Proceedings of the 2nd International Conference on Financial Technology and Business Analysis
DOI: 10.54254/2754-1169/45/20230281

Predicting Tencent's Stock Price: A Comparative Analysis of Machine Learning Algorithms

Han Li[1,a,*]

[1]*The University of Sydney, Sydney, 2006, Australia*
a. hali4378@uni.sydney.edu.au
**corresponding author*

Abstract: Chinese internet companies, such as Tencent, are bringing a new energy to the global market. As one of the largest internet companies in China, Tencent's stock price fluctuations hold significant importance for investors and the market. Accurate forecasting of stock prices is of great importance to make high-yield decisions and manage risk. This paper aims to provide investors with effective and scientific references for decision-making by analyzing various forecasting methods. The dataset China-Techgiant-Stock-Data-in-HK-Market-2022 from Kaggle is used in this paper. This study utilizes various machine learning and time series analysis methods, including linear regression, SVM, random forest, LSTM neural network, and ARIMA, to predict stock prices using historical data. The models' accuracy and performance are compared, with traditional machine learning methods (linear regression, SVM, and random forest) contrasted against deep learning and time series analysis methods (LSTM neural network and ARIMA). It turns out that LSTM has the best prediction results. This can better guide investors in their stock decisions.

Keywords: Tencent, stock price forecasting, machine learning

1. Introduction

Recently, stock price prediction has already become possible growing amount of Internet data and the development of computer tech [1]. Accurate forecasting can help investors make better decisions [2], and stock price forecasting is important for business decisions because stock prices reflect the value and future trends of a company. However, it is still a challenging task because stock prices are affected by many complex factors.

There are many approaches to stock price forecasting, such as traditional time series forecasting methods like ARIMA [3], machine learning based methods like SVM [4] and Random Forest, deep learning based methods like LSTM [5] neural networks and the simplest and most basic linear regression. Although time series methods are very widely used, they often have difficulty capturing the complex and dynamic factors that influence stock price fluctuations. Machine learning-based methods have also been studied, but they often require manual selection of features and parameters, which can affect the accuracy of predictions [6, 7]. In contrast, cutting-edge approaches such as LSTM neural networks have demonstrated remarkable proficiency in navigating the nonlinear terrain of stock price fluctuations and seizing upon the multifaceted factors that influence them.

This study aims at comparing different methods for stock price forecasting and identify the most effective method. We will use daily stock price data of Tencent, abstracted from the dataset China-

DOI: 10.54254/2754-1169/45/20230281

Techgiant-Stock-Data-in-HK-Market-2022 from Kaggle [8], and compare the performance of Linear Regression, SVM, Random Forest, LSTM neural networks, and ARIMA in predicting future stock prices(close price in each day). The research will utilize normalization techniques to preprocess the data, thereby enhancing the accuracy of the model's predictions. Python will be the programming language used for building the model, and open source tool libraries such as Tensorflow and Scikit-Learn will be combined for this purpose. Various evaluation metrics will be used in this study, including but not limited to comparison using mean absolute error, mean squared error and mean log squared error Finally, this study will give a comparison and analysis of the prediction results for different methods and suggest the optimization of prediction for different methods.

Through a comparative analysis of the prediction results generated by four different methods, this study has concluded that the LSTM neural network outperforms the traditional methods in terms of predictive accuracy., LSTM can better capture the nonlinear relationship of stock prices and improve the prediction accuracy. Therefore, investors and companies should give priority to the LSTM neural network method when making stock price predictions.

The upcoming sections of this will be presented in the following order: In Section 2, a detailed exposition of the model and methodology employed in this study for stock forecasting purposes. In Section 3, an overview of the data utilized in this research will be provided. In Section 4, the outcomes of the applied method will be presented along with a thorough analytical justification for the chosen approach. Finally, this paper conclude by describing our results and contributions. By following this structure, this paper seeks to offer an exhaustive and unified examination of the stock forecasting technique and its outcomes.

2. Method

2.1. Linear Regression

As a relatively simple yet widely applicable method, linear regression inherently models stock prices and diverse potential influencing factors to yield a basic yet versatile approach eminent for stock forecasting [9].

The general formula for a linear regression model is:

$$y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n \tag{1}$$

where y is the target variable(in this case, the stock's closing price); $x_1, x_2, \ldots, x_n$ are the input variables (in this case, the stock's open price, highest price, lowest price, and volume for the day); $b_0, b_1, b_2, \ldots, b_n$ are the coefficients of the model that under estimated.

The primary objective of the linear regression algorithm is to calculate the coefficient values that minimize the disparity between the predicted closing prices and actual closing prices in the training dataset. The model can be made to predict new data by entering the corresponding value of open_price, and after the data training results process is complete, highest_price, lowest_price, and volume for the given day.To use linear regression to predict stock prices, one first need to collect stock prices and analyze the factors that may affect stock prices. Then the interface and methods provided by statistical software can be used to train the linear regression model based on the obtainded data. This can be done using the scikit-learn package, the "LinearRegression()" function. This is shown in Figure 1. Flow of linear regression for predicting the price of a stock

DOI: 10.54254/2754-1169/45/20230281

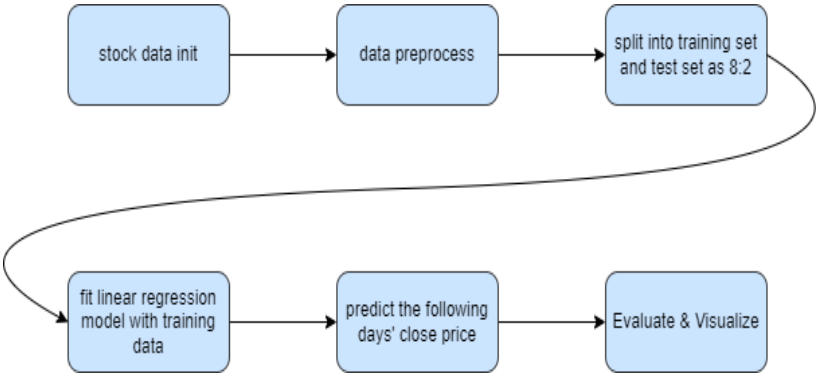

Figure 1: Flow chart of linear regression to predict stock price.

2.2. Support Vector Machine

Since the problem studied in this paper is a stock prediction problem, which is essentially a regression problem, the specific application of Support Vector Machin is Support Vector Regression [4].

SVR is a machine learning algorithm well suited for stock price prediction because it can capture nonlinear relationships using kernel functions and is robust to outliers. It can model complex relationships flexibly by using different kernel functions and thus. Also, the optimization of the minimum generalization error reduces the possibility of overfitting and is able to make good predictions for unseen data.

The goal of SVR is to find a function that maps input features (e.g., stock prices, economic indicators) to a continuous target variable (e.g., the next day's stock price). Methodology for linear combinations of kernel functions evaluated on the SVM using the form of functions:

$$f(x) = \sum_i(\alpha_i * k(x_i, x)) + b \tag{2}$$

The predicted value of the target variable for a specific input value x, denoted by f(x), is calculated using the support vector machine (SVM) algorithm. The importance of each support vector x_i is determined by the respective KKT, a_i. The LIN, K(x_i, x), measures the similarity between the support vectors x_i and the input x. Linear, polynomial, and radial basis function (RBF) are among the frequently employed kernel functions. The bias term, b, is incorporated in the SVM algorithm to account for the offset between the predicted values and the origin.

The aim of SVR is to minimize the discrepancy between the prediction and groundtruth of the target variable, while satisfying a tolerance margin ϵ. The optimization problem can be formulated as follows:

Minimize:

$$\frac{1}{2}\sum_i \sum_j a_i a_j K(x_i, x_j) - \sum_i a_i y_i \tag{3}$$

subject to:

$$\sum_i a_i y_i = 0 \tag{4}$$

$$0 \le a_i \le C \tag{5}$$

$$y_i - f(x_i) \le \varepsilon \tag{6}$$

$$f(x_i) - y_i \le \varepsilon \tag{7}$$

The proposed prediction price of stock model using Support Vector Regression (SVR) involves preprocessing and splitting historical data into training and testing sets. Technical indicators are selected as features based on past time series data. The model's kernel function and hyperparameters

DOI: 10.54254/2754-1169/45/20230281

are chosen by search method such as grid or random search to prevent overfitting and underfitting. The SVR model is trained and evaluated using metrics on the testing set, with adjustments made to the hyperparameters if necessary. After its development, the final model is utilized for predicting stock prices on new data, and its efficacy is evaluated by assessing its performance on unseen data.The flow of SVM to predict stock prices is shown in Figure 2.

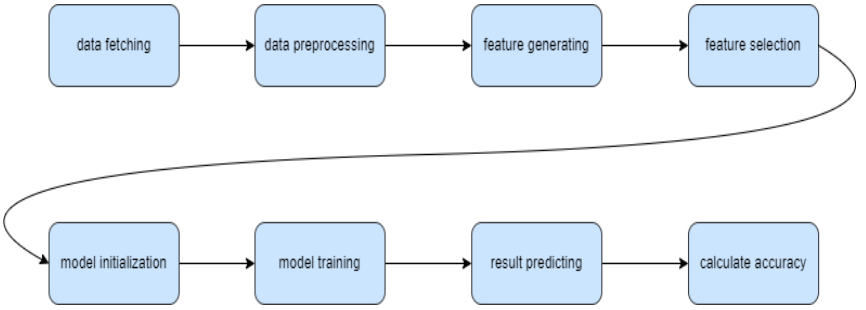

Figure 2: Flow chart of SVM to predict stock price.

2.3. Random Forest

Random forests, a decision tree-based integrated learning algorithm, is well-suited for handling large-scale, high-dimensional datasets, such as those encountered in stock price prediction. By automatically selecting important features, utilizing random sampling during tree training, and effectively handling nonlinear relationships and interaction effects, random forests are a promising solution for this critical task [10].

The general formula of the random forest model is as shown below.

$$y = f(x) + \varepsilon \tag{8}$$

where y is the shown number(i.e., desired predicted target value); x is the input number (i.e., the set of variables we use to make predictions); The random forest model, denoted by f, is an amalgamation of a decision tree constructed using a random subset of the training data and input variables. The error term, ε, accounts for the stochastic variability in y that is not accounted for by the model.

The preparation of data is crucial in stock prediction. Historical data, market indicators, and relevant information are divided into training and test sets chronologically. Time series feature selection involves extracting data from past time to calculate technical indicators like 'MA5', 'MA10', 'RSI', 'MOM', 'EMA12', 'MACD', 'MACDsignal', and 'MACDhist'. The RandomForestRegressor model from the sklearn library is used for training as it handles high-dimensional features and has good generalization capabilities. The model's accuracy is evaluated by predicting on the test set, allowing for optimization and adjustment. The flow of Random Forest to predict stock prices is shown in Figure 3.

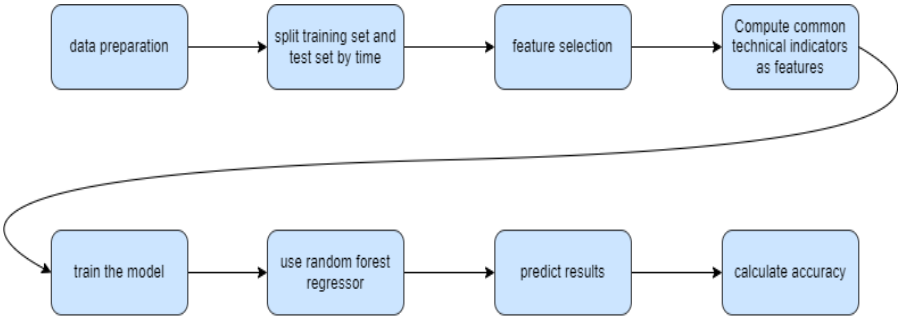

Figure 3: Flow chart of random forest to predict stock price.

DOI: 10.54254/2754-1169/45/20230281

2.4. Long Short-Term Memory

The LSTM is a type of RNN that incorporates a mechanism for selectively remembering or forgetting past information, making it well-suited for modeling sequential data, which is well-suited for predicting stock prices owing to its capacity to model long-term dependencies, capture non-linear relationships, and handle sequences of varying lengths. LSTM's scalability allows it to handle large datasets, and its performance has been demonstrated to outperform conventional models, making it a strong contender for stock price prediction [5].

At the core of LSTM is the concept of "gates," which are used to control the flow and retention of information within the network.LSTM is comprised of three gates, namely input gate, forget gate, and output gate, which regulate information flow in the LSTM cell, enabling selective reception, deletion, or output of information. The following section provides an explanation of the formulas used in LSTM.

Within a typical LSTM cell, each gate is composed of a weight matrix and a bias vector. Let x_t denote the value of the current time step and h_{t-1} denote the non-displayed state of the former time step.The output of the input gate, forget gate, and output gate are represented as i_t, f_t, and o_t respectively. The new cell state is represented as c_t, while hidden state expressed as h_t. Calculations for each of these values are as follows:

Input gate.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{9}$$

Forgetting gate.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{10}$$

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_C) \tag{11}$$

Output gate.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{12}$$

Long-term memory.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{13}$$

Short-term memory.

$$h_t = o_t * tanh(C_t) \tag{14}$$

The candidate value for updating the cell state, C_t, is represented by $C̃_t$.

To evaluate the effectiveness of predicting stock prices using LSTM models, stock data from the stock exchange was obtained and preprocessed to ensure data quality. An LSTM model was then utilized to process sequential data, predicting stock prices for day n+1 based on the previous n days (where n=26). An 80:20 ratio was used to divide the dataset into a training set and a test set, which means it comprised the first 80% of days' stock price data, while the test set comprised the last 20% of days' stock price data. The LSTM model underwent 100 epochs of training and was subsequently evaluated using both the training and test set, with predicted stock prices compared to actual stock prices to assess the model's performance. This approach provides an empirical basis for evaluating the efficacy of LSTM models in predicting stock prices. Figure 4 illustrates the flow of LSTM to predict stock prices.

DOI: 10.54254/2754-1169/45/20230281

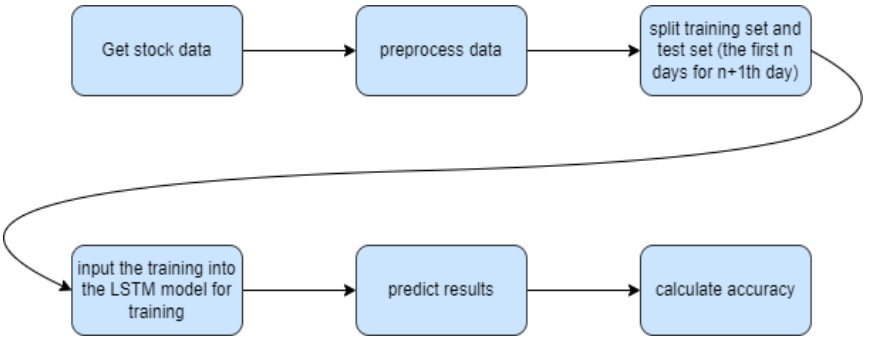

Figure 4: Flow chart of LSTM to predict stock price.

2.5. AutoRegressive Integrated Moving Average

ARIMA is a technique for analyzing data of time series, especially for forecasting future stock prices. Its flexibility enables modeling of both linear and nonlinear relationships, as well as incorporating exogenous variables to improve forecast accuracy. ARIMA models can handle non-stationary data by differencing, capturing trend and seasonal patterns. Its widespread use in finance and economics provides ample resources for implementation and interpretation. Overall, ARIMA is a powerful and adaptable tool for forecasting stock prices based on historical data [3].

The ARIMA(p, d, q) model can be represented by the following general formula:

$$Y[t] = c + \emptyset[1]Y[t-1] + \emptyset[2]Y[t-2] + \cdots + \emptyset[p]Y[t-p] + \varepsilon[t] - \theta[1]\varepsilon[t-1] - \theta[2]\varepsilon[t-2] - \cdots - \theta[q]\varepsilon[t-q] \quad (15)$$

where $Y[t]$ is the value of the time-series at time t. c is a constant (or intercept) term. $\emptyset[1]$, $\emptyset[2]$, ..., $\emptyset[p]$ are the autoregressive (AR) coefficients. $\varepsilon[t]$ is the white noise error term at time t. $\theta[1]$, $\theta[2]$, ..., $\theta[q]$ are the MA coefficients. p is the order of the AR process (the number of past values of Y to include in the model). q is the order of the MA process (the number of past errors included in the sample). The order of the disparity is d (making it stationary, depending on the number of disparities,).

First step involves data preparation, where the closing price of each trading days is collected form all historical stock data. The data should be carefully examined for completeness and the absence of any missing or anomalous values.

Subsequently, a test for stationarity is performed by constructing a plot of the close price trend, as shown Figure 5 below, revealing that the stock's opening price exhibits volatility therefore a unsteady time sequences. To eliminate the impact of random trends on modeling stock data, differencing is employed to remove the random trend.

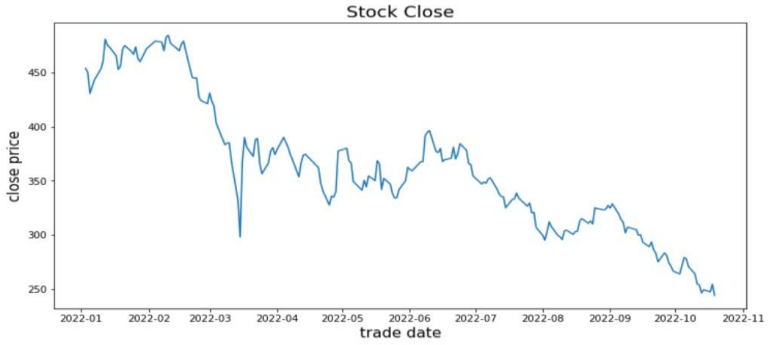

Figure 5: General trend of close prices.

DOI: 10.54254/2754-1169/45/20230281

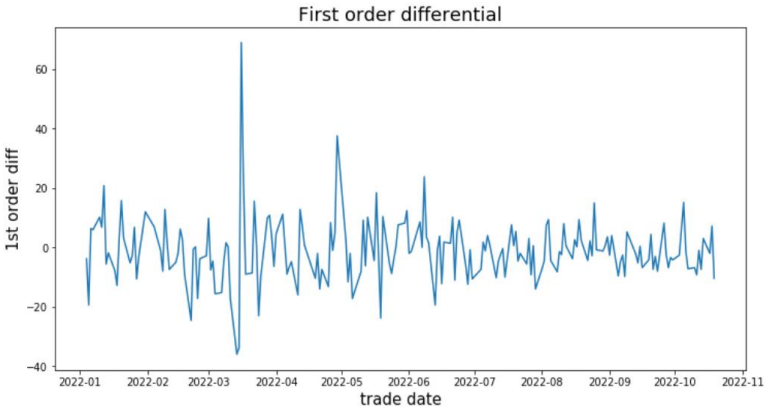

Figure 6: First order differential of close price.

The differenced time series plot, Figure 6, indicates that the random trend of the stock's opening price has become stationary.

Both ACF and PACF plots were analysed. It was possible to determine the appropriate ARIMA model that PACF assesses the correlation between lagged versions of the time series its own, and ACF quantifies the bias between a point in the time series and a different lagged point.taking into account the influence of all shorter lag periods. After examining these two plots, an ARIMA (1, 1, 1) model was selected, as depicted in Figure 7 and Figure 8.

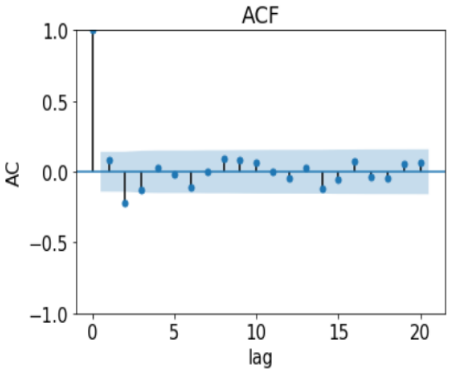

Figure 7: ACF plot.

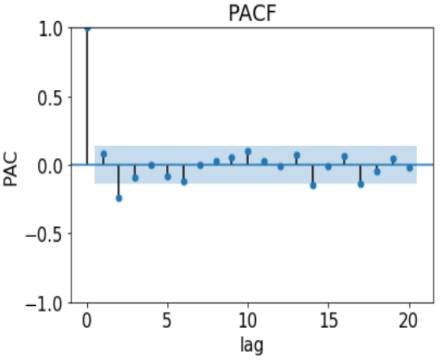

Figure 8: PACF plot.

Finally, model validation is performed by examining the autocorrelation of residuals and conducting a normality test. The analysis shows that the residuals exhibit little autocorrelation and

DOI: 10.54254/2754-1169/45/20230281

conform to a normal distribution, indicating that the model is valid. With the completion of model validation, the model can be utilized to make predictions.

The entire flow chart is shown Figure 9 below.

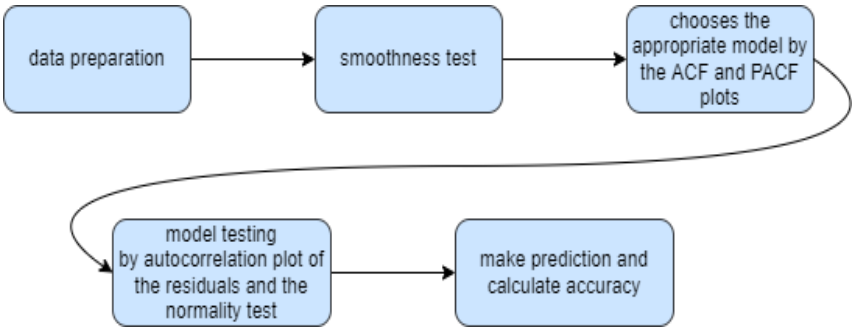

Figure 9: Flow chart of ARIMA to predict stock price.

3. Dataset

As mentioned earlier, the stock price data of Tencent is abstracted from the dataset China-Techgiant-Stock-Data-in-HK-Market-2022 from Kaggle [8], which provides daily trading information for representative technology companies in Hong Kong, including Tencent, for the year 2022. The analysis focuses on predicting the closing price, which is considered the most comprehensive evaluation of a stock's performance and a stable target for prediction due to high trading volume. This indicator reflects investors' assessment of the stock's trading status and market conditions throughout the entire trading day, summarizing its overall performance despite price fluctuations [11].

Only data from all trading days in this dataset are considered in this study and are treated as temporally continuous series. The trend and first-order difference of Tencent's stock price data in the dataset have been shown in Figure 5 and Figure 6.

In this study, the dataset is partitioned into a training set and a test set. The training set contains the stock price data for the first 80% of days and the test set contains the stock price data for the last 20%. However, the different methods used in this study treat the training and test data slightly differently, and detailed description of these differences has been provided in the preceding section.

In addition, this study calculates a statistical description of the Tencent stock data, including sample size, mean, standard deviation, maximum value, minimum value, median, shown in Table 1.

Table 1: Stock data statistics.

Sample	mean	std	min	max	median
246	343.869106	67.025239	200.8	484.4	335.3

4. Results

In this paper, the predicted stock price values of the five methods are visualized against the true values in Figure 10-14, and residual plots are calculated in Figure 15-19, for linear regression, SVM, random forest, LSTM neural network and ARIMA.

The prediction results of the five methods are shown in Figure 11-15, which can be observed that the linear regression method produced poor prediction results with a significant difference from the true value. The SVM method provided relatively consistent results with the actual price fluctuations. The random forest method could predict the stock price trend better but struggled with predicting exact closing prices. The LSTM neural network produced results that were closest to the actual value, although it faced difficulties in predicting high-frequency price fluctuations. The ARIMA method

DOI: 10.54254/2754-1169/45/20230281

had poor prediction results and could not forecast the stock's rise or fall. The residual plots of the five methods also corresponded to their respective prediction results, with the LSTM neural network exhibiting the smallest mean and variance of residuals.

In this project, the root mean square error (RMLSE), mean squared error (MSE), and mean absolute error (MAE) are used to quantitatively compare and evaluate the prediction results of the five methods, as presented in Table 2. The LSTM neural network showed the best predictive performance, followed by the random forest and SVM methods, with linear regression and ARIMA methods performing the poorest.

The superior performance of LSTM neural networks in stock price prediction can be attributed to their ability to capture the temporal dynamics, nonlinear and long-term dependencies within the data. The recurrent neural network structure of LSTM allows for modeling dynamic changes in historical data, leading to more accurate forecasting. SVM and random forest models also exhibit strong predictive power due to their capacity to capture complex nonlinear relationships and interactions. However, LSTM networks may outperform these models in handling long-term dependencies. In contrast, linear regression and ARIMA methods may not be well-suited for analyzing complex nonlinear and time-series data. Linear regression models can only capture linear relationships, while ARIMA models are limited to stationary time series and cannot handle non-stationary data.

The selection of appropriate models is crucial for achieving accurate predictions, as different models have varying strengths and weaknesses depending on the context of application. Linear regression is typically suited for modeling linear relationships between variables, but may not perform well for non-linear data. Support vector machines (SVM) possess strong generalization ability and robustness to outliers, and thus are suitable for classification and regression analyses with small sample sizes. Decision trees can effectively capture non-linear relationships and interactions between variables, but are prone to overfitting and getting trapped in local optima. LSTM neural networks are well-suited for processing time-series data, and can capture long-term dependencies and complex non-linear relationships. However, they require a large amount of training data and computational resources. There are also many other time series forecasting models to consider trying out, such as Prophet [12].

Table 2: Quantitative Evaluation Results of Each method.

	RMLSE	MSE	MAE
Linear Regression	2.7767	10452.5927	78.4731
SVM	2.1924	400.5248	16.1150
Random Forest	1.9992	213.8012	10.8773
LSTM	1.9737	170.7337	9.9695
ARIMA	2.6685	2862.8156	46.3916

5. Conclusion

This study employs a variety of ML and time series analysis methods, including linear regression, SVM, LSTM neural network, and ARIMA to forecast stock prices using historical data from the China-Techgiant-Stock-Data-in-HK-Market-2022 dataset available on Kaggle. Tencent's stock data was selected for analysis. The five methods were then utilized for training and forecasting, and a comparative analysis was conducted, with LSTM neural network producing the most accurate predictions, followed by random forest, SVM, and linear regression and ARIMA.

The findings suggest that LSTM has strong performance in processing time series data, which can assist investors in making more precise predictions and decisions, ultimately leading to improved investment returns and reduced risks. However, there is still much room for improvement in the

DOI: 10.54254/2754-1169/45/20230281

accuracy of prediction results, and even the LSTM neural network, which produced the best results, could not achieve perfect accuracy. Future research may focus on enhancing the performance of LSTM neural networks by utilizing more extensive training data, implementing more complex network structures, or integrating additional learning techniques. Furthermore, alternative time series prediction methods could also be explored.

References

[1] *Chung, H, Shin, K.: Genetic algorithm-optimized long short-term memory network for stock market prediction. Sustainability 10(10), 3765 (2018).*

[2] *Mutswenje, V.S.: A survey of the factors influencing investment decisions: the case of individual investors at the NSE (Doctoral dissertation, University of Nairobi), (2009).*

[3] *Box, G.E.P., Pierce, D.A.: Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. Journal of the American statistical Association 65(332), 1509-1526 (1970).*

[4] *Hearst, M.A., Dumais, S.T., Osuna, E., et al.: Support vector machines. IEEE Intelligent Systems and their applications 13(4), 18-28 (1998).*

[5] *Graves, A., Graves, A.: Long short-term memory. Supervised Sequence Labelling With Recurrent Neural Networks, 37-45 (2012).*

[6] *Cai, J., Luo, J., Wang, S., et al.: Feature selection in machine learning: A new perspective. Neurocomputing 300, 70-79 (2018).*

[7] *Vijh, M., Chandola, D., Tikkiwal, V.A., et al.: Stock closing price prediction using machine learning techniques. Procedia computer science 167, 599-606 (2020).*

[8] *Kaggle, 2022. China TechGiant stock data in HK market [online] Available at: https://www.kaggle.com/datasets/liqiang2022/china-techgiant-stock-data-in-hk-market-2022 [Accessed 31 March 2023].*

[9] *Su, X., Yan, X., Tsai, C.L.: Linear regression. Wiley Interdisciplinary Reviews: Computational Statistics 4(3), 275-294 (2012).*

[10] *Biau, G., Scornet, E.: A random forest guided tour. Test 25, 197-227 (2016).*

[11] *Comerton-Forde, C., Putniņš, T.J.: Measuring closing price manipulation. Journal of Financial Intermediation 20(2), 135-158 (2011).*

[12] *Facebook, 2023. Prophet [online] Available at: https://facebook.github.io/prophet/ [Accessed 31 March 2023].*