

Research Article

Impact of Technical Indicators and Leading Indicators on Stock Trends on the Internet of Things

Chien-Ming Chen , Yuxiao Gong , and Jimmy Ming-Tai Wu 

Shandong University of Science and Technology, Qingdao, China

Correspondence should be addressed to Jimmy Ming-Tai Wu; wmt@wmt35.idv.tw

Received 20 March 2022; Revised 2 May 2022; Accepted 9 May 2022; Published 27 May 2022

Academic Editor: Kuruva Lakshmana

Copyright © 2022 Chien-Ming Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Internet of things (IoT) has had an enormous impact on the financial industry. With IoT, people can obtain real-time financial information; moreover, investment and financial management have become more flexible and diverse. Because of their high returns and strong liquidity, stocks have become essential commodities through which people invest and manage money. However, high returns are often associated with high risks. Therefore, it is important for investors to forecast the trends of future stock prices. This study uses a new stock trend prediction framework to predict changes in the stock price direction on the next trading day using data from the past 30 trading days. This framework uses two-dimensional convolutional neural networks to classify stock prices into three categories: up, down, and flat. In addition, to analyze the influence of different types of input on the prediction model, historical data, futures, options, technical indicators, and mixed data are taken as the model's input. Experiments on US and Taiwan stocks proved the validity of the prediction model. The method proposed in this study is compared with buy-and-hold and random choice trading strategies. Results show that the model's profitability is better than the two baseline strategies.

1. Introduction

The rapid development of the Internet of things (IoT) [1, 2] technology has intricately connected all things possible, bringing in efficiency and vitality to all walks of life [3, 4]. Now, IoT has been used in various environments [5–7]. In the financial industry, financial risks [8], fraud [9], and crises [10, 11] can be detected in time through IoT. In addition, investors can obtain real-time financial information through IoT to invest in appropriate instruments [12]. The financial market is an integral part of social and economic development. Changes in the financial market reflect changes in an economy's operations. As a prominent financial exchange center, the stock market also plays an essential role in society and the economy. Therefore, research and forecasting of stocks have been an appealing topic to many scholars and investors.

Early econometric researchers mainly used the time-series analysis method in stock price prediction research. They used statistical means to analyze and model past stock

prices to identify potential stock price patterns. Classical algorithms include the autoregressive and moving average (ARMA) model [13], autoregressive integrated moving average (ARIMA) model [14], generalized autoregressive conditional heteroscedasticity (GARCH) model [15, 16], and the fusion of several statistical models [17, 18]. However, these statistical models have specific preconditions for the series: the ARMA model requires a stationary series, the ARIMA model requires the series to be stationary after difference, the GARCH model requires the series to be stationary, and the residual of the sequence is a pure random series with zero mean and heteroscedasticity. Stock price series are usually nonstationary or difference-stationary. Although the difference can make the nonstationary series stationary, it loses the time-series correlation.

In the 1980s, there was an upsurge in machine learning research worldwide and machine learning technology developed rapidly in various fields. Various shallow machine learning models have been widely used, such as support vector machines (SVM) [19, 20]. These studies show that

machine learning is feasible for stock forecasts. Subsequently, some researchers combined the metering economic model with the machine learning model [21, 22], attempting to obtain the advantages of the two types of methods, thereby achieving better predictive performance. However, it is challenging for conventional shallow machine learning models to fit the complex stock-forecasting problem. In the 21st century, with the advent of big data and advanced computer technology, deep learning technology has gained importance. As complex nonlinear functions can be fitted and features can be extracted automatically, deep learning is a promising method for predicting stocks. Convolutional neural networks (CNNs) [23] and long short-term memory (LSTM) [24] are most frequently used in stock-forecasting research. Numerous studies have shown that deep learning models outperform classic machine learning models [25–27].

Most previous deep learning models use the historical data of stocks to predict stock prices. However, the stock market is ineffective and investors are not rational. Therefore, the historical prices of stocks cannot fully reflect the future change trends of stocks and the predictive ability of the model using only historical stock price data is limited. Researchers have added macrofinancial information [28], financial news [29], and social media sentiment [30, 31] to the input characteristics to resolve this limitation. However, economic news and social media sentiments tend to lag. By the time the information is released on the social network, the event's impact on the stock price would have taken effect, and the stock price would have begun to correct. Moreover, the cost of obtaining accurate and valuable information is high.

This study uses a two-dimensional convolutional neural network (CNN) prediction framework. It can capture the correlation between features and the time dependence of the features. In addition, this study uses historical stock data, technical analysis indicators, and leading indicators (futures and options) as inputs for the forecasting model. To a certain extent, technical analysis and leading indicators reflect future changes in stock prices. Leading indicators change the trend before the stock prices change, which can send market signals in advance, and technical indicators can reflect the inherent connotation of the stock market. In addition, we experimented and analyzed the effects of different input information on the prediction performance and profitability of the model. Furthermore, we discuss the input characteristics that are more appropriate in various stock markets. The main contributions of this study can be summarized as follows:

- (1) A three-classification framework based on CNN is used to predict the change direction of stock price. It uses historical data, technical indicators, and leading indicators as input. Compared with previous work, it increases the number of influencing variables and decreases the lag.
- (2) Experiments are conducted with different inputs on different stock data and the influence of different inputs on the prediction results is analyzed.

- (3) Stock trading was simulated to calculate the rate of return and compare the proposed prediction framework with buy-and-hold and random choice strategies. The results show that the proposed prediction framework is better than the two baseline strategies.

The remainder of this paper is organized as follows. Section 2 briefly reviews the applications of machine learning and deep learning to stock forecasting. Section 3 introduces stock forecasting, data processing, and the framework of the proposed forecasting method. Section 4 presents the experimental setup and results. Finally, Section 5 summarizes the results.

2. Related Work

This section reviews recent research on the application of machine learning in stock forecasting, mainly in stock price trend forecasting. These studies can be divided into traditional machine learning and deep learning research, as described in Subsections 2.1 and 2.2, respectively. Table 1 presents a brief overview of this section.

2.1. Traditional Machine Learning in Stock Forecasting.

Traditional machine learning methods, such as support vector machine (SVM), random forest (RF), and artificial neural network (NN), have been widely used and have achieved relatively accurate results in stock forecasting.

Kara et al. [32] used NN and SVM to predict the Istanbul Stock Exchange's National 100 Index (ISE 100). They used ten technical indicators, such as 10-day simple moving average, 10-day weighted moving average, and momentum, as inputs. The experimental results showed that NN has a better prediction performance than SVM. Ballings et al. [33] collected yearly data from 5767 listed European companies. They used various machine-learning techniques to predict stock movements one year later than one day later. Among these methods, RF was the best predictor, followed by SVM, kernel factory (KF), AdaBoost, NN, and k-nearest neighbor (KNN). The worst was logistic regression (LR). A wavelet neural network (WNN) is a variant of a neural network. It used a wavelet base instead of a nonlinear sigmoid function in the hidden layer. Fang et al. [34] used WNN to predict the next trading day's opening price; they also used genetic algorithms (GA) to find the best parameters and weights.

As the prediction performance of a machine learning model largely depends on the input features, researchers have used various feature selection methods to extract the best feature subset to improve the accuracy of the prediction model. Ni et al. [35] proposed a fractal feature selection SVM prediction model, which combined the fractal dimension with the ant colony algorithm (ACO) for feature selection of 19 technical indicators and then used SVM for prediction. They compared their method with five feature-selection methods: Information gain, symmetrical uncertainty, ReliefF, correlation-based feature selection, and OneR. Finally, it was proved that the SVM could obtain higher precision with relatively few features using the fractal

TABLE 1: Prediction methods, input features, and model types.

Methods		Data				Target		
		HD ¹	TI ²	F/O ³	OFD ⁴	2-class	3-class	Regression
#1 [32]	SVM, NN		√			√		
#2 [33]	RF, SVM, KF, Adaboost, NN, KNN, LR				√	√		
#3 [34]	WNN + GA	√						√
#4 [35]	Fractal dimension + ACO + SVM		√			√		
#5 [36]	SVM + SVM		√			√		
#6 [37]	HS + NN, GA + NN		√					√
#7 [38]	GA + XGBoost	√	√			√		
#8 [39]	2D CNN	√	√			√		
#9 [40]	2D CNN, 3D CNN	√	√			√		
#10 [41]	CNN, LSTM, MLP, RF	√					√	
#11 [42]	2D CNN	√		√			√	
#12 [43]	LSTM	√		√			√	
#13 [44]	CNN + LSTM	√		√			√	
#14 [45]	NTN				√	√		
#15 [46]	Attention LSTM + transformer	√				√		

¹HD: historical data. Basic stock price information, including opening price, the highest price, the lowest price, closing price, and volume. ²TI: technical indicators. It refers to a collection of stock data calculated by mathematical formulas. ³F/O: futures/options. Information about futures contracts and options contracts for stocks. ⁴OFD: other financial data. Information beyond the stock itself—GDP, unemployment rate, other macroeconomic indicators, crude oil prices, gold prices, information about other stocks related to this stock, financial news, and so on.

feature selection method. Lin et al. [36] first used a correlation-based SVM filter to rank and select technical indicators to generate feature subsets and weights and then used a quasilinear SVM to predict the Taiwanese stocks.

As is well known, if there are N features, we can obtain $2^N - 1$ combinations. When the feature dimension is extremely high, the number of combinations increases exponentially. Traditional feature selection methods are time-consuming and prone to a “combinatorial explosion” of search. Therefore, many intelligent optimization algorithms have been developed to solve these complex optimization problems. Göçken et al. [37] used two heuristic algorithms, genetic algorithm (GA), and harmony search (HS) to select the optimal feature subset from 45 technical indicators and then used NN for stock closing price prediction. The heuristic algorithm also determines the number of hidden layer neurons. The experiment on the BIST 100 Index showed that the hybrid model using an intelligent optimization algorithm has a better prediction performance than an ordinary NN. Yun et al. [38] also used the GA for feature selection. However, they extracted the best feature subset from historical price data and technical indicators and used XGBoost as a predictor.

2.2. Deep Learning in Stock Forecasting. Deep learning has been one of the most used stock forecasting methods in recent years. Among various depth learning models, CNNs and recurrent neural networks (RNNs) are two essential models for time-series prediction.

Gunduz et al. [39] used a two-dimensional (2D) CNN to predict the daily stock price trend of the BIST 100 Index. They proposed a CNN-Corr classifier, which rearranged the features according to the correlation between the fea-

tures and then input the rearranged features into the CNN for prediction. Hoseinzade and Haratizadeh [40] proposed a CNNpred architecture with two variations, 2D-CNNpred and 3D-CNNpred. 2D-CNNpred was aimed at training a general model for all markets using historical data and technical indicators. 3D-CNNpred trains a model for each market by adding market dimensions to consider the correlation between different stock markets.

Kamalov [41] used NNs to predict significant changes in stock prices. They set a threshold of significance and used NNs to predict when stock prices would exceed that threshold. They also compared the RF and RSI models with NN models, CNN, LSTM, and MLP. The results showed that the NN model is more accurate at predicting significant changes in asset prices than other methods. LSTM is best as it is suitable for processing time series. CNN followed because it could extract the spatial relationships between the data. Wu et al. used stock futures and options data as additional information. They conducted experiments on CNN [42], LSTM [43], and hybrid models of CNN and LSTM [44]. Experiments proved that futures and options data made sense for predicting stock prices and the prediction performance of the hybrid model was better than that of the single model.

In addition to CNN and RNN, other deep learning models have been applied to stock forecasts. Ding et al. [45] constructed a new neural tensor network (NTN) to extract meaningful information from news texts and predict the impact of news events on stock prices for better trade. Yoo et al. [46] proposed a new stock movement prediction method, data-axis transformer with multilevel contexts (DTML). This method considers the influence of a stock’s historical data on its future prices. It takes advantage of the

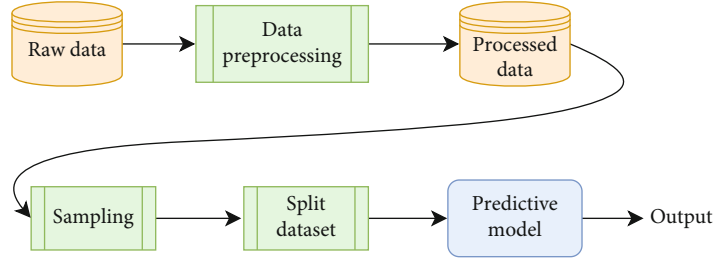


FIGURE 1: Process of stock forecasting.

transformer's long attention to dynamically consider the influence of other stocks on stock prices. However, because the transformer considers short-term dependencies as important as long-term ones, this method cannot capture the local features of the data.

3. Methodology

This section details the problem definition, data processing method, and the proposed stock prediction framework. In addition, historical data, futures, options, and technical indicators are described. Figure 1 shows the stock price trend prediction process used in this study. First, the original data, which refers to historical data, futures, options, technical indicators, or combinations, were processed. Samples were then collected from the processed data and divided into training, verification, and test sets. These samples were then input into the prediction model. The details of the data-processing method and model architecture are introduced later.

3.1. Problem Definition. In actual investments, investors do not buy or sell stocks on the rise or fall of stock price. They only buy or sell stocks when the stock price goes up or down beyond their set limits. Therefore, the two-category model that predicts whether future stock prices will rise or fall is unable to meet the problem requirement. Therefore, the stock forecast is defined as a three-category problem in this study to predict the future stock price rise, fall, or no rise or fall. In other words, it predicts the buy, sell, and hold signals.

However, investors have different risk preferences; therefore, they set different boundaries. Steady investors can tolerate lower losses. They prefer relatively stable to high profits. In contrast, active investors can tolerate greater risks because they know that higher returns often accompany higher risk. Thus, stock prices can be divided into rising, falling, and flat categories by setting thresholds for rising and falling. In this study, the direction of stock price change is defined as follows:

$$y_i = \begin{cases} 1, & \text{if } c_i > \text{upperbound} \\ -1, & \text{if } c_i < \text{lowerbound} \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

where y_i is the price direction on trading day i , c_i is the value of the change in the closing price on trading day i ,

upperbound is the rising threshold, and lowerbound is the falling threshold. c_i can be calculated as follows.

$$c_i = p_i - p_{i-1}, \quad (2)$$

where p_i is the closing price on trading day i . For the closing price sequence of a stock, $P = \{p_1, p_2, p_3, \dots, p_N\}$, where N denotes the number of trading days. Stock price change sequence $C = \{p_2 - p_1, p_3 - p_2, \dots, p_N - p_{N-1}\} = \{c_2, c_3, c_N\}$. If c_i satisfies $c_i > \text{upperbound}$, y_i is 1 (rising). Similarly, if c_i satisfies $c_i < \text{lowerbound}$, y_i is -1 (falling). Otherwise, y_i is 0 (flat). The traders can set these two thresholds. In particular, when upperbound = lowerbound = 0, it is equivalent to a binary classification model.

Given a data matrix $X_t = (X_{t-d+1}, X_{t-d+2}, \dots, X_{t-1}, X_t)^T$, $X_i = (x_{i1}, x_{i2}, \dots, x_{im})$, where d is the number of past days and m is the dimension of the input features. The problem is to predict \hat{y}_{t+1} , the price change direction of the next trading day, using data matrix X_t .

3.2. Data and Preprocessing. This section explains the data used in this study and how they were handled. The data included historical, futures, options, and technical indicators.

3.2.1. Historical Data. These refer to certain basic information for stock trading. The properties are as follows. The opening price refers to the transaction price of the first transaction per share after the trading day's opening. The highest/lowest price is the highest/lowest of all transaction prices on the current trading day. Close price is the weighted average price of all transactions in a minute before the last transaction of the trading day. Transaction volume is the trading quantity of a trading day. Factors that have influential stocks are reflected in historical prices. Moreover, the price of stocks has certain regularities. Hence, finding potential laws from historical stock data can help predict future share prices.

3.2.2. Futures. It refers to the futures contract. Like stocks, futures contracts also have opening price, highest price, lowest price, closing price, and trading volume. Futures reflect investors' expectations for stock prices. The futures contract agrees to deliver assets at a specified price over a period of time. The trader must perform the contract at the agreed price on the delivery date, regardless of the current stock price. When the futures contract expires, the stock price rises or falls, and there will be some futures contracts that are

TABLE 2: Technical indicators used in this paper.

Category	Technical analysis indicators
Overlap study indicator	BBANDS, DEMA, EMA, HMA1, HT_TRENDLINE, KAMA, MAMA, MAVP, MIDPOINT, MIDPRICE, T3, TRIMA, WMA
Momentum indicator	ADX, ADXR, APO, AROON, AROONOSC, BOP, CCI, CMO, DX, MACD, MFI, MINUS_DI, MINUS_DM, MOM, PLUS_DI, PLUS_DM, PPO, ROC, ROCP, ROCR, ROCR100, RSI, STOCH, STOCHF, STOCHRSI, TRIX, ULTOSC, WILLR
Volume indicators	AD, ADOSC, OBV
Volatility indicator	ATR, NATR, TRANGE
Price transform indicator	TYPPRICE, WCLPRICE
Cycle indicator	HT_DCPERIOD, HT_DCPHASE, HT_PHASOR, HT_SINE, HT_TRENDMODE
Statistic indicator	BETA, CORREL, LINEARREG, LINEARREG_ANGLE, LINEARREG_INTERCEPT, LINEARREG_SLOPE, STDDEV, TSF, VAR

¹HMA: Hull moving average. Hull moving average was put forward by Hull [47] in 2015 to solve the lag of the moving average. It is smoother than the general moving average and more sensitive to price changes. The formulas are as follows: $HMA_n(\text{price}) = WMA_{\sqrt{n}}(2 \times WMA_{\lfloor n/2 \rfloor}(\text{price}) - WMA_n(\text{price}))$, where $WMA_n(\text{price}) = \sum_{i=0}^{n-1} \text{price}_{t-i} \times (n-i) / n \times (n+1/2)$.

higher than or lower than the current stock price. Therefore, near delivery days, traders will long or short futures for cash out so that the futures price and the spot price gradually converge. Therefore, futures can help predict stock prices.

3.2.3. Options. Like futures, options are also a type of contract. However, options trade not in material object assets but in the right to buy or sell. The options market deals with the right to buy or sell a certain number of commodities agreed in the contract at a specific price within a certain period. There are two types of options: call options and put options. A call option is a right to buy and a put option is a right to sell. The options owner can choose to exercise or not exercise or sell the right. When investors are optimistic about a stock, they will buy the call options of the stock in the options market; otherwise, they will buy put options. Therefore, options trading can reflect investors' sentiment to a certain extent.

3.2.4. Technical Indicators. Technical indicators are derived from a series of operations on primary stock data. They can reflect some potential information about stocks, which can help investors predict the future stock trend. Bollinger bands, for example, can show a safe high or low level of a stock price, providing an important reference for future price movements. When the width of the band is small, if the stock price starts to break through the upper track line, then the price will usually continue to go up; if the price goes down through the lower track line, then the price will usually continue to go down. Technical indicators used in this study are derived from TA-Lib (<http://ta-lib.org>). There are seven categories of technical indicators used, overlap study indicators (14 kinds), momentum indicators (28 kinds), volume indicators (3 kinds), volatility indicators (3 kinds), price transform indicators (2 kinds), cycle indicators (5 kinds), and statistical functions (9 kinds), listed in Table 2. For a more detailed introduction, please refer to TA-Lib documentation (http://mrjbq7.github.io/ta-lib/doc_index.html).

These data are helpful for predicting stock prices, especially futures and options data, which are leading indicators that change before the price of a stock. In the original data-

set, the features had different orders of magnitude. Features of a larger order of magnitude dominate the predictive model results. Correspondingly, a small order-of-magnitude feature has little effect on the results. However, small-magnitude features may also be critical. Therefore, to eliminate the adverse effects of different orders of magnitude, the data must be normalized or standardized to scale in a specific range. This study used a sliding window method to normalize and generate samples. Algorithm 1 shows the method. Each sample includes data from the past d trading days, normalized using the maximum, minimum, and average values of the past T trading days (Lines 3 to 9).

The stock series has autocorrelation, which decays over time. Therefore, to select the appropriate T , autocorrelation function (ACF) graphs and partial autocorrelation function (PACF) graphs are drawn and partially shown in Figure 2. In the ACF graph, the x -axis coordinate represents the order of delay. The y -axis coordinates represent the autocorrelation coefficient between the closing price sequence $S_t = \{p_{x+1}, p_{x+2}, \dots, p_t\}$ and the closing price sequence of delay x -order $S_{t-x} = \{p_1, p_2, p_{t-x}\}$. In the PACF graph, the x -axis coordinates represent delay order. The y -axis coordinate represents the autocorrelation coefficient between the closing price y_t and the closing price of the delayed x -order y_{t-x} after excluding the influence of the middle $x-1$ intermediate variables. The blue area in the figure represents the error band and the data in this area are considered to be insignificant. As shown in Figure 2, the influence of the closing price history on the current closing price decays over time, and this effect is cyclical. Careful consideration sets T to 30, that is, sets the length of the historical data to 30 days. Moreover, d is set to 120, which is approximately the number of trading days in half a year.

3.3. Model Architecture. In this section, we describe the structure of the proposed predictive model. In this study, a two-dimensional CNN was used. Figure 3 illustrates the model framework.

This model consists of two parts: convolution and full connection. The convolution part was used for feature

```

Input:  $X = \{X_1, X_2, \dots, X_N\}$ , raw data;  $T$ : sliding windows length;  $d$ : length of time period;
Output: sample set  $S$ .
1: initial  $i = T$ ;
2: repeat
3:    $\text{mean} = \text{mean}(X_{i-T+1}, \dots, X_i)$ ;
4:    $\text{max} = \text{max}(X_{i-T+1}, \dots, X_i)$ ;
5:    $\text{min} = \text{min}(X_{i-T+1}, \dots, X_i)$ ;
6:   for each  $X_j (i-d+1 \leq j \leq i)$  do
7:      $X_j^* = X_j - \text{mean}/\text{max} - \text{min}$ 
8:   end for
9:    $S_i = \{X_{i-d}^*, \dots, X_i^*\}$ ;
10:   $i = i + 1$ ;
11: until ( $i > N$ )
12:  $S = \{S_T, S_{T+1}, \dots, S_N\}$ 

```

ALGORITHM 1: Data normalization and sampling.

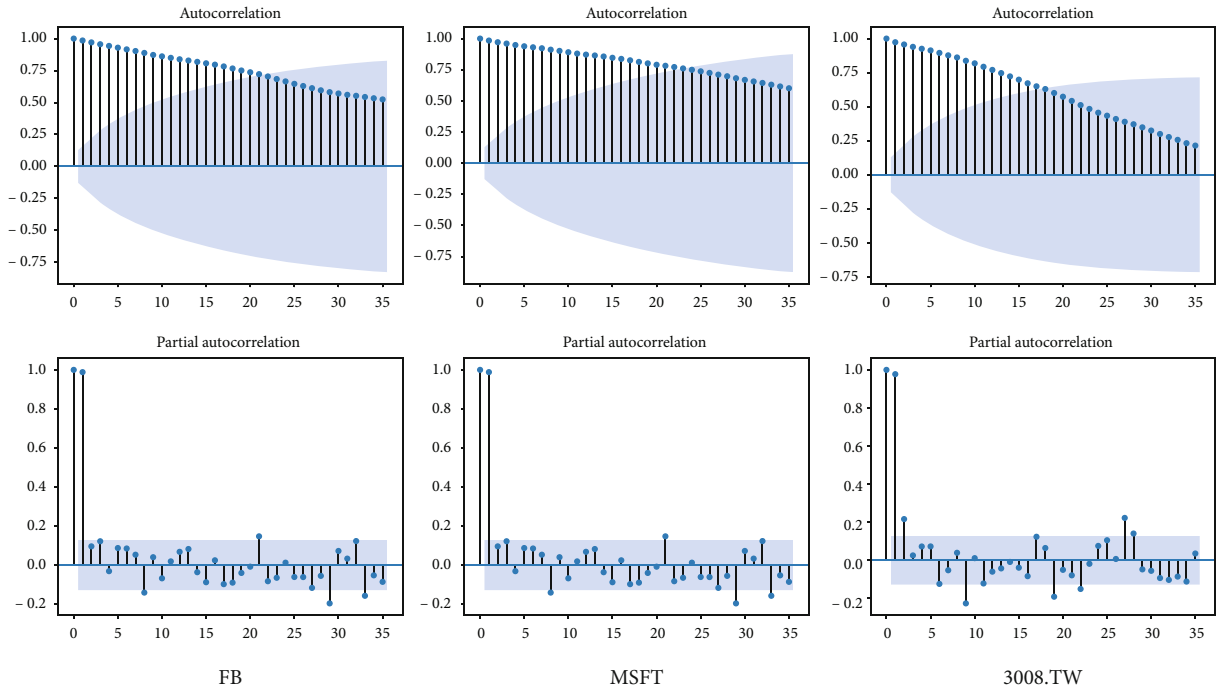


FIGURE 2: Examples of autocorrelation function graph and partial autocorrelation function graph.

extraction. It consisted of four convolution and four pooling layers. The fully connected part was used for classification and consisted of three fully connected layers. In addition, batch normalization and dropout layers are added to prevent overfitting of the model. The functions of each layer are as follows.

3.3.1. Input Layer. The input layer treats the input tensor as an image with shape $(d, m, 1)$. d is the number of the past trading days, representing the model using the data of the past d days for prediction. m is the dimension of the input feature.

3.3.2. Convolutional Layer. The convolution layer implements convolution operation on the input feature map to

obtain local information. The two-dimensional tensor $x^L \in \mathbb{R}^{H \times W^L}$ represents the L layer input of the CNN. $x_{u,v}^L$ is the element in row u and column v of this tensor. The convolution operations can be represented by Equation 3.

$$x_{u,v}^{L+1} = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} w_{i,j} \times x_{u+i,v+j}^L + b. \quad (3)$$

The convolution kernel of this layer is $w \in \mathbb{R}^{H \times W}$, $w_{i,j}$ is the value in row i column j of the convolution kernel, and b is the bias term. W and H are the width and height of the convolution kernel. The convolution kernels are generally square. Figure 4 shows an example of a convolution

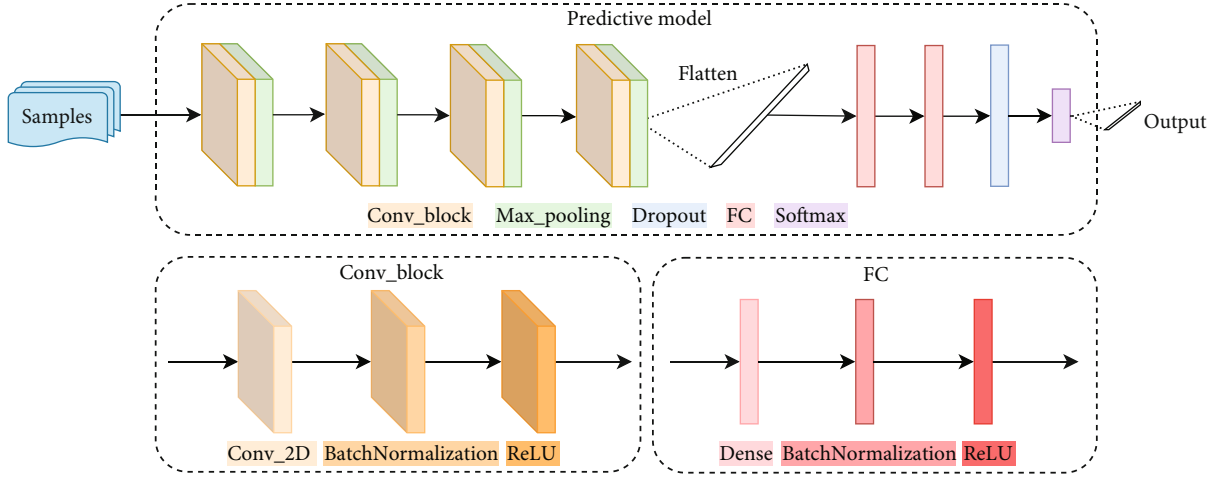


FIGURE 3: Model framework.

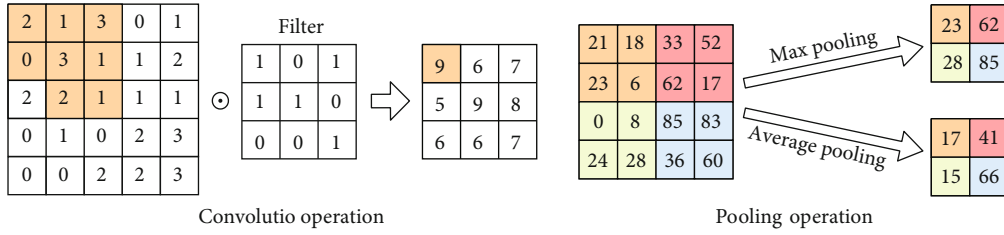


FIGURE 4: Examples of convolution and pooling operations.

operation with a bias of 0. In this example, $2 \times 1 + 1 \times 0 + 3 \times 1 + 0 \times 1 + 3 \times 1 + 1 \times 0 + 2 \times 0 + 2 \times 0 + 1 \times 1 = 9$. Similarly, other values can be obtained. The convolutional layer contains multiple convolutional kernels. After the convolution operation, each convolution kernel extracts a type of feature and generates a new feature map. The convolution layer is used to extract features in this way. After the convolution operation, the activation function is also required, which adds nonlinearity to the network so that the network can fit nonlinear problems. The activation function uses ReLU. The formula is shown in Equation 4.

$$\sigma(x) = \max(0, x). \quad (4)$$

3.3.3. Pooling Layer. It is also called the downsample layer. It mainly uses the pooling operation to select features. There are two kinds of pooling operations: max and average pooling. The example is shown in Figure 4. Max (average) pooling takes the maximum (average) of all values in the region covered by the pooling kernel as a result.

3.3.4. Fully Connected Layer. It integrates the local information extracted from the convolution layer or pooling layer and then maps the learned features to the sample label space. The operation in Equation (5) is carried out in the part of the full connection.

$$x_j^{L+1} = \sigma \left(\sum_{i=1}^N w_{i,j} \times x_i^L + b \right), \quad (5)$$

where σ is the activation function (ReLU, Softmax), i is a neuron in the L layer, L is a neuron in the $L + 1$ layer, x_i^L is the value of unit i , $w_{i,j}$ is the weight of neuron i connected to neuron j , and b is the bias term. The Softmax activation function is used for classification in the last fully connected layer. Equation (6) shows the formula.

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_{c=1}^C e^{x_c}}, \quad (6)$$

where x_i is the output value of neuron i , and C is the number of categories. It produces a value between 0 and 1, representing the probability of being classified as a class.

3.3.5. Batch Normalization Layer. Before executing the activation function, a batch normalization layer is added to improve the network performance. In 2015, Ioffe and Szegedy [48] first proposed batch normalization to alleviate the gradient dispersion problem of deep network, make the network more stable, and speed up the convergence speed of the model. Currently, it is widely used and has attracted extensive attention.

3.3.6. Dropout Layer. Dropout can effectively prevent overfitting. It will randomly discard some neurons during training to avoid the model falling into the optimal local solution. However, dropouts are not preferred in most networks after batch normalization is presented because Ioffe and Szegedy [48] have found that adding dropout to networks containing batch normalization often degrades network performance.

Therefore, Li et al. [49] recommended adding the dropout after all batch normalization layers. Therefore, this study adds a dropout layer before the output layer.

The method first processes the feature matrix into feature images and then the feature images are input into the prediction model for training and testing. The goal of model training is to minimize classification errors. The errors are expressed in Equation (7).

$$\text{loss} = - \sum_{c=1}^C y_c \times \log \hat{y}_c, \quad (7)$$

where C denotes the number of categories. For the training data, the Adam optimization algorithm was used to minimize the loss of all samples.

4. Experiments

This section presents the experimental setup and results. For stock forecasting models, profit is of significance. Therefore, simulated transactions were also conducted to evaluate the model's profitability. In addition, the forecasting model was compared to other trading strategies.

4.1. Experimental Settings. To analyze the influence of different input data on the prediction models, ten stocks were selected to conduct experiments, five of which are from the US market and five from the Taiwan market. In addition to historical data, the five US stocks have options, and the five Taiwan stocks have both options and futures. Data were collected from Yahoo Finance (<http://finance.yahoo.com>) and Taiwan Fubon Securities (<https://www.fubon-ebroker.com>). Options data are collected on each trading day for the 11 pairs of calls and puts closest to the spot price. The stock symbols and names of the companies are listed in Table 3.

After collecting and collating the data, a threshold setting method based on historical data was used to ensure that the sample categories of the training set were balanced. First, according to the change sequence in the past period, we find the upperbound and lowerbound such that the number of samples classified as rising, falling, and flat in this time range is approximately equal. Then, the new data are classified with the determined upperbound and lowerbound. Next, the data are marked and preprocessed using the aforementioned methods. The data were then split into training, validation, and testing sets. Owing to the characteristics of the time series, the splitting of the datasets is in chronological order. The first 80% were training sets, the next 10% were validation sets, and the last 10% were test sets. The price trend of the dataset is shown in Figures 5 and 6. The methods in this study used Python and Keras implementations.

The purpose of forecasting stock trends is to make a profit, so the profitability of the forecasting model is important. This study uses a simple trading strategy to buy stocks at low prices and then sell them at high prices. Algorithm 2 presents the pseudocode. If the stock price of the next trading day is predicted to rise, all funds are used to purchase

TABLE 3: Stock symbols and company names.

Symbol	Company name
AAPL	Apple Inc.
AMZN	Amazon.com, Inc.
FB	Meta Platforms, Inc.
IBM	International Business Machine Corporation
MSFT	Microsoft Corporation
1301.TW	Formosa Plastics Corporation (FPC)
2330.TW	Taiwan Semiconductor Manufacturing Company Limited (TSMC)
2357.TW	ASUSTeK Computer Inc. (ASUS)
2454.TW	MediaTek Inc. (MTK)
3008.TW	LARGAN Precision Co., Ltd. (LARGAN Precision)

stocks, and then the number of shares and remaining funds is calculated (Lines 3 to 5). The stocks are maintained until the stock price of the next trading day is predicted to decrease. Then, all shares are sold and the current funds are calculated (Lines 8 to 10). This process is repeated until the end of the period (Line 18). The total assets are calculated if the shares are still held (Lines 19 to 21). Finally, the rate of return is calculated (Line 22). Then, the buy-and-hold (B&H) and random choice (RC) trading strategies are used as the baseline to compare profit levels. The buy-and-hold strategy buys the stock at the beginning of the testing phase, holds it until the end of the testing phase, and sells it. The random choice strategy decides to buy, sell, or maintain the same probability.

4.2. Experimental Results. This section presents the results for ten stocks using different input data. After combining the different data types, there are seven types of input data for US stocks and 15 types for Taiwan stocks. Experiments were conducted on ten stocks using different inputs. Figures 7–10 show a comparison of the model accuracy and profit rate of different inputs, where “ H ” represents historical data, “ F ” represents futures, “ O ” represents the options, and “ T ” represents technical indicators. The other symbols are combinations of the four types of data. The experimental results show that the performance of the model and the return on the test set vary significantly with different inputs and the appropriate inputs are different for different stocks.

Overall, using technical indicators as input, predictive models can obtain better results for the five US stocks, especially AAPL. The US stock market is mature and dominated by institutional investors. Stock price is related to a company's performance and investor sentiment about the stock. In addition, stock price fluctuations are relatively stable, with no sharp rise or fall, and the price trend is relatively smooth. Technical indicators are useful tools for traders to make decisions. They can identify current trends more quickly and help predict future trends. Therefore, the technical indicators can improve the accuracy of the model.

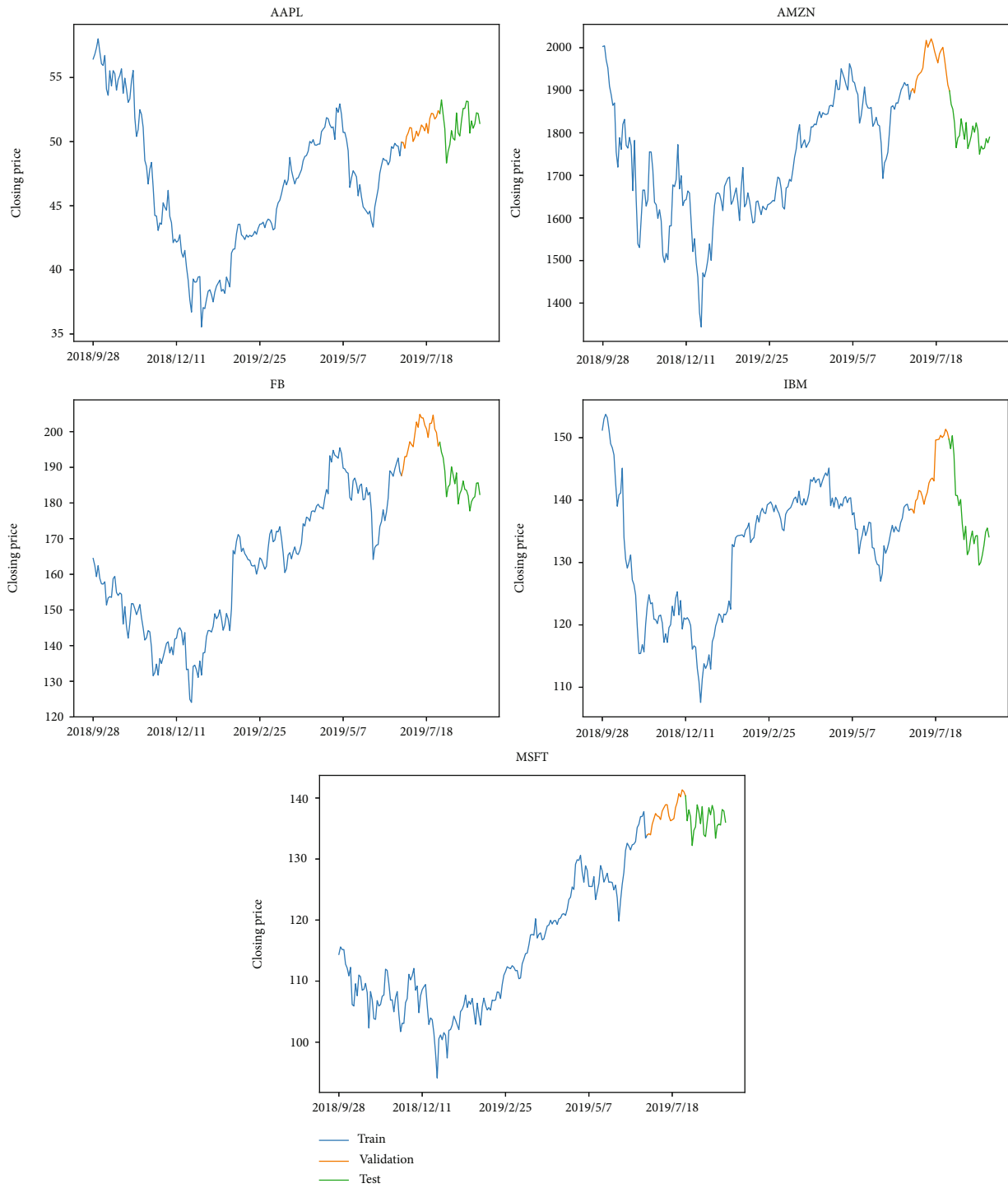


FIGURE 5: Closing price of five US stocks.

In five Taiwan stocks, using historical data and futures or futures, options, and technical indicators as input, the model's performance is better, with an accuracy of 0.4 or more. Taiwan's stock market is still immature, with more retail investors and speculators in the market. Moreover, stock prices fluctuate widely. Futures react to the expected stock prices by traders. If the trader is bullish on a stock, he

or she will buy a certain number of stock futures at a specific price and sell it at a high price when the stock price rises. However, if the trader is bearish about a stock, he/she will sell futures, and if the stock price falls, he/she buys it at a low price. When investors buy many futures, futures prices rise; conversely, when they sell many futures, futures prices fall. Therefore, futures price changes can signal stock price

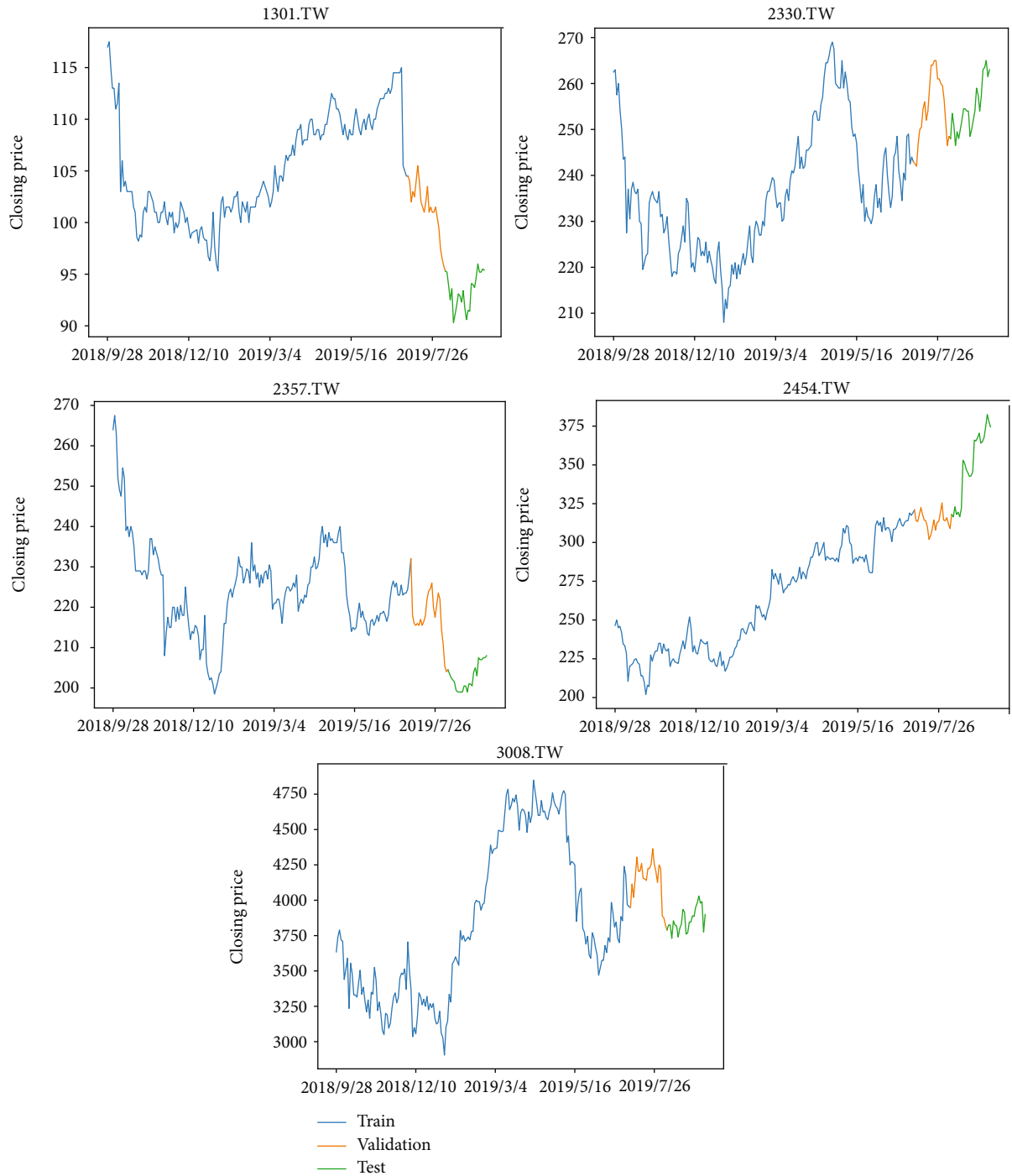


FIGURE 6: Closing price of five Taiwan stocks.

changes. Thus, using futures price movements as additional information can improve stock prediction performance.

Surprisingly, it is not always that the more the dimensions of the model's input data, the better the performance. For example, in the case of FB stocks, the accuracy of the model is lower than that of a single input when futures and options data are taken as input. A similar situation exists for the other stocks. There are two possible reasons for this.

First, the data may have included some unrelated information for the model. These may have added noises that could have affected the model's performance. Second, the amount of data used was small. In model training, the sample size was small, but the feature dimension was large, and the model overfits the training sample, thus reducing the generalization ability. Collecting more data or using data enhancement methods can alleviate this problem.

```

Input:  $Y = \{y_1, y_2, \dots, y_N\}$ : predict label;  $C = \{c_0, c_1, \dots, c_N\}$ : closing price;
Output: Profit rate: rate
1: initial init_money = 100000, money = init_money, i = 1, share = 0;
2: while  $i \leq N$  do
3:   if  $y_i == 1$  then
4:     share = money/ $c_{i-1}$ ;
5:     money = money - share  $\times c_{i-1}$ ;
6:      $j = i + 1$ ;
7:     while  $j \leq N$  do
8:       if  $y_j == -1$  then
9:         money = money + share  $\times c_{j-1}$ ;
10:        share = 0;
11:         $i = j$ ;
12:       else
13:          $j = j + 1$ ;
14:       end if
15:     end while
16:   end if
17:    $i = i + 1$ ;
18: end while
19: if share  $\neq 0$  then
20:   money = money + share  $\times c_N$ 
21: end if
22: rate = (money - init_money)/init_money

```

ALGORITHM 2: Simulate trades and calculate profit rate.

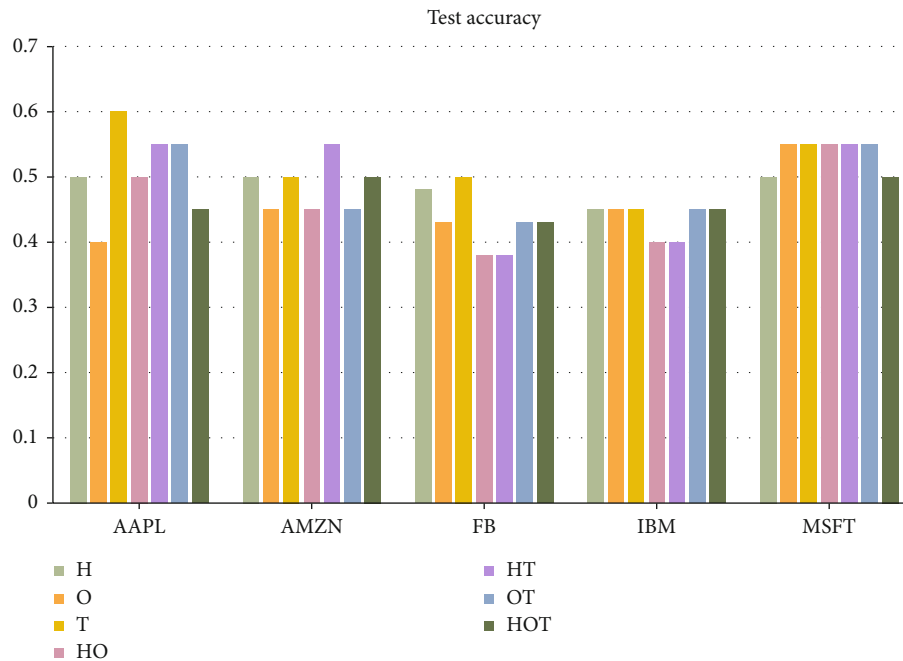


FIGURE 7: The accuracy of the CNN prediction model using different inputs in US stocks.

In terms of stock profits, the profit of the CNN prediction model is almost always better than the two baseline strategies. However, the model's profit with high prediction accuracy is not necessarily high. For example, in AMZN stock, using historical data and technical indicators, the model's prediction accuracy is the highest, reach-

ing 55%. Nevertheless, in simulated trading, the yield using only futures data was the highest. Using historical data and technical indicators on five stocks in the US and futures or options on five stocks in Taiwan, the income obtained is not necessarily the highest, but is relatively stable.

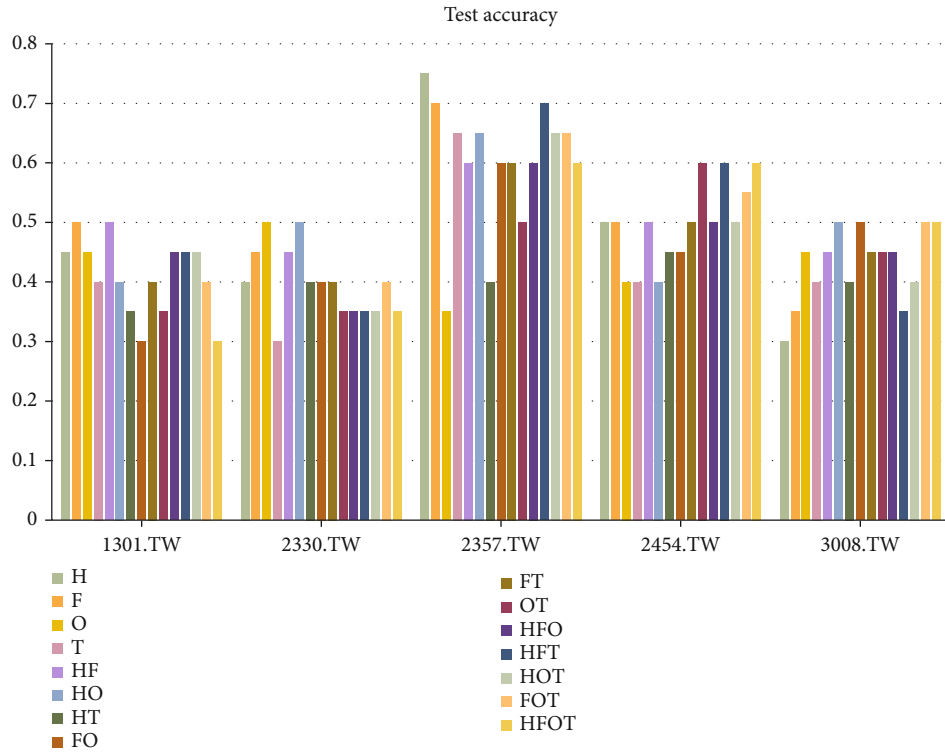


FIGURE 8: The accuracy of the CNN prediction model using different inputs in Taiwan stocks.

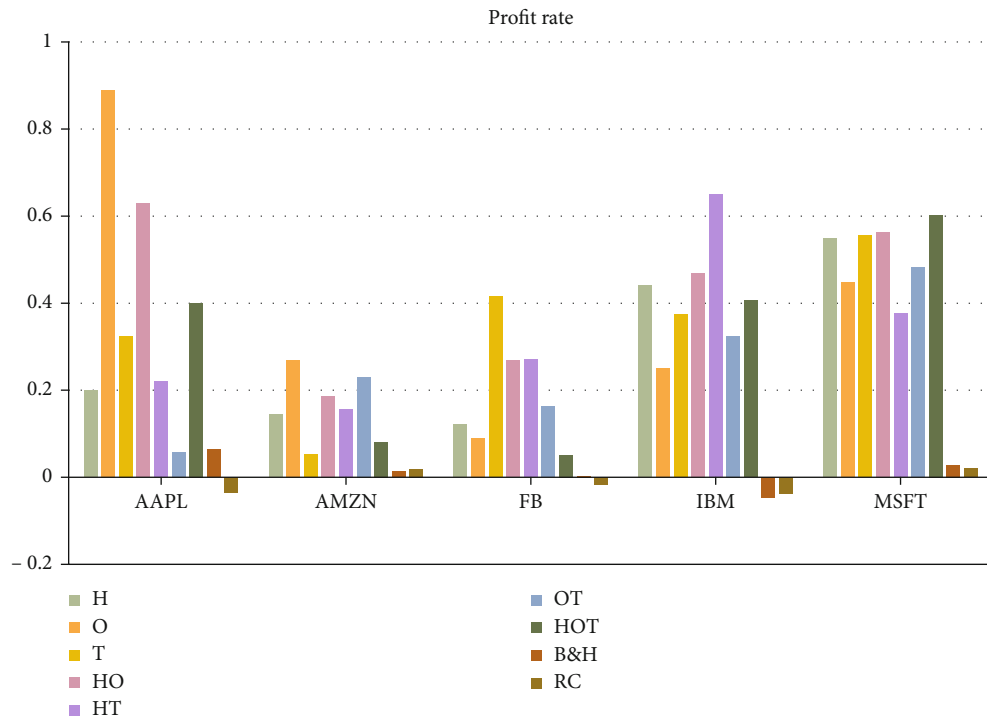


FIGURE 9: Profit rate of simulated tradings in US stocks.

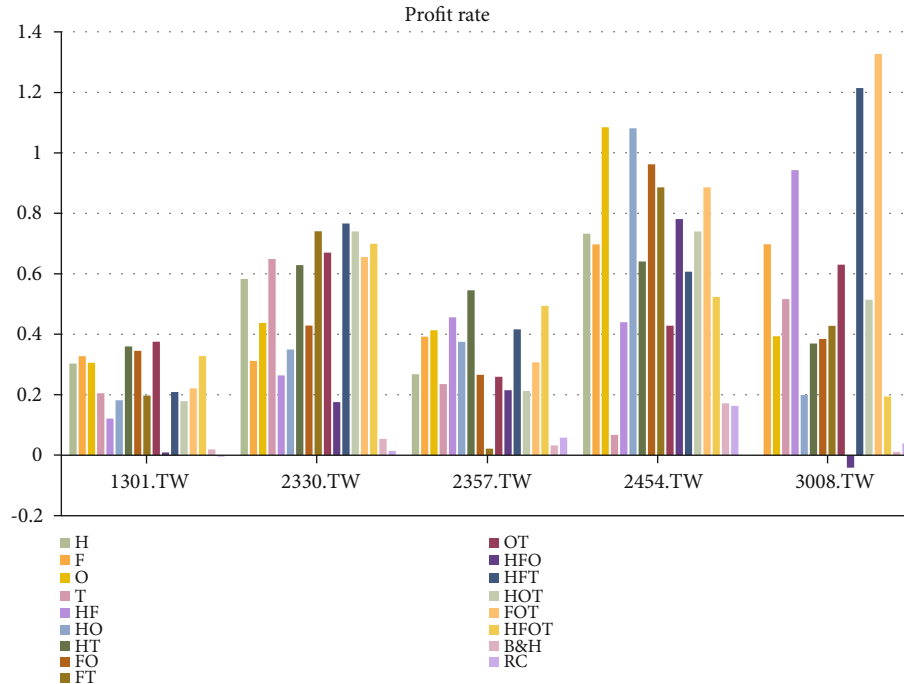


FIGURE 10: Profit rate of simulated tradings in Taiwan stocks.

5. Conclusion and Future Work

In this study, a three-class stock-forecasting model was proposed. By setting the rising and falling thresholds, stock prices were divided into three categories: up, down, and flat. The forecasting model used data from the past 30 trading days to predict whether the stock price volatility will exceed the threshold on the next trading day. If the change value of the stock price exceeds the threshold, a signal will be given to the investor to help in their decision. Subsequently, this study used historical data, leading indicators, and technical indicators as the model's input and analyzed the influence of different input data on the model. Experiments were conducted on ten stocks in the US and Taiwan stock markets. The results show that future stock trends can be predicted using technical index data. In the US market, the accuracy of the model can be improved by adding technical index data. In the Taiwanese market, futures data are even more helpful. In addition, the simulation transaction experiment further proves that the prediction model is profitable.

In the future, the structure of the model can be improved to improve the prediction accuracy further and provide more reliable results for investors. In addition, it is not always that more data added will lead to better prediction. Additional data may introduce noise or overfitting in the models. Therefore, feature selection methods can be used to select important features. Additionally, in the trading simulation, all funds are used to buy or sell all the stocks. Designing a better trading strategy to help investors determine the share of each transaction can result in a higher, more stable return. Moreover, a complete trading system can be built by combining a predictive model with trading strategies and risk assessments.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the Shandong Provincial Natural Science Foundation (ZR201911150391).

References

- [1] G. T. Reddy, R. Kaluri, P. K. Reddy, K. Lakshmana, S. Koppu, and D. S. Rajput, "A novel approach for home surveillance system using IoT adaptive security," in *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM)*, Jaipur-India, 2019 Amity University Rajasthan.
- [2] S. A. Kumar, T. Vealey, and H. Srivastava, "Security in internet of things: challenges, solutions and future directions," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pp. 5772–5781, IEEE, Koloa, HI, USA, 2016.
- [3] C. M. Chen, X. Deng, W. Gan, J. Chen, and S. Islam, "A secure blockchain-based group key agreement protocol for IoT," *The Journal of Supercomputing*, vol. 77, no. 8, pp. 9046–9068, 2021.
- [4] M. A. Khan, I. Ullah, A. Alkhalifah et al., "A provable and privacy-preserving authentication scheme for UAV-enabled intelligent transportation systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, 2021.
- [5] S. A. Chaudhry, A. Irshad, J. Nebhen et al., "An anonymous device to device access control based on secure certificate for

- Internet of medical things systems,” *Sustainable Cities and Society*, vol. 75, article 103322, 2021.
- [6] X. Huang, H. Xiong, J. Chen, and M. Yang, “Efficient revocable storage attribute-based encryption with arithmetic span programs in cloud-assisted Internet of Things,” *IEEE Transactions on Cloud Computing*, 2021.
 - [7] C. M. Chen, X. Li, S. Liu, M. E. Wu, and S. Kumari, “Enhanced authentication protocol for the Internet of Things environment,” *Security and Communication Networks*, vol. 2022, article 8543894, 13 pages, 2022.
 - [8] H. Zeng, “Influences of mobile edge computing-based service preloading on the early-warning of financial risks,” *The Journal of Supercomputing*, vol. 78, pp. 11621–11639, 2022.
 - [9] D. Choi and K. Lee, “An artificial intelligence approach to financial fraud detection under IoT environment: a survey and implementation, security and communication,” *Networks*, vol. 2018, article 5483472, 15 pages, 2018.
 - [10] I. V. Pustokhina, D. A. Pustokhin, S. N. Mohanty, P. A. G. García, and V. García-Díaz, “Artificial intelligence assisted Internet of Things based financial crisis prediction in FinTech environment,” *Annals of Operations Research*, pp. 1–21, 2021.
 - [11] M. Al Duhayyim, H. Alsolai, F. N. Al-Wesabi et al., “Optimized stacked autoencoder for IoT enabled financial crisis prediction model,” *CMC-Computers Materials & Continua*, vol. 71, no. 1, pp. 1079–1094, 2022.
 - [12] H. Li, J. Hua, J. Li, and G. Li, “Stock forecasting model FS-LSTM based on the 5G Internet of Things,” *Wireless Communications and Mobile Computing*, vol. 2020, 7 pages, 2020.
 - [13] M. F. Anaghi and Y. Norouzi, “A model for stock price forecasting based on ARMA systems,” in *2012 2nd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)*, pp. 265–268, IEEE, Beirut, Lebanon, 2012.
 - [14] 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, IEEE, Cambridge, UK, 2014.
 - [15] P. H. Franses and D. Van Dijk, “Forecasting stock market volatility using (non-linear) Garch models,” *Journal of Forecasting*, vol. 15, no. 3, pp. 229–235, 1996.
 - [16] D. Alberg, H. Shalit, and R. Yosef, “Estimating stock market volatility using asymmetric Garch models,” *Applied Financial Economics*, vol. 18, no. 15, pp. 1201–1208, 2008.
 - [17] C. N. Babu and B. E. Reddy, “Selected indian stock predictions using a hybrid ARIMA-Garch model,” in *2014 International Conference on Advances in Electronics Computers and Communications*, pp. 1–6, IEEE, Bangalore, India, 2014.
 - [18] C. N. Babu and B. E. Reddy, “Prediction of selected Indian stock using a partitioning-interpolation based ARIMA-Garch model,” *Applied Computing and Informatics*, vol. 11, no. 2, pp. 130–143, 2015.
 - [19] R. K. Nayak, D. Mishra, and A. K. Rath, “A Naive SVM-KNN based stock market trend reversal analysis for Indian benchmark indices,” *Applied Soft Computing*, vol. 35, pp. 670–680, 2015.
 - [20] M. Subha and S. T. Nambi, “Classification of stock index movement using k-nearest neighbours (k-NN) algorithm,” *WSEAS Transactions on Information Science and Applications*, vol. 9, no. 9, pp. 261–270, 2012.
 - [21] M. Kumar and M. Thenmozhi, “Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models,” *International Journal of Banking, Accounting and Finance*, vol. 5, no. 3, pp. 284–308, 2014.
 - [22] Y. Musa and S. Joshua, “Analysis of ARIMA-artificial neural network hybrid model in forecasting of stock market returns,” *Asian Journal of Probability and Statistics*, vol. 6, no. 2, pp. 42–53, 2020.
 - [23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
 - [24] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
 - [25] M. U. Gudelek, S. A. Boluk, and A. M. Ozbayoglu, “A deep learning based stock trading model with 2-D CNN trend detection,” in *2017 IEEE symposium series on computational intelligence (SSCI)*, pp. 1–8, IEEE, Honolulu, HI, USA, 2017.
 - [26] D. M. Nelson, A. C. Pereira, and R. A. de Oliveira, “Stock market’s price movement prediction with LSTM neural networks,” in *2017 International joint conference on neural networks (IJCNN)*, pp. 1419–1426, IEEE, Anchorage, AK, USA, 2017.
 - [27] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, and K. Soman, “Stock price prediction using LSTM, RNN and CNN-sliding window model,” in *2017 International conference on advances in computing, communications and informatics (ICACCI)*, pp. 1643–1647, IEEE, Udupi, India, 2017.
 - [28] Y. C. Chen and W. C. Huang, “Constructing a stock-price forecast CNN model with gold and crude oil indicators,” *Applied Soft Computing*, vol. 112, article 107760, 2021.
 - [29] S. Mohan, S. Mullanpudi, S. Sammeta, P. Vijayvergia, and D. C. Anastasiu, “Stock price prediction using news sentiment analysis,” in *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 205–208, IEEE, Udupi, India, 2019.
 - [30] J. Kordonis, S. Symeonidis, and A. Arampatzis, “Stock price forecasting via sentiment analysis on Twitter,” in *Proceedings of the 20th Pan-Hellenic Conference on Informatics*, pp. 1–6, ACM, New York, 2016.
 - [31] N. Oliveira, P. Cortez, and N. Areal, “The impact of microblogging data for stock market prediction: using Twitter to predict returns, volatility, trading volume and survey sentiment indices,” *Expert Systems with Applications*, vol. 73, pp. 125–144, 2017.
 - [32] Y. Kara, M. A. Boyacioglu, and Ö. K. Baykan, “Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul stock exchange,” *Expert Systems with Applications*, vol. 38, no. 5, pp. 5311–5319, 2011.
 - [33] M. Ballings, D. Van den Poel, N. Hespeels, and R. Gryp, “Evaluating multiple classifiers for stock price direction prediction,” *Expert Systems with Applications*, vol. 42, no. 20, pp. 7046–7056, 2015.
 - [34] Y. Fang, K. Fataliyev, L. Wang, X. Fu, and Y. Wang, “Improving the genetic-algorithm-optimized wavelet neural network for stock market prediction,” in *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 3038–3042, IEEE, Beijing, China, 2014.
 - [35] L. P. Ni, Z. W. Ni, and Y. Z. Gao, “Stock trend prediction based on fractal feature selection and support vector machine,” *Expert Systems with Applications*, vol. 38, no. 5, pp. 5569–5576, 2011.
 - [36] Y. Lin, H. Guo, and J. Hu, “An SVM-based approach for stock market trend prediction,” in *The 2013 International Joint*

- Conference on Neural Networks (IJCNN)*, pp. 1–7, IEEE, Dallas, TX, USA, 2013.
- [37] M. Göçken, M. Özçalıcı, A. Boru, and A. T. Dosdoğru, “Integrating metaheuristics and artificial neural networks for improved stock price prediction,” *Expert Systems with Applications*, vol. 44, pp. 320–331, 2016.
- [38] K. K. Yun, S. W. Yoon, and D. Won, “Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process,” *Expert Systems with Applications*, vol. 186, article 115716, 2021.
- [39] H. Gunduz, Y. Yaslan, and Z. Cataltepe, “Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations,” *Knowledge-Based Systems*, vol. 137, pp. 138–148, 2017.
- [40] 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.
- [41] F. Kamalov, “Forecasting significant stock price changes using neural networks,” *Neural Computing and Applications*, vol. 32, no. 23, pp. 17655–17667, 2020.
- [42] J. M. T. Wu, Z. Li, G. Srivastava, M. H. Tasi, and J. C. W. Lin, “A graph-based convolutional neural network stock price prediction with leading indicators,” *Software: Practice and Experience*, vol. 51, no. 3, pp. 628–644, 2021.
- [43] J. M. T. Wu, L. Sun, G. Srivastava, and J. C. W. Lin, “A long short-term memory network stock price prediction with leading indicators,” *Big Data*, vol. 9, no. 5, pp. 343–357, 2021.
- [44] J. M. T. Wu, Z. Li, N. Herencsar, B. Vo, and J. C. W. Lin, “A graph-based CNN-LSTM stock price prediction algorithm with leading indicators,” *Multimedia Systems*, pp. 1–20, 2021.
- [45] X. Ding, Y. Zhang, T. Liu, and J. Duan, “Deep learning for event-driven stock prediction,” in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, United States, 2015 AAAI.
- [46] J. Yoo, Y. Soun, Y. C. Park, and U. Kang, “Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2037–2045, ACM SIGKDD, New York, 2021.
- [47] A. Hull, “How to reduce lag in a moving average,” *Report Prepared by Alan Hull*<https://alanhull.com/hull-moving-average>.
- [48] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, pp. 448–456, PMLR, Lille, France, 2015.
- [49] X. Li, S. Chen, X. Hu, and J. Yang, “Understanding the disharmony between dropout and batch normalization by variance shift,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2682–2690, CVPR, California, 2019.