing the risk in the trading books of banks [11,19]. On top of that, it gives investors a broader view to make better-informed decisions.

Although multi-step stock price forecast offers enhanced capability when compared with single-step stock price forecast. It also poses significant challenges. The first challenge stems from the intrinsic stochastic nature of stock prices, a phenomenon well-documented in prior research [5,11,20], which points out stock price typically contains stochastic noises. Additionally, the non-stationary distribution of stock prices, characterized by shifting means and variances over time, exacerbates the difficulty of accurate multi-step forecasts. These problems coupled together make accurate multi-step forecasts more challenging. Moreover, many previous works [7,24] use an iterative method for multi-step forecast, this method is more susceptible to cumulative error as errors in early predictions can propagate and magnify in later predictions [28]. To address the aforementioned challenges, we introduce PCIE (Patched Channel Integration Encoder), an encoder model tailored for stock price forecasting.

Additionally, we introduce a data preprocessing technique that simply combines both the stock price  $p$  and its change in price  $\Delta p$  as inputs for the model.

This methodology enhances the performance of our model, as well as that of the baseline models. By incorporating both absolute prices and their fluctuations, our approach provides a more comprehensive representation of market dynamics, which facilitates more accurate forecasts and predictions across various forecasting scenarios.

To demonstrate the effectiveness of PCIE, we conducted comprehensive experiments that demonstrate our model’s superior performance in comparison to existing state-of-the-art models with respect to overall forecast and prediction accuracy. Additionally, we assessed the performance enhancements facilitated by our novel data preprocessing method.

The main contributions of this paper are summarized as:

- Demonstrate the effectiveness of tokenizing stock price for multi-step stock price forecast and prediction.
- We propose a model that achieve **SOTA** (state-of-the-art) performance for multi-step stock price forecast and prediction
- We introduce a novel data preprocessing method that enhances the accuracy of stock price forecasts and predictions.

## 2 Related Works

Stock price forecasting and prediction are both popular and challenging tasks, therefore there is a large amount of literature on this topic. In the existing body of literature, we can categorize works into several distinct groups to effectively position our research.

**Technical and Fundamental Based** Technical-based models aim to forecast future stock prices using quantitative market data, including price and volume. Earlier works focus on using traditional machine learning approaches such as Autoregressive models [13], ARIMA models [1], Fourier Decomposition [30] and Support Vector Machine(SVM) [8,17]. With the rapid advancement of computational capacity and neural network capability, numerous studies have utilized neural networks. This includes using attention-based Long Short-Term Memory (LSTM) [5], Transformer [24] and Graph Neural Network based [2,18]. In contrast, Fundamental Analysis (FA) employs external data sources, such as news [9], social media information [20], earnings call [27], or relational knowledge graphs [6], to predict price movements. In our research, we concentrate on using Technical Analysis (TA) to assess our methods for processing quantitative financial data.

**Classification and Regression** For classification models, the objective is to determine whether stock prices will increase or decrease in the subsequent time step, the classification can be either binary [3] or multi-class [4]. In comparison, Regression models [11,24], the goal is to predict the stock price directly. This provides investors with enhanced information for making informed decisions, such as the ability to purchase the best-performing stocks and use the information in other asset/derivative pricing models. On top of that, most of

the regression models either choose to predict the percentage of change in stock price(SPP) [26] or forecast the stock price directly(SPF) [30] as these objectives tend to have very different underlying distribution. We aim to address the regression task in our work. On top of that, our model tackles stock price forecast as well as stock price prediction simultaneously.

**Single and Multiple Steps** Majority of the current works on stock prediction focus on single-step stock price prediction [21] and forecast [31] for the next time step. On top of that, works such as [7] perform single-step stock price prediction and iteratively apply that to obtain a multi-step prediction. Conversely, research literature that focuses on multi-step forecasting or prediction is rare since accurate multi-step forecasting or prediction is a challenging task. Works such as [11,30] had attempted to tackle this task. For this work, we will tackle multi-step stock price forecasting as well as prediction since it offers investors more information to make long-term and risk-adjusted decisions.

### 3 Methodology

#### 3.1 Task Overview

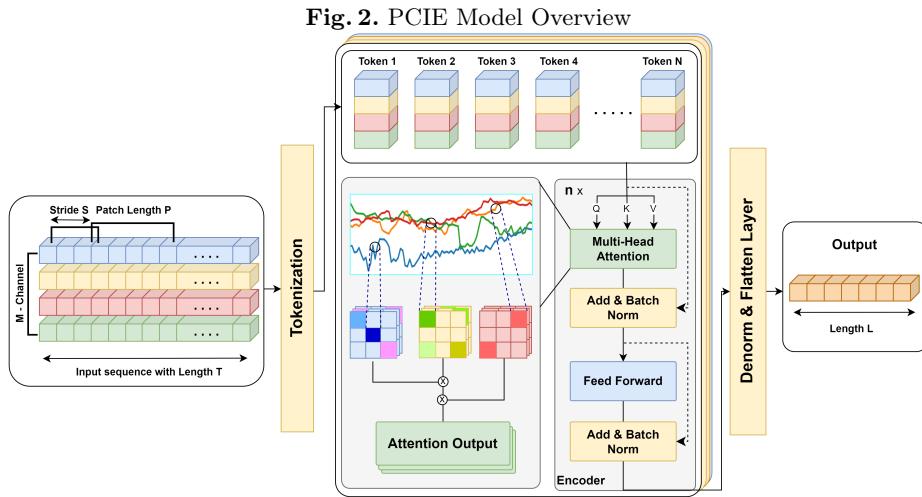
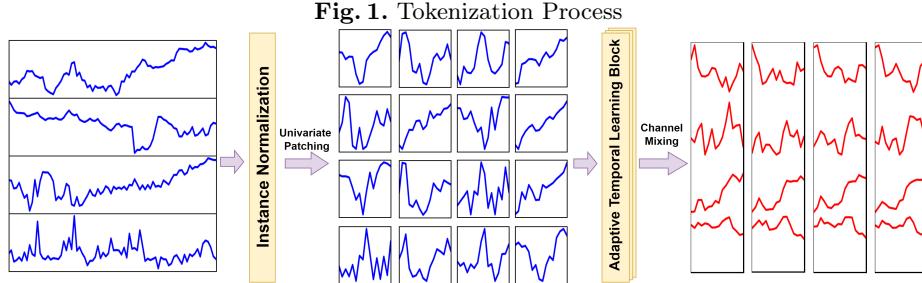
Multi-step Stock Price Forecast and prediction can be formulated as the following problem: Given an input sequence of X with T trading intervals,  $X = \{x_1, x_2, \dots, x_T\}$ , where  $x_t$  at each trading interval t is a vector  $C \in \{o_t, h_t, l_t, c_t, v_t, opt_t, hpt_t, lpt_t, cpt_t, vpt_t\}$ . It consists of open price  $o_t$ , high price  $h_t$ , low price  $l_t$ , close price  $c_t$ , volume traded  $v_t$ . For  $\{opt_t, hpt_t, lpt_t, cpt_t, vpt_t\} \in C$ , each feature is just the percentage change between t and  $t - 1$  of its original feature, i.e.  $opt_t = \frac{o_t - o_{t-1}}{o_{t-1}} * 100$ .

For stock price forecast task, the target will be the close price with a sequence length of  $L$ , target sequence =  $\{c_1, c_2, \dots, c_L\}$ . For stock price prediction task, the target will be the close price percentage change with a sequence length of  $L$ , target sequence =  $\{cp_1, cp_2, \dots, cp_L\}$ .

The data sampling frequency determines the length of the trading interval, it can range from a day when using daily trading data down to 1 millisecond when using high-frequency data.

#### 3.2 Model Overview

As shown in Figure 2, our method begins by preprocessing the stock price data, dividing it into partially overlapping segments. Each segment is then processed through an adaptive temporal learning module, which maps it into a latent space representation. This process allows the module to focus on the local characteristics of each segment. The transformation (see Figure 1) projects local semantic and characteristic information onto the latent representations, enriching the contextual data available for the subsequent self-attention encoder. This structured representation significantly enhances the model's capability to discern relevant patterns within complex financial time series data.



In the subsequent stage, segments from each series are concatenated to facilitate channel mixing. Channel mixing integrates the information across different series, allowing the model to learn correlations across multiple series. This is particularly crucial in stock market analysis, where the target series is often influenced by the movements of other series within the same dataset.

The concatenated vectors are then input into the self-attention mechanism. This model component excels in identifying intricate dependencies between elements in a sequence and extracting patterns in the input sequence. It allows the model to adaptively recognize and respond to significant temporal dynamics and interdependencies between channels.

Lastly, the outputs from the self-attention mechanism are flattened and passed through a linear layer, which obtains the direct multi-step forecasting of stock prices. This method is preferred as it tends to be less prone to cumulative errors in predictions [28], enhancing the reliability of the forecast.

### 3.3 Tokenization Process

As illustrated in Figure 1, the tokenization process includes univariate patching, adaptive temporal learning, and channel mixing. This approach parallels techniques used in natural language processing [12], where a word is divided into sub-word tokens and subsequently projected onto embedding vectors, thereby facilitating the model’s comprehension of its semantic meaning and how it relates to other words. Similarly, in this context, the tokenization process aids in the effective representation of the dataset. On top of that, it enables the model to better capture the correlation between time steps.

**Univariate Patching** Given an input sequence  $X$  with a length of  $L$  and consisting of  $M$  series. Each series is denoted as  $X^{(i)}$ , as each  $X^{(i)}$  is divided into  $N$  patches where  $x_P^{(i)} \in X^{(i)P \times N}$ . Where  $P$  is the patch length and  $S$  is the stride. The number of patches  $N$  is defined by equation (1).

$$N = \left\lceil \frac{(L - P)}{S} \right\rceil + 1 \quad (1)$$

Depending on the stride  $S$ , each patch can be overlapped or non-overlapped. The end of the sequence is padded with the value of the last element of that sequence, with the padding length equal to the stride length.

There are 2 main advantages when using patches. Firstly, it allows the temporal learning block to extract temporal information from the patches within the local scope of  $P$ . Stock price data typically contains a considerable amount of noise [23], therefore limiting the scope for temporal learning block will allow it to produce a better-learned representation. Secondly, the number of input vectors is reduced from  $L$  to approximately  $L/S$ . The computational complexity and memory cost can be decreased quadratically by a factor of  $S$ . This allows the model to process a longer input sequence which can lead to increase in accuracy and faster training time.

**Adaptive Temporal Learning Block** The Adaptive Temporal Learning Block (ATL) is designed to dynamically select the most effective temporal learning methodology from a range of options, including linear, series-independent linear and Multilayer-Perceptrons (MLP) with the objective of producing learned vectors that best represent the underlying patterns for different datasets. The ATL block is updated alongside the model during backpropagation to produce the best representation. It takes an input patch(i.e. vector) with a dimension of  $1 \times P$  and outputs a vector with a dimension of  $1 \times d_{patch}$ . This transformation is systematically executed once across all series  $\{X^1, \dots, X^m\}$  independently.

The linear model uses a shared linear layer between all the series to encode the representation of each patch. When the underlying distribution and pattern are similar between each series, a better temporal mapping can be learned by using a shared linear model  $W_{linear} \in \mathbb{R}^{d_{patch} \times P}$ .

$$X_{d\_patch}^{(i)} = W_{linear} * X_P^{(i)} + b_{linear} \quad (2)$$

The series-independent linear model maps a linear layer to each input series, therefore there are  $n$  linear layers for an input sequence with  $n$  series. Each linear layer is defined as  $W_{linear}^{(i)} \in \mathbb{R}^{d\_patch \times P}$ . This method works well when the underlying distribution and patterns between each input series have considerable deviation.

$$X_{d\_patch}^{(i)} = W_{linear}^{(i)} * X_P^{(i)} + b_{linear}^{(i)} \quad (3)$$

The Multilayer Perceptron (MLP) is employed to transform input patches into learned vectors with higher representational capacity. This choice is crucial for effectively capturing non-linear relationships and complex patterns in stock prices, which are influenced by a multitude of interconnected factors. The added complexity of MLP allows it to produce better representations when the patterns are complex.

$$X_{d\_patch}^{(i)} = \text{MLP}(X_P^{(i)}) \quad (4)$$

**Channel Mixing** A channel mixing step will be performed on the output of the ATL block, illustrated by Figure 1. This is done by sequentially flattening the output from the ATL block. On top of that, a learnable additive position encoding  $W_{pos} \in \mathbb{R}^{d\_model \times N}$  is applied to represent the order of the patches.

For an output with a series  $\{X^1, \dots, X^M\}$ , with a position  $p$  for the patches. The size of the input vector for self-attention will be  $d\_model = d\_patch * M$ , illustrated by equation (5).

$$X_{d\_model} = \text{Flatten}(X_{d\_patch}^{(1)}, \dots, X_{d\_patch}^{(M)}) + W_{pos} \quad (5)$$

### 3.4 Channel Mixing Self-attention

A modified scaled dot-product attention based on [22] is used for mapping the learned temporal vectors to the latent representation. For each head  $h = \{1, \dots, H\}$ , 3 attention results matrices  $\{Q_h, K_h, V_h\}$  can be obtained by the following equations (6).

$$Q_h = (X_{d\_model})W_h^Q \quad K_h = (X_{d\_model})W_h^K \quad V_h = (X_{d\_model})W_h^V \quad (6)$$

The final attention score for each head is calculated by equation (7), and the value of each head is combined by equation (8).

$$\text{Attention}_h(Q_h, K_h, V_h) = \text{softmax} \left( \frac{Q_h(K_h)^T}{\sqrt{d_k}} \right) V_h \quad (7)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O \quad (8)$$

We substitute the layer normalization in the multi-head attention block with batch normalization as shown in Figure 2, since batch normalization shows better performance for time series data [16,29].

### 3.5 Feed-forward Output Layers

A flatten layer coupled with a linear layer  $W_f \in \mathbb{R}^{L_f \times d_{model}}$  is used to obtain the multi-step stock price forecast with length  $L_f$  in a single step, this is also called direct multi-step forecast. This can be illustrated by the equation (9). Comparing with other transformer [24,25] based model which uses an iterative forecast method that suffers from cumulative error, direct multi-step forecast is less susceptible to cumulative error [28,32].

$$X_{L_f} = W_f * \text{Flatten}(X_A) + b_{linear} \quad (9)$$

### 3.6 Loss Function

An MSE loss function is employed to quantify the discrepancy between the predicted value and the ground truth. The overall objective function is (10).

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (10)$$

### 3.7 Instance Normalization

The statistical properties including mean and standard deviation often change over time in Stock price data [5,30]. To better address this problem, we utilize an instance normalization technique proposed by [10], this technique coupled with our temporal learning block and patching techniques addresses the problem of distribution shift in stock price data.

## 4 Experiments

### 4.1 Dataset

We extensively evaluate PCIE and our data preprocessing technique on 2 comprehensive real-world stock datasets, the statistics of the datasets are listed in Table 1.

**Table 1.** Statistics of the datasets

Dataset	Duration	# Stocks	# Trading Days
US_71	2016/01/04 to 2023/12/29	71	2011
US_14L	2005/01/04 to 2023/12/29	14	4780

For US\_71, it contains the historical price of 71 high trade volume stocks from the U.S. stock market, representing the top 6-9 stocks in capital size and trade volume across the 9 major industries. This approach is similar to other previous works on stock price forecast or prediction [26,30]. We collect the historical price from 2016/01/04 to 2023/12/29 with a daily sampling interval.

For US\_14L, we selected 14 stocks characterized by high market capitalization and trading volume. We extended the time frame for data collection, capturing historical prices from 2005/01/04 to 2023/12/29, on a daily basis. This longer time horizon was chosen to evaluate the models' performance under conditions of significant distributional shifts, which are common in stock data over extended periods. Both datasets are collected from Yahoo Finance and anomaly detection is performed to filter out the invalid data. We split all data into training, validation and testing with a ratio of 7:1:2. This is consistent across all models as well as datasets.

## 4.2 Baseline Models

We choose some of the SOTA models as baseline models to demonstrate the capability of our models. On top of that, as the multi-step stock price forecast and prediction tasks have not been widely explored, we also include SOTA models for multivariate long-time series forecasts. The reasoning behind this is that stock price data can be considered a type of time series data.

**Informer** [32] is a transformer-style SOTA model for long-time series forecasts that uses a encoder-decoder style structure. It features distilling self-attention which reduces the self-attention output size and computation complexity. It also pads zero on its decoder input to allow direct multi-step forecasts compared to traditional iterative multi-step forecasts.

**Autoformer** [25] is a transformer-style SOTA model for long-time series forecasts that uses a encoder-decoder style structure. It adds series decomposition and auto-correlation blocks in each attention block. Fast Fourier transform is used in the self-attention to learn the auto-correlation for each series. In addition, a convolution and moving-average based series decomposition is applied to the input data.

**DVA** [11] is a diffusion variational autoencoder for stock price prediction. It utilizes a variation autoencoder to generate the sequence prediction and the diffusion process which adds Gaussian noise to the sequence prediction. Finally applying a clean-up block to remove the unwanted noise.

**PatchTST** [16] is a SOTA encoder-only transformer for time series forecasts. It performs patching on each series and each series is fed into the encoder individually. However, this approach ignores the dependency between each series.

### 4.3 Experiment Settings

To compare performances for different forecast and prediction lengths, all models are evaluated on a range of different forecast and prediction lengths  $L \in \{10, 20, 40, 60\}$ . The chosen lengths are typical liquidity horizons required by financial regulators[11]. We tuned all baseline models according to the recommendations of their corresponding papers for a fair comparison. Our model is tuned according to the MSE loss from the validation dataset. An Adam optimizer is used, the learning rate of is adjusted following the OneCycleLR policy and the maximum learning rate is 0.0001. The batch size is 16. The maximum epoch is 50 with a patience of 19. The patch length  $P$  is 4 and the stride  $S$  is 1. All the other parameters are tuned according to the characteristics of the dataset. Our experiments are implemented with pytorch 2.1 and cuda 11.8. All of the experiments are done on a NVIDIA GeForce RTX 3090.

### 4.4 Results

The results obtained from our experiments are detailed in Table 2. The Patched Channel Integration Encoder (PCIE) model has shown superior performance in multi-step stock price forecasting and prediction when compared to the baseline models including PatchTST, Informer, Autoformer, and D-Va. The evaluation metric used across all models is the mean squared error (MSE), complemented by the mean absolute error (MAE) to provide a comprehensive view of the models’ accuracy.

The correlation between channels is crucial for stock price data, and models like PatchTST, which convert the input into univariate series, fail to perform well because they omit these essential inter-channel correlations. By treating each channel independently, PatchTST loses valuable information about the relationships between different channels, leading to suboptimal performance in stock price forecasting and prediction.

Models like D-Va and Autoformer perform well on prediction tasks but underperform on forecasting tasks due to their underlying assumptions about the input data distribution. D-Va, for instance, uses a diffusion process that adds noise based on a learned distribution to the target sequence. This approach works well for prediction tasks where the distribution shift is less significant. However, for forecasting tasks, where the distribution of the data can change significantly over time, D-Va’s assumptions lead to poorer performance. Similarly, Autoformer employs a learned Fast Fourier Transform (FFT) attention mechanism, which assumes a stationary distribution. This assumption holds less effectively for forecasting tasks, where the data distribution evolves, resulting in subpar performance.

**Table 2.** Performance comparison

Model	PCIE		PatchTST [16]		D-Va [11]		Autoformer [25]		Informer [32]	
Metric	MSE ↓	MAE ↓	MSE ↓	MAE ↓	MSE ↓	MAE ↓	MSE ↓	MAE ↓	MSE ↓	MAE ↓
US_71 Forecast	10	<b>0.0690</b> <b>0.1784</b>	0.0851	0.1903	0.2229	0.3338	0.1292	0.2584	0.1527	0.2904
	20	<b>0.1352</b> <b>0.2554</b>	0.1650	0.2985	0.2047	0.3193	0.2112	0.3261	0.3483	0.4271
	40	<b>0.2635</b> <b>0.3618</b>	0.2986	0.3987	0.3269	0.4240	0.3134	0.4103	0.3802	0.4613
	60	<b>0.3337</b> <b>0.4156</b>	0.3787	0.4496	0.4190	0.4895	0.3897	0.4593	0.4351	0.5010
US_71 Prediction	10	<b>1.0027</b> <b>0.7109</b>	1.0660	0.7406	1.0259	0.7278	1.0205	0.7263	1.1650	0.7729
	20	<b>0.9961</b> <b>0.7106</b>	1.0570	0.7369	1.0329	0.7296	1.0284	0.7273	1.1484	0.7730
	40	<b>0.9793</b> <b>0.7035</b>	1.0272	0.7253	1.0189	0.7124	1.0720	0.7442	1.0878	0.7516
	60	<b>0.9983</b> <b>0.7157</b>	1.0138	0.7208	1.0054	0.7243	1.0420	0.7301	1.0882	0.7451
US_14L Forecast	10	<b>0.1458</b> <b>0.2590</b>	0.1655	0.2782	0.3472	0.4046	0.3009	0.3881	0.2573	0.3510
	20	<b>0.2794</b> <b>0.3625</b>	0.2942	0.3736	0.3893	0.4562	0.4543	0.4789	0.3285	0.3970
	40	<b>0.5570</b> <b>0.5203</b>	0.5705	0.5242	0.7245	0.6120	0.7498	0.6275	0.7037	0.6043
	60	<b>0.8251</b> <b>0.6355</b>	0.8488	0.6446	0.9461	0.7012	0.9885	0.7248	0.9257	0.6990
US_14L Prediction	10	<b>1.5725</b> <b>0.8721</b>	1.6414	0.8845	1.5892	0.8736	1.5920	0.8757	1.6828	0.9140
	20	<b>1.5181</b> <b>0.8601</b>	1.5958	0.8804	1.5728	0.8773	1.5457	0.8692	1.6675	0.9004
	40	<b>1.4746</b> <b>0.8520</b>	1.5337	0.8681	1.5539	0.8708	1.5078	0.8627	1.6192	0.8801
	60	<b>1.4611</b> <b>0.8502</b>	1.5116	0.8639	1.5251	0.8709	1.4906	0.8590	1.5362	0.8714

Informer exhibits the worst overall performance among the models evaluated. One contributing factor is the convolution block used to reduce the attention output size, which inadvertently causes information loss. This loss of information is detrimental to the model's ability to accurately capture the complex patterns and dependencies in stock price data, leading to its inferior performance.

By addressing the preservation of inter-channel correlations, adapting to temporal distribution shifts, employing direct multi-step forecasting, and enhancing data representation through tokenization, PCIE overcomes the limitations of existing models and achieves superior performance in both stock price forecasting and prediction tasks.

Furthermore, the performance improvement detailed in Table 3 demonstrates that our novel data preprocessing method improves the performance across different models.

**Table 3.** Improvement in overall performance when mixing data

	PCIE	PatchTST	Informer	Autoformer	D-Va
US_71	2.762%	2.075%	0.656%	3.307%	1.647%
US_14L	3.985%	2.851%	0.907%	2.877%	1.559%

#### 4.5 Ablation Study

The ablation study was conducted to assess the impact of tokenization on the performance of the PCIE model. The experiment involved comparing the PCIE

model with and without the tokenization process across two datasets. Results summarized in Table 4 illustrate the critical role of tokenization in enhancing the model’s performance.

**Table 4.** Ablation Study

Model	PCIE		PCIE (No Tokenization)	
	Metric	MSE ↓	MAE ↓	MSE ↓
US_71	10	<b>0.0690</b>	<b>0.1784</b>	0.0838
	20	<b>0.1352</b>	<b>0.2554</b>	0.1556
	40	<b>0.2635</b>	<b>0.3618</b>	0.2964
	60	<b>0.3337</b>	<b>0.4156</b>	0.4031
Prediction	10	<b>1.0027</b>	<b>0.7109</b>	1.1056
	20	<b>0.9961</b>	<b>0.7106</b>	1.0793
	40	<b>0.9793</b>	<b>0.7035</b>	1.0456
	60	<b>0.9983</b>	<b>0.7157</b>	1.1613
US_14L	10	<b>0.1458</b>	<b>0.2590</b>	0.1712
	20	<b>0.2794</b>	<b>0.3625</b>	0.3179
	40	<b>0.5570</b>	<b>0.5203</b>	0.6483
	60	<b>0.8251</b>	<b>0.6355</b>	0.9455
US_14L	10	<b>1.5725</b>	<b>0.8721</b>	1.5957
	20	<b>1.5356</b>	<b>0.8683</b>	1.5638
	40	<b>1.5129</b>	<b>0.8664</b>	1.5536
	60	<b>1.4801</b>	<b>0.8577</b>	1.5418

The correlation between each channel in stock price data is critical because certain patterns or signals become apparent only when analyzing the relationships between multiple channels within a specific time frame. For example, the correlation between the open price  $o_p$  and the low price  $l_p$  during the time interval  $\in \{t, \dots, t + P\}$  can reveal important market dynamics that are not visible when considering these prices independently.

The process of tokenization effectively captures these inter-channel relationships by producing a representation of the patterns within the time interval  $\in \{t, \dots, t + P\}$ . Each token represents a segment of data that includes the intricate correlations between different financial indicators over time.

Furthermore, after tokenization, the resultant tokens have a higher dimensionality compared to individual time steps. This increased dimensionality allows for the application of larger attention blocks, which can handle more complex patterns across longer input sequences. The channel mixing attention block leverages this higher-dimensional representation to understand and integrate more sophisticated patterns, facilitating enhanced analysis and prediction across multiple input tokens. This approach ensures that the model comprehensively cap-

tures both temporal and spatial dependencies, leading to more accurate and robust stock price forecasting and prediction.

## 5 Conclusion and Future Work

In conclusion, this study introduced the Patched Channel Integration Encoder (PCIE), a novel approach to multi-step stock price forecasting and prediction. The PCIE model incorporates innovative techniques such as univariate patching and adaptive temporal learning, and it has demonstrated state-of-the-art performance in forecasting and predicting stock prices across multiple datasets and prediction intervals.

The ablation study further highlighted the significance of tokenization in enhancing the model's effectiveness, validating our design choices and the synergy between the model components. The integration of absolute prices and their fluctuations through our novel data preprocessing technique has proven to be particularly beneficial, offering a more comprehensive representation of market dynamics.

For future work, several avenues can be explored to extend the capabilities of the PCIE model. Firstly, incorporating additional data types such as macroeconomic indicators or sentiment analysis from news articles could enrich the model inputs and potentially improve predictive performance. Secondly, exploring more sophisticated attention mechanisms could provide deeper insights into the temporal and spatial relationships in financial data. Finally, expanding the model to include online and adaptive learning could make it more robust and responsive to market changes, thereby increasing its practical applicability for dynamic trading strategies.

**Acknowledgments.** Dr. Gregory and Professor Adrian from Optiver provided valuable suggestions and advice for this research.

## References

1. Ariyo, A.A., Adewumi, A.O., Ayo, C.K.: Stock price prediction using the arima model. In: 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation. pp. 106–112. IEEE (2014)
2. Chen, Y., Wei, Z., Huang, X.: Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 1655–1658. CIKM '18, Association for Computing Machinery, New York, NY, USA (2018). <https://doi.org/10.1145/3269206.3269269>
3. Ding, Q., Wu, S., Sun, H., Guo, J., Guo, J.: Hierarchical multi-scale gaussian transformer for stock movement prediction. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (Jul 2020). <https://doi.org/10.24963/ijcai.2020/640>
4. Fan, C., Lu, H., Huang, A.: A Novel Differentiable Rank Learning Method Towards Stock Movement Quantile Forecasting (09 2023). <https://doi.org/10.3233/FAIA230328>

5. Feng, F., Chen, H., He, X., Ding, J., Sun, M., Chua, T.S.: Enhancing stock movement prediction with adversarial training. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (Aug 2019). <https://doi.org/10.24963/ijcai.2019/810>
6. Feng, F., He, X., Wang, X., Luo, C., Liu, Y., Chua, T.S.: Temporal relational ranking for stock prediction. ACM Transactions on Information Systems pp. 1–30 (Apr 2019). <https://doi.org/10.1145/3309547>
7. Feng, F., Wang, X., He, X., Ng, R., Chua, T.S.: Time horizon-aware modeling of financial texts for stock price prediction. In: Proceedings of the Second ACM International Conference on AI in Finance (Nov 2021). <https://doi.org/10.1145/3490354.3494416>
8. Hegazy, O., Soliman, O.S., Salam, M.A.: A machine learning model for stock market prediction. arXiv preprint arXiv:1402.7351 (2014)
9. Hu, Z., Liu, W., Bian, J., Liu, X., Liu, T.Y.: Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. RePEc: Research Papers in Economics - RePEc,RePEc: Research Papers in Economics - RePEc (Jan 2017)
10. Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.H., Choo, J.: Reversible instance normalization for accurate time-series forecasting against distribution shift. In: International Conference on Learning Representations (2021), <https://openreview.net/forum?id=cGDAkQo1COp>
11. Koa, K.J., Ma, Y., Ng, R., Chua, T.S.: Diffusion variational autoencoder for tackling stochasticity in multi-step regression stock price prediction. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 1087–1096 (2023)
12. Kudo, T., Richardson, J.: Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226 (2018)
13. Li, L., Leng, S., Yang, J., Yu, M., et al.: Stock market autoregressive dynamics: A multinational comparative study with quantile regression. Mathematical Problems in Engineering **2016** (2016)
14. Lin, H., Zhang, D., Liu, W., Bian, J.: Learning multiple stock trading patterns with temporal routing adaptor and optimal transport. RePEc: Research Papers in Economics - RePEc,RePEc: Research Papers in Economics - RePEc (Jun 2021)
15. Lohan, S., Sidhu, A., Kakran, S.: The impact of investor's attention on global stock market: Statistical review of literature. International Journal of Business Forecasting and Marketing Intelligence **9**(2), 179–196 (2024)
16. Nie, Y., Nguyen, N., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers (Nov 2022)
17. Pai, P.F., Lin, C.S.: A hybrid arima and support vector machines model in stock price forecasting. Omega **33**(6), 497–505 (2005)
18. Peng, H., Yang, J.: Stock Movement Prediction via Attention-Aware Multi-Order Relation Graph Neural Network (09 2023). <https://doi.org/10.3233/FAIA230476>
19. Raunig, B.: The longer-horizon predictability of german stock market volatility. International Journal of Forecasting pp. 363–372 (Apr 2006). <https://doi.org/10.1016/j.ijforecast.2005.11.003>
20. Sawhney, R., Agarwal, S., Wadhwa, A., Shah, R.R.: Deep attentive learning for stock movement prediction from social media text and company correlations. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (Jan 2020). <https://doi.org/10.18653/v1/2020.emnlp-main.676>

21. Tuncer, T., Kaya, U., Sefer, E., Alacam, O., Hoser, T.: Asset price and direction prediction via deep 2d transformer and convolutional neural networks. In: Proceedings of the Third ACM International Conference on AI in Finance (Nov 2022). <https://doi.org/10.1145/3533271.3561738>
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30** (2017)
23. Verma, R., Verma, P.: Noise trading and stock market volatility. Journal of Multinational Financial Management **17**(3), 231–243 (2007)
24. Wang, C., Chen, Y., Zhang, S., Zhang, Q.: Stock market index prediction using deep transformer model. Expert Systems with Applications **208**, 118128 (2022)
25. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in Neural Information Processing Systems **34**, 22419–22430 (2021)
26. Xu, Y., Cohen, S.B.: Stock movement prediction from tweets and historical prices. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Jan 2018). <https://doi.org/10.18653/v1/p18-1183>
27. Yang, L., Li, J., Dong, R., Zhang, Y., Smyth, B.: Numhtml: Numeric-oriented hierarchical transformer model for multi-task financial forecasting. Proceedings of the AAAI Conference on Artificial Intelligence pp. 11604–11612 (Jul 2022). <https://doi.org/10.1609/aaai.v36i10.21414>
28. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 11121–11128 (2023)
29. Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., Eickhoff, C.: A transformer-based framework for multivariate time series representation learning. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (Aug 2021). <https://doi.org/10.1145/3447548.3467401>
30. Zhang, L., Aggarwal, C., Qi, G.J.: Stock price prediction via discovering multi-frequency trading patterns. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Aug 2017). <https://doi.org/10.1145/3097983.3098117>
31. Zhao, Y., Du, H., Liu, Y., Wei, S., Chen, X., Zhuang, F., Li, Q., Liu, J., Kou, G.: Stock movement prediction based on bi-typed hybrid-relational market knowledge graph via dual attention networks
32. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 11106–11115 (2021)