Computing and Informatics, Vol. 43, 2024, 38–63, doi: 10.31577/cai_2024_1_38

PROPOSED BAYESIAN OPTIMIZATION BASED LSTM-CNN MODEL FOR STOCK TREND PREDICTION

Bey Kun Chan, Olanrewaju Victor Johnson, XinYing Chew*

Universiti Sains Malaysia
School of Computer Sciences
11800 Gelugor, Penang, Malaysia
e-mail: {beykun18, olajohnson}@student.usm.my, xinying@usm.my

Khai Wah Khaw

Universiti Sains Malaysia
School of Management
11800 Gelugor, Penang, Malaysia
e-mail: khaiwah@usm.my

Ming Ha Lee

Swinburne University of Technology Sarawak Campus
Faculty of Engineering, Computing and Science
93350 Kuching, Sarawak, Malaysia
e-mail: mhlee@swinburne.edu.my

Alhamzah Alnoor

Southern Technical University
Management Technical College
Basrah, Iraq
e-mail: Alhamzah.malik@stu.edu.iq

* Corresponding author

Abstract. Stock prediction is prominent in the field of Artificial Intelligence. Stock prediction problems are handled either as a regression or classification task. Studies in the literature have also shown success for hybrid learning to stock prediction. But little attention is paid to finding out the effect of spatial feature extraction/distortion over the temporal effect of the deep neural network and vice versa for the problem under study. The paper, therefore, proposes a hybrid long short-term memory (LSTM) network over a convolutional neural network (CNN) called LSTM-CNN as against the popular CNN-LSTM model. The daily price movement of the S & P 500 index data is utilized. A sliding window technique is considered to obtain a balanced data of 20-days window data from the S & P 500. The proposed stock prediction model is investigated further for an optimal set of hyperparameters using the Bayesian optimization (Bo) technique. In addition, the proposed model is compared with optimized CNN, LSTM, and CNN-LTSM models. The optimized LSTM-CNN model is found to outperform the other models with accuracy, precision, and recall values of 0.9741, 0.9684, and 0.9800, respectively. The proposed model is established to provide a better stock trend prediction.

Keywords: Stock management, hybrid learning, deep learning, optimization, prediction

1 INTRODUCTION

Investing in a certain financial product is crucial to protect one's wealth from inflation. The stock market comes to be a very popular option among many investment portfolios. Though, investment comes with its own risk and reward, and the stock market is not exemptional. The stock market is projected as a very volatile type of investment due to its highly non-linear nature [1]. The prices of the assets are affected by investors' emotions, earning reports, and news events including pandemics, war, oil, and gold prices [2, 3, 4, 5]. In addition, Abu-Mostafa et al. [6] and Deboeck [7] have pointed out that the financial trends as time series are noisy and non-stationary. These behaviors of the stock market make it a difficult task to model the price trend. Despite the complexities and the fluctuations evident with the stock market and financial assets, traders, investors, and fund managers could predict the direction of the market using technical analysis. A survey in [8] shows, over 10 000 institutional portfolios in which about one-third use technical analysis and the portfolios that were managed, technical analysis outperformed those which do not utilize technical analysis. The Singapore Stock Exchange traders were also able to make substantial profits by applying technical indicators when making trading decisions [9]. One practical aspect of achieving many milestone performances is stock trend prediction (STP). STP has been widely studied due to its beneficial impact on stock market analysis and still draws the attention of many researchers [10]. Four major approaches to STP are vastly investigated in the literature including statistics [11], pattern recognition [12], machine learning (ML) [13], and sentiment analysis [14]. Moreover, deep learning (DL) techniques, an aspect of ML, are gaining apt attention in recent studies of STP due to their ability to handle complex non-linear computational intelligence [15]. Diverse DL techniques are since suggested in the literature for handling STP.

Earlier, Gençay [16] suggested that technical traders make their decisions based on the idea that market patterns are assumed to reoccur in the future and thus the past patterns can be used to predict the future trend if they occur again. He has also found strong evidence suggesting that the non-linear stock market can be predictable by using historical buy and sell signals. This is very motivating as it suggests that there are patterns in the stock prices that can be used to predict the direction of the stock. In recent times, a convolutional neural network (CNN) [17], a state-of-the-art algorithm that does well in recognizing patterns in image data is employed for STP. In addition, the long short-term memory (LSTM) model is found also suitably dominant and preferable DL model when it comes to financial time-series prediction because it fits well with the time-varying characteristic of financial data [18]. On the other hand, CNN works better on static data such as image data which is non-time varying. Since there are spatial features in the stock price chart that are often used by technical traders, researchers have also found creative ways to transform 1-D financial time-series data into 2-D image-like data. For example, the authors in [19] have transformed the 1-D financial time series to image data by using Gramian Angular Field (GAF). The GAF imaging method is proposed by [20] in which the authors also proposed Markov Transition Field (MTF) method to encode the time-series data to image data. Gudelek et al. [21] use the sliding window technique to encode the time-series data into 28-by-28 image data. The 28-by-28 image data consists of the price, volume, and values of 26 technical indicators across 28 days. The diverse abilities of DL techniques in dealing with either the spatial or temporal features in data led to the formation and acceptance of hybrid DL models. Hence, the popularity of the CNN-LSTM model in the studies of STP.

Meanwhile, the use of CNN as the top layer tends to disrupt the temporal features of the time series data. It may, therefore, reduce the effectiveness of the LSTM layer. On the other hand, LSTM outperforms CNN in stock prediction. It is not surprising as we are dealing with time series data and LSTM can capture temporal features better than CNN [22, 23]. Researchers often use CNN to perform feature extraction and feed the output to the LSTM layer [22, 23, 24]. But the drawback of using CNN as the top layer on LSTM remains unresolved. More so, Jiang [10] suggested the need to provide a more comparative structure of neural networks to fully understand stock predictions. The question is: "are temporal features more important than spatial features in stock trend prediction?" Therefore, the paper proposes an LSTM-CNN approach to the study of stock prediction on one part as a way of improving on the drawback of CNN-LSTM and the other compares the different architecture of hybrid models of CNN and LSTM.

The benefit of daily stock trend prediction lies in providing investors and traders with valuable insights into the potential direction of stock prices over the short term, which includes:

Short-term trading opportunities: Daily stock trend predictions can help short-term traders identify potential entry and exit points, facilitating more agile trading strategies to capitalize on short-term price movements [13].

Risk management: By having an idea of the daily trend, investors can adjust their position sizes or implement stop-loss orders to manage risk more effectively [16].

Enhanced decision-making: Daily trend predictions, when combined with other forms of analysis (e.g., fundamental analysis), can assist investors in making more informed decisions about their portfolios [21].

Trading volume: Daily predictions may lead to increased trading activity, which can improve liquidity and market efficiency [23].

Algorithmic trading: Many algorithmic trading strategies [25, 26, 27] depend on short-term price trends, and daily predictions can be valuable inputs for these strategies.

Sector and market analysis: Daily trend predictions can help identify broader market trends or sector-specific movements, allowing investors to adjust their allocations accordingly [28].

Behavioral finance: Studying daily trends can shed light on investor sentiment and market psychology, which can be informative for understanding market dynamics [29].

Learning and research: Daily predictions can serve as training data for developing and refining prediction models, advancing the field of machine learning and artificial intelligence in finance [30].

The rest of the paper is organized as follows. In Section 2, we discuss the related works. Section 3 provides the methodology used in the paper. Section 4 presents the results and discussion. Finally, the conclusion as well as suggestions for future research are discussed in Section 5.

2 RELATED WORKS

There are numerous different methods of stock forecasting. These methods include vast statistical and ML techniques. In general, statistical models for stock forecasting are statistical assumptions about stock price movement. The assumption allows us to calculate the probability of the stock trend and make predictions thereof. The training and predicting processes of statistical models are very well defined by statistics and calculations. This means that all the predicted results can be traced and understood easily making the model more transparent. The Autoregressive Integrated Moving Average (ARIMA) model, also known as the Box-Jenkins model, is commonly used in statistical methods. It is a model that is fitted to time-series data to perform analysis and forecasting. ARIMA relies only on past data of the time series and the error from the previous forecasting. The authors in [31] compared the ARIMA model with Artificial Neural Network (ANN) using only stock closing price from the New York Stock Exchange as input and found that ANN performed superior to the ARIMA model. One important reason why ARIMA performed worse than ANN is that the ARIMA model assumes that the data are stationary [32]. This makes it difficult to model financial time-series which are highly non-linear. The authors in [11, 31] found that the ARIMA model is more robust in short-term prediction. However, it cannot achieve convergence in the long term [33]. This means that the prediction accuracy of ARIMA will get worse in the long run.

In short, although the ARIMA model performs quite well in the short term, it has some fatal limitations. Firstly, as a linear statistical model, it assumes that the data are linearly correlated [34] and stationary [6, 7]. This assumption makes it fail to capture the non-linear pattern from financial time-series data. Secondly, it does not converge to the financial time-series in the long-term causing more computational cost [33]. Lastly, it only accepts univariate time-series data, thus means it is challenging for ARIMA to imitate technical traders by using technical indicators and stock prices. Gong and Sun [28] used a Logistic Regression (LR) model to predict the next month's stock price trend of Shenzhen Development Stock A using the current month's data. The inputs used were closing price, opening price, highest price, lowest price, composite weight price, daily turnover, the total amount traded, and total volume traded. LR model was found to have much lower complexity and higher accuracy in the study when compared to hybrid models such as Radial Basis Function-Artificial Neural Network (RBF-ANN). However, the LR model only performed better with less training data. When there were more training data, the RBF-ANN performed better. Although LR is popular in ML recently, however, there exist other sophisticated models for prediction problems.

The authors in [35] used a three-layer back-propagation ANN as a benchmark to compare with 2 support vector machine (SVM) models. The SVM models utilized the Gaussian kernel and polynomial kernel to map input data to a higher-dimensional space. Five different datasets including Standard & Poor 500 stock index futures (CME-SP), United States 30-year government bond (CBOT-US), United States 10-year government bond (CBOT-BO), German 10-year government bond (EUREX-BUND) and French government stock index futures (MATIF-CAC40) were used. The experiment conducted as a regression problem to forecast future prices of the financial products uses training data consisting of 5 input variables that map to 1 output variable. The input variables include 1 Exponential Moving Average (EMA) and 4 lagged Relative Differences in Percentage of Price (RDP) values while the mapped output is an RDP of 5 days in advance. Meanwhile, the price was first smoothed by a 3-day EMA. The models were evaluated using normalized mean squared error (NMSE), mean absolute error (MAE), directional symmetry (DS), and weighted directional symmetry (WDS). The authors noted that the Gaussian-SVM performed better than the Polynomial-SVM but the experimental results for Polynomial-SVM were not given. The experimental results also showed that the Gaussian-SVM performed better than the three-layer back-propagation network. Across all the datasets, the Gaussian-SVM has lower NMSE, MAE, and WDS than the three-layer back-propagation network, although the WDS of the former is greater than the latter. It indicates that Gaussian-SVM performed better than the three-layer back-propagation network.

The research conducted in [29] compared a three-layer ANN, Polynomial-SVM, and Radial Basis Function-SVM (RBF-SVM) by predicting the stock direction of the daily Istanbul Stock Exchange (ISE) National 100 Index. The dataset consists of 10 technical indicators as inputs mapped to a binary label as output for each entry. The value of the output was determined by comparing the current price and the previous price. For example, if the current price is higher than the previous price, the output is “1” which means an uptrend and vice versa. The experimental results showed that the ANN achieved an average accuracy of 75.74 % while Polynomial-SVM and RBF-SVM achieved 71.52 % and 62.23 %, respectively. A three-layer multilayer perceptron (MLP) was compared with an Elman Recurrent Neural Network and a Linear Regression model to predict the Tehran Stock Market stock price [36]. The inputs used were the highest daily price, lowest daily price, and the average daily price of the last 1 to 10 days. The output is the predicted price of the next day. For example, if only the inputs consist of only 1-day prices, the second-day price will be predicted. If the inputs consist of 10-day prices, the eleventh-day price will be predicted. In addition to price prediction, price change prediction was also performed which relates more to our research objective. The MLP performed the worst with an accuracy of slightly over 50 %. This is not much better than flipping a coin to guess the trend of the stock. Another research [15] utilized a three-layer MLP to predict buy, sell, or hold signals for the Walmart (WMT) stock. Three technical indicators were used as inputs to the MLP model for prediction. An overall accuracy of 65.52 % was achieved. Results from the different studies show that MLP lacks capacity to capture either the spatial or temporal features of stock, hence may not be a choice model for stock prediction.

CNN was employed to perform regression and classification tasks on stock forecasting [12]. The data used consists of 17 Exchange Traded Funds (ETFs) listed on the New York Stock Exchange. The data includes the *computed price* calculated, the volume traded, and 25 values of technical indicators resulting in a total of 28 features. It is worth noting that the 25 values of technical indicators were from 8 technical indicators with different time lags. The 28 features were collected across 28 trading days. It was achieved by utilizing the sliding window technique. It results in 28 parallel time-series that were stacked upon each to form a 2-D-like image data. The 29^{th}-day price and trend were then used as target labels for the stock trend prediction. The CNN model trained for regression task (price prediction) can be used for trend prediction as well. This was done by discretizing the predicted price. As a result, the CNN model trained for the classification task (trend prediction) achieved an accuracy of 78.46 % for binary classification. Using 9 variables including month, date of the month, day of the week, change of daily close price, change of daily open price, change of daily highest price, change of daily lowest price, change of daily volume traded, and change of daily high-low price difference percentages, a CNN prediction model was built on NIFTY50 [37]. Meanwhile the CNN model lacks temporal feature extraction of stock data, which is an important aspect of stock prediction. In comparison with CNN, other models including Multivariate Regression, Decision Tree Regression, Bagging Regression, Boosting Regression, Random Forest (RF), ANN, and SVM models were tested. The models were evaluated using the Root Mean Square Error (RMSE) and the product-moment correlation value between the predicted value and the actual value of NIFTY50. The results showed that the CNN model performed the best with the highest correlation value and lowest RMSE when compared to other models.

A study by Nelson et al. [38] compared LSTM, MLP, RF, and Pseudo-Random models in predicting the stock trend. The data used were labeled with 2 classes namely "up" or "down". If the stock price of the current 15-minute timeframe is higher than the previous 15-minute timeframe, then it is labeled as "up" and vice versa. The models were trained and tested on 5 stocks from the Brazilian Stock Exchange (Bovespa) namely BOVA11, BBDC4, CIEL3, ITUB4, and PETR4. The results indicate that the LSTM has the highest accuracy across all 5 stocks in predicting the stock trend. An average accuracy of 55.9 % was achieved by LSTM. While the LSTM performed the best among all the models, the accuracy cannot be considered as good. We think that the LSTM achieved low accuracy because of the large amount of input. The researchers used 175 technical indicators which are too overwhelming leading to the curse of dimensionality. Whereas training the LSTM model may improve the model's performance. See [22] for a further comparative study of DL models for stock forecasting.

In a similar vein, a hybrid model is believed to improve model accuracy. The authors in [21] proposed a hybrid model of CNN-LSTM to predict the stock market index of Shanghai (STI), Japan (N225), Singapore (STI), and Indonesia (JSX). The study compared the performances of CNN, LSTM, and CNN-LSTM. The performances of the models were evaluated using RMSE. The stock price time series of STI, N225, STI, and JSX was collected as 4 parallel time series in a time step of 4 days. The data were then used to train the models. The results indicated that the CNN-LSTM model performed the best with the lowest RMSE when predicting the stock index prices. However, the CNN-LSTM model maybe lacking in capturing temporal dependencies in the data while also handling multi-dimensional features consist in the stock data.

3 METHODOLOGY

This section discusses the materials and methods used in the research paper. The data is based on the past 20 days' S & P 500 chart data to predict the next-day end of a financial time-series classification task. The output will be either "uptrend" or "downtrend", encoded as 1 and 0, respectively.

3.1 Data Used

The financial asset used is the Standard & Poor's 500 (S & P 500) index with the ticker symbol 'GSPC' sourced from Yahoo Finance [30]. The S & P 500 index tracks the performance of the 500 leading publicly traded companies on stock exchanges in the United States of America. The data include the closing price, opening price, highest price, lowest price, volume traded of the S & P 500 index. The data is collected in a daily timeframe from 1st November 1999 to 31st December 2019 (20 years and 2 months). It is worthwhile to point out that the volume of data from the final 2 months in 1999 is required to compute the technical indicators, whereas the actual data volume used for model training and testing is from 1st January 2000 to 31st December 2019 (20 years). Values of technical indicators derived from the S & P 500 prices are thereafter computed using the Technical Analysis Library (TA-Lib) [39]. The technical indicators computed are listed in Table 1 Note that the number of outputs for each technical indicator may be different. Thus, we treat each output as an individual feature even though they might be from the same technical indicator. The formulas for each technical indicator are stated below.

Technical Indicators	Parameters	No. of Outputs
Simple Moving Average	MA Length: 5	1
Simple Moving Average	MA Length: 10	1
Exponential Moving Average	MA Length: 5	1
Exponential Moving Average	MA Length: 10	1
Weighted Moving Average	MA Length: 5	1
Weighted Moving Average	MA Length: 10	1
Relative Strength Index	RSI Length: 14	1
Stochastic	%K Length: 14	
	%D Smoothing: 3	2
Commodity Channel Index	Length: 20	1
Moving Average Convergence Divergence	Fast Length: 12	
	Slow Length: 26	
	Signal Smoothing: 9	3
Total		13

Table 1. List of technical indicators collected

Simple Moving Average:

$$SMA_{t,n} = \frac{p_1 + p_2 + p_3 + \cdots + p_t}{n}, \tag{1}$$

where $SMA_{t,n}$ is the SMA at period t, p_t is the asset price at period t, and n is the number of days to average.

Exponential Moving Average:

$$EMA_{t,n} = p_t * \frac{2}{(n+1)} * EMA_{t-1,n} * \left(1 - \frac{2}{(n+1)}\right), \tag{2}$$

where $EMA_{t,n}$ is the EMA at period t, p_t is the asset price at period t, and n is the number of days to average.

Weighted Moving Average:

$$WMA_{t,n} = \frac{p_1(n) + p_2(n-1) + p_3(n-2) + \cdots + p_t}{\frac{n*(n+1)}{2}}, \tag{3}$$

where $WMA_{t,n}$ is the WMA at period t, p_t is the asset price at period t, and n is the number of days to average.

Relative Strength Index:

$$RSI_{t,n} = 100 - \frac{100}{1 + \frac{AvgUp_n}{AvgDn_n}}, \tag{4}$$

where $RSI_{t,n}$ is the RSI at period t, $AvgUp_n$ is the average gain of price in the last n days, $AvgDn_n$ is the average decrease of price in the last n days, and n is the time.

Stochastic:

$$\begin{aligned} \%K_{t,n} &= \frac{p_t - Lowest_n}{Highest_n - Lowest_n}, \\ \%D_{t,d} &= SMA_{t,d}(\%K), \end{aligned} \tag{5}$$

where $\%K_{t,n}$ is the Fast K value at period t, p_t is the asset price at period t, $Lowest_n$ is the lowest price in the past n days, $Highest_n$ is the highest price in the past n days, n is the time, $\%D_{t,d}$ is the simple moving average of Fast K at period t across d days.

Commodity Channel Index:

$$CCI_{t,n} = \frac{TP_t - SMA_{t,n}(TP)}{0.015 * MD}, \tag{6}$$

where TP_t is the typical price ($TP = \frac{High+Low+Close}{3}$) at period t, $SMA_{t,n}(TP)$ is the simple moving average of typical price over n days at period t, and MD is the mean deviation ($MD = \frac{\sum_{i=1}^{n} TP_i - SMA_{t,n}(TP)}{n}$).

Moving Average Convergence Divergence:

$$\begin{aligned} MACD_t &= EMA_{t,12}(p) + EMA_{t,26}(p), \qquad (7) \\ MACD_{signal_t} &= SMA_{t,9}(MACD), \\ MACD_{hist_t} &= MACD - MACD_{signal}, \end{aligned}$$

where p is the current asset price, $EMA_{t,12}(p)$ is the exponential moving average of asset price at period t over 12 days, and $SMA_{t,9}(MACD)$ is the simple moving average of $MACD$ at period t over 9 days.

3.2 Model Architecture

The proposed Bo-based LSTM-CNN consists of an LSTM model and the CNN model. The LSTM receives the data input and constructs temporal features from the data as input into the CNN model. The CNN then works on the temporal features input to make a classification of the stock trend – "0" or "1". The proposed model is shown in Figure 1. Meanwhile, the 'm layers' and 'n layers' in Figure 1 mean that there might be repeating layers of LSTM, CNN, and Pooling layer for the models. These numbers require optimization. Models are known to behave differently with the same or different sets of hyperparameters. The Bo is used to optimize the LSTM model. Bo is more efficient than Grid Search and Random Search (as other most popular optimization algorithms) because the hyperparameters are selected in an informed manner [40].

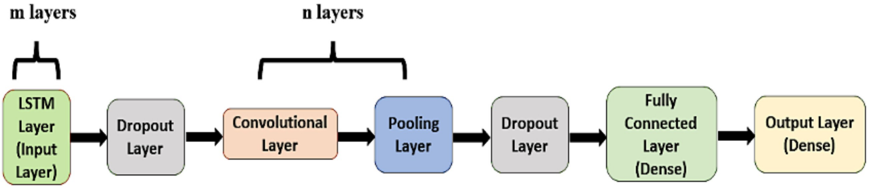


Figure 1. The architecture of the proposed hybrid LSTM-CNN

4 RESULTS AND DISCUSSION

After selecting the final input features, we performed data normalization using the min-max technique $[-1, 1]$ and data segmentation using the sliding window technique, following the suggestion in [41], which studied stock prediction using data segmented with a window size of 20, 40, 60, and 80. The authors found that the window size of 20 performed the best. Thus, our window size is 20. Spearman's rank correlation coefficient in a 20-days window was computed and the average is found. The extracted 20-day data using the sliding window technique is shown in Figure 2. In each figure, Figure 2 a) depicts the extracted trend, while Figure 2 b) demonstrates the 20-day sliding-window technique. The reason coefficients in a 20-days window are computed is the different ranges in the whole dataset. For example, the price ranges from 0 to infinity while the Relative Strength Index (RSI) has a fixed range (0 to 100). Calculating the correlation coefficient for the whole data at once, it is expected that the technical indicators with fixed ranges will have low correlations to the close price. The effect is as the price continues to increase; the technical indicators fluctuate between 2 fixed values. Thus, it is practically infeasible to extract useful information on the whole data. The results of Spearman's rank correlation coefficient on the whole data and the 20-day window are shown in Figure 3. Figure 3 b) shows a better correlation coefficient result than the Figure 3 a). Hence, 2 features from the moving averages and 3 features from the other technical indicators are most correlated to the close price of the S & P 500. The features are EMA_5, WMA_5, RSI, STOCH_K, and CCI. 10 attributes in total are, therefore, selected including the close price for training the model.

The min-max helps to scale down features with higher magnitudes, thereby nullifying their weighing effect on other features with smaller magnitudes. The target variable Y (close price) is thereafter labeled "1" as the uptrend and "0" as the downtrend. We determine the trend by comparing the closing price of the final day in the predictor with the next day's closing price. If the closing price on the final day is lower than the next day, then the predictor's label will be "1" (uptrend), else the label is "0" (downtrend). In short, we will be using the 20 days' worth of input features (1 predictor) to predict the trend of the S & P 500 on the 21st day (next the day). The one day ahead prediction is beneficial to STP as ealier discussed in latter part of the introduction of this paper and popularly suggested in the literature [25, 26, 27]. Refer to the following rule for a formal description of the predictors.

$$Y = \begin{cases} 1, & \text{if } close_i < close_{i+1}, \\ 0, & \text{if } close_i \geq close_{i+1}. \end{cases}$$

Note that, $close_i$ means the closing price of the S & P 500 on the i^{th} day.

Finally, the dataset is split into 80 % training, 10 % validation, and 10 % testing sets, randomly, respectively.

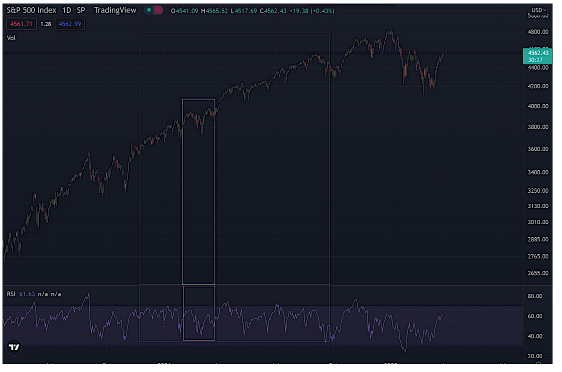
Other models such as CNN, LSTM, and CNN-LSTM were introduced to compare with the LSTM-CNN for performance evaluation. There is no practical guide, to the best of our knowledge, on how to select the hyperparameter range when performing optimization. As suggested in [42] the learning rate should be within the range of 0.000002 and 1. The models are, therefore, subjected to Bo hyperparameter search to obtain parameters at which the model is performing at its best.

Initially, 50 trials for each model are performed. This means that 50 different sets of hyperparameter values are tested for each model. For each trial, the models are trained for 5 epochs. Ideally, the models should be trained for more epochs. Besides, the models' performances should be shown to sync with the optimization process. For example, for each combination of hyperparameter values, the model is trained for 100 epochs with the Early Stopping criterion and the best model trained with the optimal hyperparameters will not require final training. Unfortunately, we have limited resources and can only train each combination for a low number of epochs. Therefore, final training is needed to train the models using the optimal hyperparameters found. The benefit of using a low number of epochs is that the combination of hyperparameter values results in a model that is fast learning during the start of the training. The downside is that fast initial learning does not guarantee the best performance at the end. The optimal set of hyperparameter values will be chosen based on the validation accuracy during the training. In other words, we are optimizing the models based on their validation accuracy. The hyperparameters search before and after the training are presented in Tables 2, 3, 4, 5.

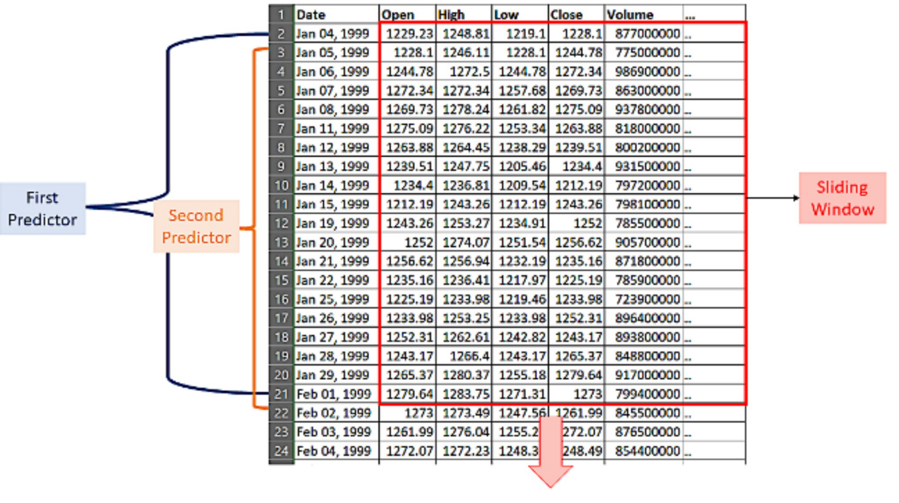
We consider the distribution of the data and observe that the percentages of both classes are close, as shown in Figure 4. It means that the dataset is balanced. The four models are then constructed using the training, validation, and test sets.

The prediction output by the trained models is a probability of the form given by.

$$Trend = \begin{cases} 1, & \text{if } prediction\ probability > 0.5, \\ 0, & \text{if } prediction\ probability \leq 0.5. \end{cases}$$



a) Extracted stock trend

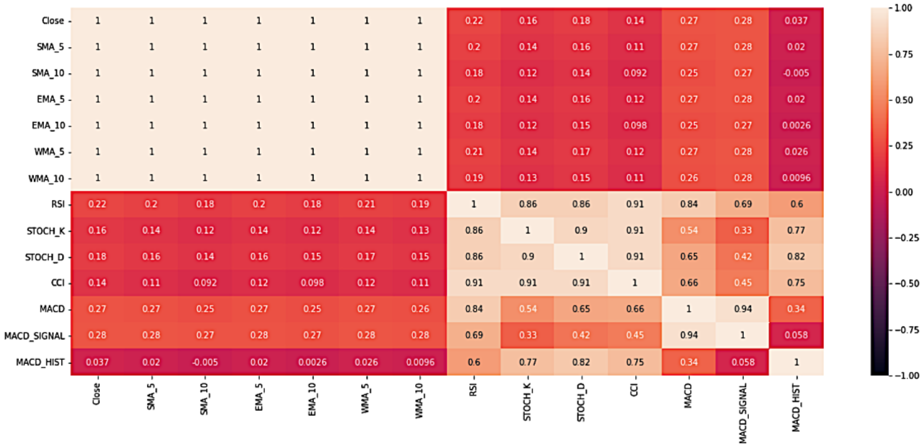


1	Date	Open	High	Low	Close	Volume	...
2	Jan 04, 1999	1229.23	1248.81	1219.1	1228.1	877000000	..
3	Jan 05, 1999	1228.1	1246.11	1228.1	1244.78	775000000	..
4	Jan 06, 1999	1244.78	1272.5	1244.78	1272.34	986900000	..
5	Jan 07, 1999	1272.34	1272.34	1257.68	1269.73	863000000	..
6	Jan 08, 1999	1269.73	1278.24	1261.82	1275.09	937800000	..
7	Jan 11, 1999	1275.09	1276.22	1253.34	1263.88	818000000	..
8	Jan 12, 1999	1263.88	1264.45	1238.29	1239.51	800200000	..
9	Jan 13, 1999	1239.51	1247.75	1205.46	1234.4	931500000	..
10	Jan 14, 1999	1234.4	1236.81	1209.54	1212.19	797200000	..
11	Jan 15, 1999	1212.19	1243.26	1212.19	1243.26	798100000	..
12	Jan 19, 1999	1243.26	1253.27	1234.91	1252	785500000	..
13	Jan 20, 1999	1252	1274.07	1251.54	1256.62	905700000	..
14	Jan 21, 1999	1256.62	1256.94	1232.19	1235.16	871800000	..
15	Jan 22, 1999	1235.16	1236.41	1217.97	1225.19	785900000	..
16	Jan 25, 1999	1225.19	1233.98	1219.46	1233.98	723900000	..
17	Jan 26, 1999	1233.98	1253.25	1233.98	1252.31	896400000	..
18	Jan 27, 1999	1252.31	1262.61	1242.82	1243.17	893800000	..
19	Jan 28, 1999	1243.17	1266.4	1243.17	1265.37	848800000	..
20	Jan 29, 1999	1265.37	1280.37	1255.18	1279.64	917000000	..
21	Feb 01, 1999	1279.64	1283.75	1271.31	1273	799400000	..
22	Feb 02, 1999	1273	1273.49	1247.56	1261.99	845500000	..
23	Feb 03, 1999	1261.99	1276.04	1255.2	272.07	876500000	..
24	Feb 04, 1999	1272.07	1272.23	1248.3	248.49	854400000	..

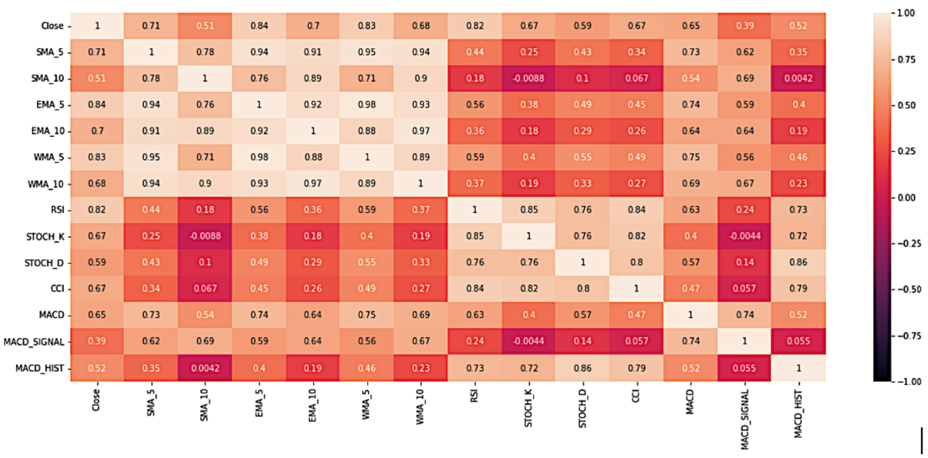
b) Extracted data using the sliding window technique

Figure 2. The 20-day sliding window technique

Accuracy and loss functions are plotted for each model. The results are as presented in Figures 5, 6, 7, 8. Figure 5 illustrates the model accuracy and loss for the CNN model. It could be observed that the CNN model is well-trained. It is neither overfitted nor under-fitted. The training and validation accuracies are close to each other, and the improvement only stagnates after around 70 epochs



a) All the observations days



b) 20-day window

Figure 3. Heatmap correlation coefficient

of training. The same can be said for the training and validation loss. The loss kept decreasing until around 80 epochs of training and there is not much deviation between training and validation loss. Besides, the model achieves high accuracy and low loss. Likewise, the LSTM and CNN-LSTM behave in the same manner as the CNN, as shown in Figures 6 a) and 6 b), and in Figures 7 a) and 7 b). However, one interesting outcome is that the LSTM model obtains better training results than the CNN model in shorter training time with training and validation accuracies climbing above 90 % during the start of the training, while the CNN-LSTM gains better performance than the two models at the same start of the training.

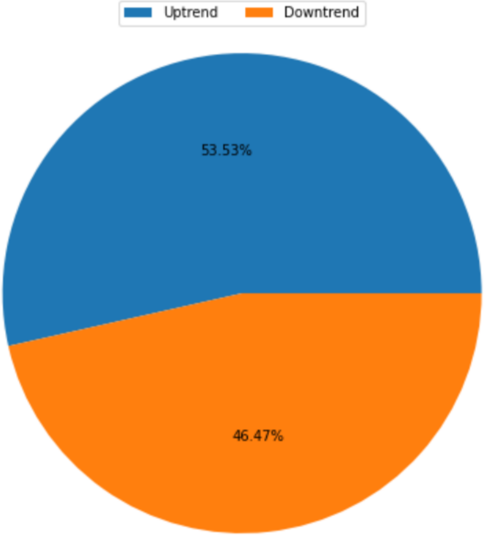


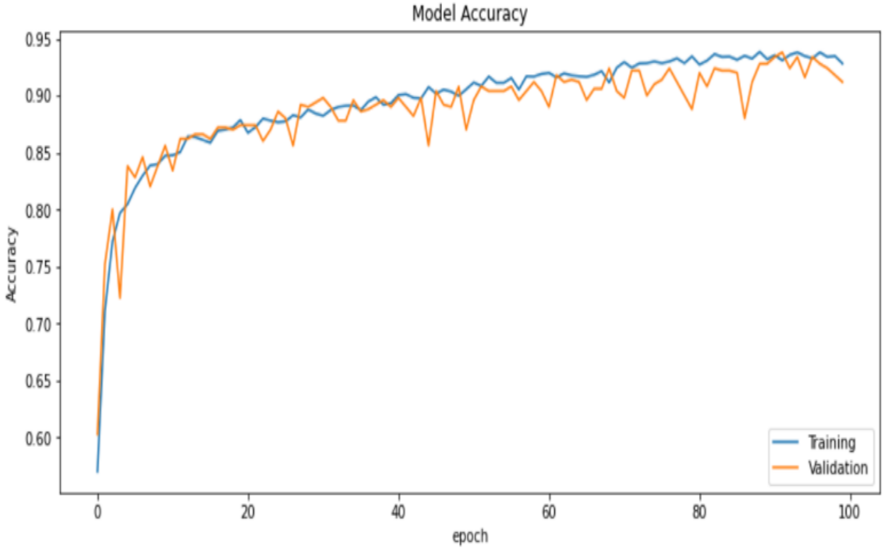
Figure 4. Classes distribution of the dataset used

Hyperparameter	Value
Dropout	0.2
Loss	Binary Cross-entropy
Optimizer	Adam
Batch Size	32
Strides (Pooling Layer)	1
Activation (Convolutional Layer)	ReLU
Activation (Fully Connected Layer)	ReLU
Activation (Output Layer)	Sigmoid

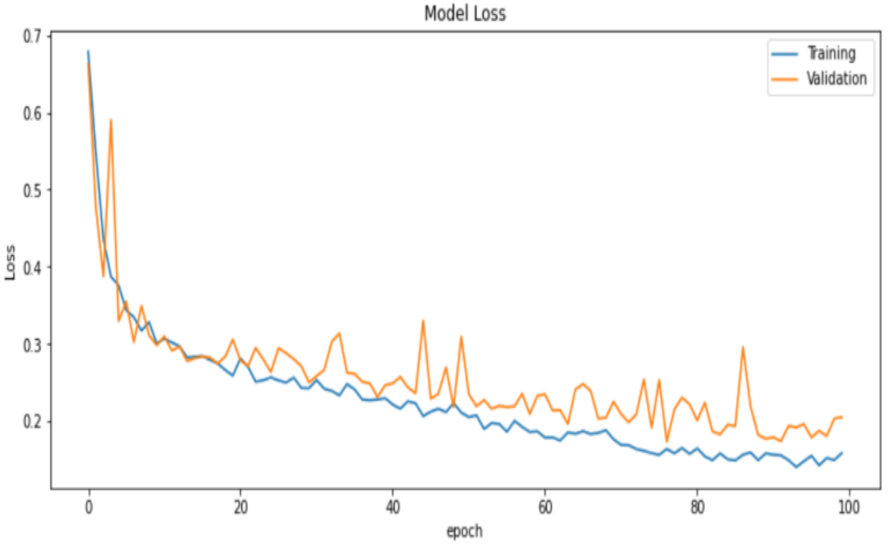
Table 2. Fixed hyperparameters

Moreover, the LSTM-CNN model achieved the highest accuracy and low loss. The model is well-trained. There is not much deviation between the training and validation accuracies. While there is a slight fluctuation in the loss during the end of the training, the deviation between the training and validation loss is not huge, as shown in Figures 8 a) and 8 b).

Considering model performance evaluation; accuracy, precision, and recall of each of the models are computed. The result shows that Bayesian-optimized LSTM-CNN obtained the best results for all 3 performance metrics used, as shown in Figure 9. The second-best model is the CNN-LSTM model, followed by LSTM, and lastly CNN (also Bayesian-optimized). Based on the metrics, the hybrid DL models performed better than the conventional DL models. It implies that there is a competitive advantage in using the hybrid model in stock trend prediction.

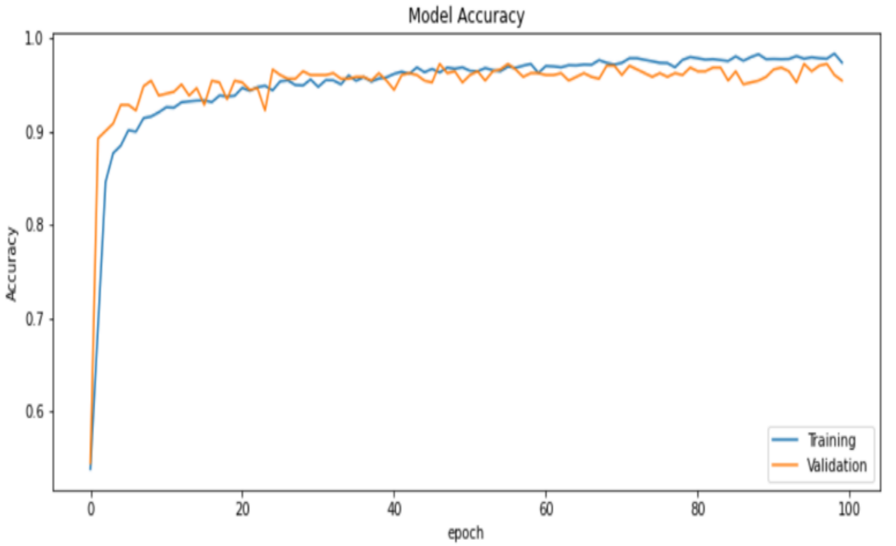


a) Model's training and validation accuracy

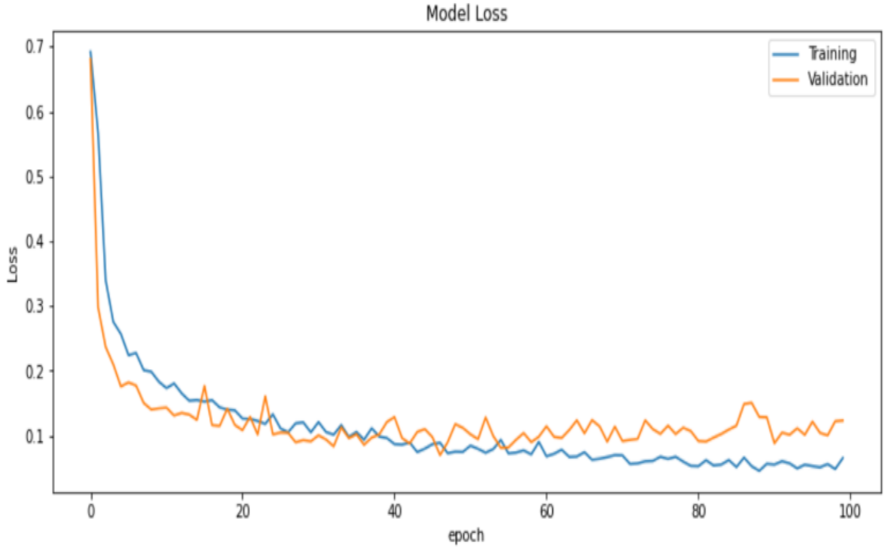


b) Model's training and validation loss

Figure 5. The CNN model

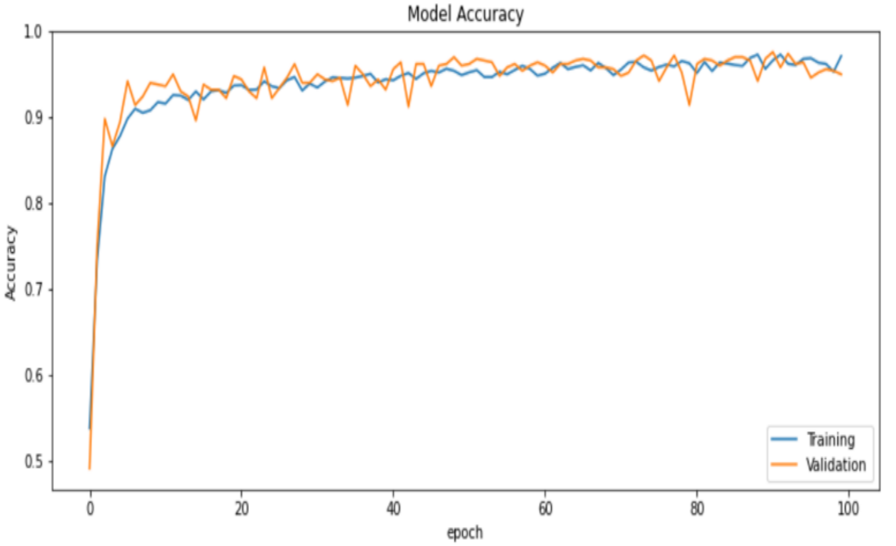


a) Model's training and validation accuracy

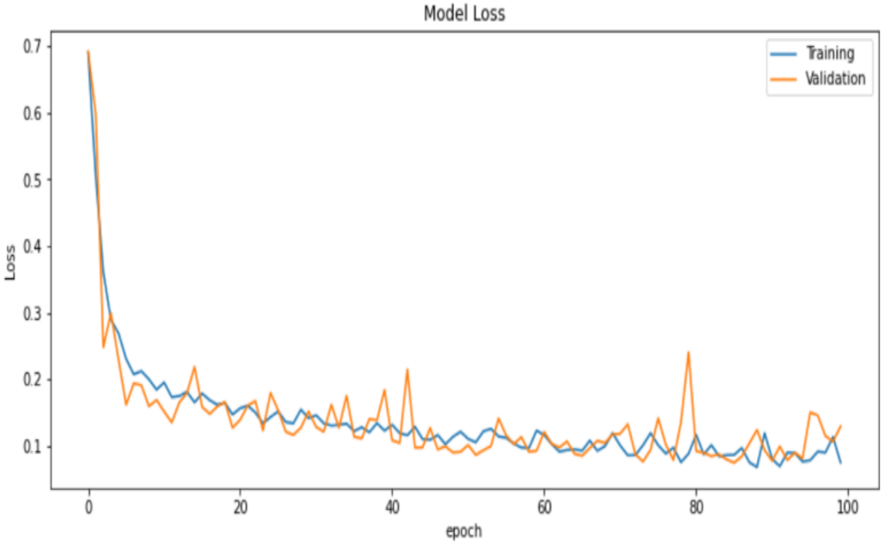


b) Model' training and validation loss

Figure 6. The LSTM model

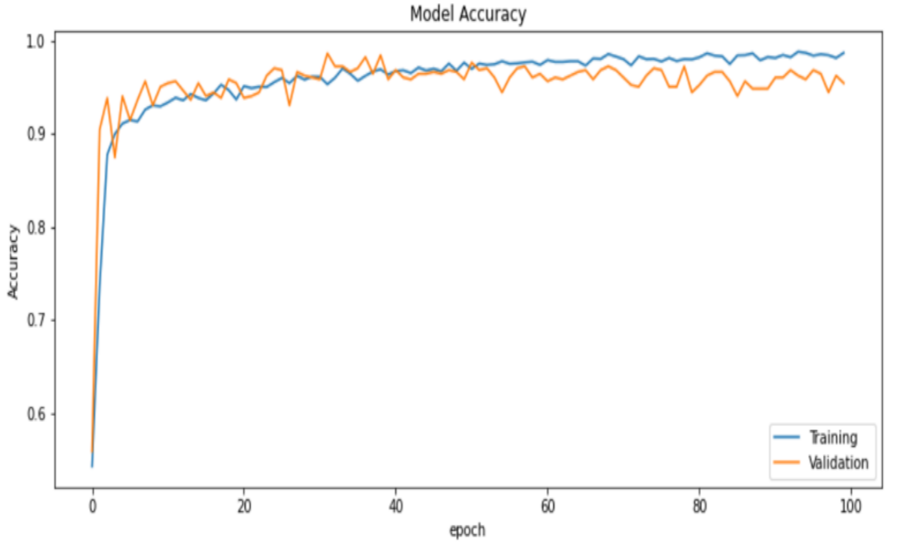


a) Model's training and validation accuracy

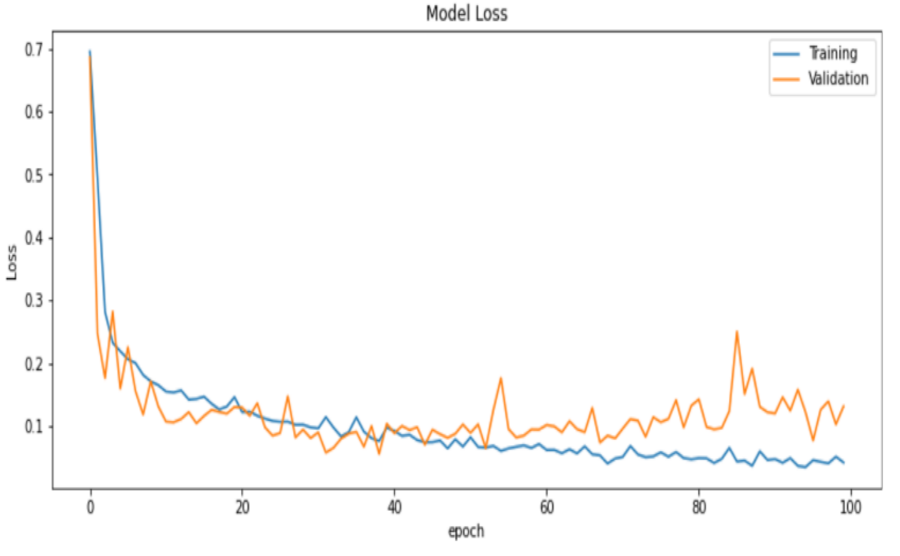


b) Model's training and validation loss

Figure 7. The CNN-LSTM model



a) Model's training and validation accuracy



b) Model's training and validation loss

Figure 8. The LSTM-CNN model

Hyperparameter	Range of Value	Optimal Value
Number of Convolutional and Pooling Layers	1–3	3
Number of Convolutional Kernels	4–64	
1st layer		4
2nd layer		64
3rd layer		4
Kernel Size	1–10	
1st layer		4
2nd layer		10
3rd layer		6
Pool Size	2–20	
1st layer		18
2nd layer		2
3rd layer		16
Number of Nodes in Fully Connected Layer	4–64	8
Learning Rate for Adam Optimizer	0.000002–0.9	0.001

Table 3. Hyperparameters search space for CNN using Bo

Hyperparameter	Range of Value	Optimal Value
Number of LSTM Layers	1–3	1
Number of LSTM Units	4–64	8
Number of Nodes in Fully Connected Layer	4–64	48
Learning Rate for Adam Optimizer	0.000002–0.9	0.005

Table 4. Hyperparameters search space for LSTM using Bo

Besides, LSTM also outperformed CNN which tells us that the temporal features are more important than spatial features in stock trend prediction. It is also true when comparing CNN-LSTM and LSTM-CNN. LSTM-CNN performed better than CNN-LSTM which confirms that there is a disruption to the temporal features if CNN is used as the top layer in the hybrid model. It is also worth noting that all 4 models achieved an accuracy of more than 90 %. The CNN and LSTM models

Hyperparameter	Range of Value	Optimal Value	
		CNN-LSTM	LSTM-CNN
Number of Convolutional and Pooling Layers	1–3	1	1
Number of LSTM Layers	1–3	1	1
Number of Convolutional Kernels	4–64	24	64
Kernel Size	1–10	6	60
Pool Size	2–20	9	3
Number of LSTM Units	4–64	12	2
Number of Nodes in Fully Connected Layer	4–64	12	64
Learning Rate for Adam Optimizer	0.000002–0.9	0.005	0.005

Table 5. Hyperparameters search space for CNN-LSTM and LSTM-CNN using Bo

in [38, 43] only achieved an average accuracy of 60.02 % and 54.26 %, respectively. In another study [44], the highest accuracies achieved by CNN and CNN-LSTM are 75.97 % and 79.94 %, respectively. The proposed hybrid model has, to compete with the best performances compared to others models in the literature.

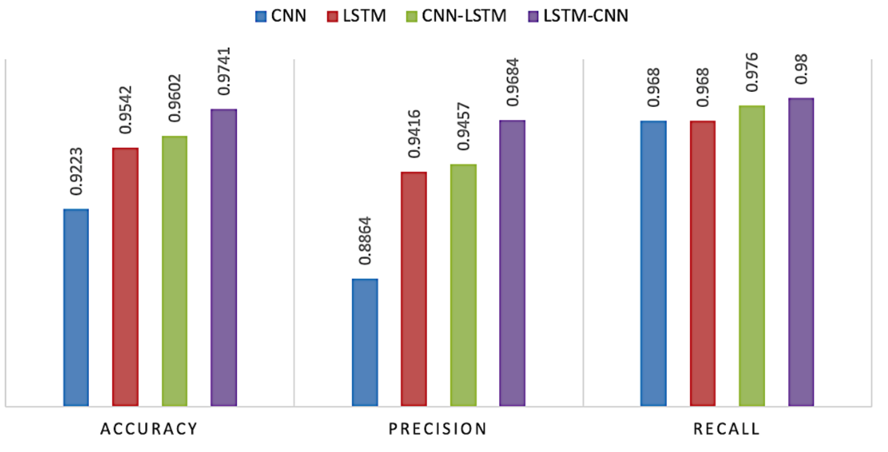


Figure 9. Comparison of models' performances

5 CONCLUSION

The proposed hybrid LSTM-CNN has been successfully investigated for stock trend prediction. Hyperparameter search to improve the model performance using the Bo technique has also been studied. In addition, the model is compared with other variants of CNN, LSTM, and CNN-LSTM based on the same optimization technique. The Bo-based LSTM-CNN achieved better results than the compared models for stock trend prediction. Furthermore, findings suggest that the conventional LSTM model can achieve good performance (accuracy $> 90\,\%$) in a short amount of training time when compared to the conventional CNN model. Thus, this confirms our hypothesis that temporal features are more important than spatial features in stock trend prediction.

Using temporal features to train the model is more efficient than that spatial features. Besides, we have also found that the hybrid models performed better than the conventional deep learning models. This is following the research studies in the literature. The contributions of the paper are as follows:

1. The proposed novel LSTM-CNN model defeated the current state-of-the-art model for the stock trend prediction.
2. Performed optimization on models so that all models achieved high accuracies in the stock trend prediction.

3. Discovered that using temporal features to train a model is more efficient than using spatial features for the stock trend prediction.

Bo technique has been provided in the paper. But there remain more optimization algorithms to test on the proposed hybrid model. Future research may cover the efficiency and effectiveness of these different optimization algorithms for stock trend prediction. Secondly, we suggest the model be tested on a stock trend prediction model in real-time trading. Researchers may investigate whether it is worth having a complex model.

Finally, a hybrid LSTM-CNN model for STP has been successfully created by exploiting temporal characteristics to acquire meaningful understanding into spatial aspects of stock data in order to generate predictions. The proposed approach has demonstrated improved performance over the current state-of-the-art in stock prediction, thereby making the automation of technical trading a reality. The approach is helpful in real-life applications for technical traders, whose job involves identifying chart patterns and signals across thousands of stocks before buying or selling a stock.

Acknowledgement

This work is funded by the Ministry of Higher Education Malaysia, Fundamental Research Grant Scheme [Grant No. FRGS/1/2022/STG06/USM/02/4], for the Project entitled "Efficient Joint Process Monitoring Using a New Robust Variable Sample Size and Sampling Interval Run Sum Scheme".

REFERENCES

[1] Abhyankar, A.—Copeland, L. S.—Wong, W.: Uncovering Nonlinear Structure in Real-Time Stock-Market Indexes: The S & P 500, the DAX, the Nikkei 225, and the FTSE-100. Journal of Business and Economic Statistics, Vol. 15, 1997, No. 1, pp. 1–14, doi: 10.1080/07350015.1997.10524681.

[2] Tursoy, T.—Faisal, F.: The Impact of Gold and Crude Oil Prices on Stock Market in Turkey: Empirical Evidences from ARDL Bounds Test and Combined Cointegration. Resources Policy, Vol. 55, 2018, pp. 49–54, doi: 10.1016/j.resourpol.2017.10.014.

[3] Singhal, S.—Choudhary, S.—Biswal, P. C.: Return and Volatility Linkages Among International Crude Oil Price, Gold Price, Exchange Rate and Stock Markets: Evidence from Mexico. Resources Policy, Vol. 60, 2019, pp. 255–261, doi: 10.1016/j.resourpol.2019.01.004.

[4] Gharib, C.—Mefteh-Wali, S.—Jabeur, S. B.: The Bubble Contagion Effect of COVID-19 Outbreak: Evidence from Crude Oil and Gold Markets. Finance Research Letters, Vol. 38, 2021, Art. No. 101703, doi: 10.1016/j.frl.2020.101703.

[5] Ahmed, S.—Hasan, M. M.—Kamal, M. R.: Russia–Ukraine Crisis: The Effects on the European Stock Market. European Financial Management, Vol. 29, 2023, No. 4, pp. 1078–1118, doi: 10.1111/eufm.12386.

[6] Abu-Mostafa, Y. S.—Atiya, A. F.: Introduction to Financial Forecasting. Applied Intelligence, Vol. 6, 1996, No. 3, pp. 205–213, doi: 10.1007/BF00126626.

[7] Deboeck, G. J.: Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets. John Wiley and Sons, 1994.

[8] Smith, D. M.—Faugère, C.—Wang, Y.: Head and Shoulders Above the Rest? The Performance of Institutional Portfolio Managers Who Use Technical Analysis. Research in Finance, Vol. 29, 2014, pp. 167–189, doi: 10.1108/S0196-3821(2013)0000029010.

[9] Wong, W. K.—Manzur, M.—Chew, B. K.: How Rewarding Is Technical Analysis? Evidence from Singapore Stock Market. Applied Financial Economics, Vol. 13, 2003, No. 7, pp. 543–551, doi: 10.1080/0960310022000020906.

[10] Jiang, W.: Applications of Deep Learning in Stock Market Prediction: Recent Progress. Expert Systems with Applications, Vol. 184, 2021, Art. No. 115537, doi: 10.1016/j.eswa.2021.115537.

[11] Ariyo, A. A.—Adewumi, A. O.—Ayo, C. K.: Stock Price Prediction Using the ARIMA Model. 2014 UKSim-AMSS 16^{th} International Conference on Computer Modelling and Simulation, IEEE, 2014, pp. 106–112, doi: 10.1109/UKSim.2014.67.

[12] Wu, Y. P.—Wu, K. P.—Lee, H. M.: Stock Trend Prediction by Sequential Chart Pattern via K-Means and AprioriAll Algorithm. 2012 Conference on Technologies and Applications of Artificial Intelligence, IEEE, 2012, pp. 176–181, doi: 10.1109/TAAI.2012.42.

[13] Shen, S.—Jiang, H.—Zhang, T.: Stock Market Forecasting Using Machine Learning Algorithms. Department of Electrical Engineering, Stanford University, Stanford, CA, 2012.

[14] Picasso, A.—Merello, S.—Ma, Y.—Oneto, L.—Cambria, E.: Technical Analysis and Sentiment Embeddings for Market Trend Prediction. Expert Systems with Applications, Vol. 135, 2019, pp. 60–70, doi: 10.1016/j.eswa.2019.06.014.

[15] Sezer, O. B.—Ozbayoglu, A. M.—Dogdu, E.: An Artificial Neural Network-Based Stock Trading System Using Technical Analysis and Big Data Framework. Proceedings of the SouthEast Conference (ACM SE'17), 2017, pp. 223–226, doi: 10.1145/3077286.3077294.

[16] Gencay, R.: The Predictability of Security Returns with Simple Technical Trading Rules. Journal of Empirical Finance, Vol. 5, 1998, No. 4, pp. 347–359, doi: 10.1016/S0927-5398(97)00022-4.

[17] LeCun, Y.—Haffner, P.—Bottou, L.—Bengio, Y.: Object Recognition with Gradient-Based Learning. In: Forsyth, D. A., Mundy, J. L., di Gesú, V., Cipolla, R. (Eds.): Shape, Contour and Grouping in Computer Vision. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1681, 1999, pp. 319–345, doi: 10.1007/3-540-46805-6_19.

[18] Ozbayoglu, A. M.—Gudelek, M. U.—Sezer, O. B.: Deep Learning for Financial Applications: A Survey. Applied Soft Computing, Vol. 93, 2020, Art. No. 106384, doi: 10.1016/j.asoc.2020.106384.

[19] Barra, S.—Carta, S. M.—Corriga, A.—Podda, A. S.—Recupero, D. R.: Deep Learning and Time Series-to-Image Encoding for Financial Forecasting.

IEEE/CAA Journal of Automatica Sinica, Vol. 7, 2020, No. 3, pp. 683–692, doi: 10.1109/JAS.2020.1003132.

[20] Wang, Z.—Oates, T.: Imaging Time-Series to Improve Classification and Imputation. Proceedings of the 24^{th} International Conference on Artificial Intelligence (IJCAI '15), 2015, pp. 3939–3945.

[21] Gudelek, M. U.—Boluk, S. A.—Ozbayoglu, A. M.: A Deep Learning Based Stock Trading Model with 2-D CNN Trend Detection. 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 2017, pp. 1–8, doi: 10.1109/SSCI.2017.8285188.

[22] Widiputra, H.—Mailangkay, A.—Gautama, E.: Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction. Complexity, Vol. 2021, 2021, Art. No. 9903518, doi: 10.1155/2021/9903518.

[23] Lu, W.—Li, J.—Li, Y.—Sun, A.—Wang, J.: A CNN-LSTM-Based Model to Forecast Stock Prices. Complexity, Vol. 2020, 2020, Art. No. 6622927, doi: 10.1155/2020/6622927.

[24] Liu, S.—Zhang, C.—Ma, J.: CNN-LSTM Neural Network Model for Quantitative Strategy Analysis in Stock Markets. In: Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E. S. M. (Eds.): Neural Information Processing (ICONIP 2017). Springer, Cham, Lecture Notes in Computer Science, Vol. 10635, 2017, pp. 198–206, doi: 10.1007/978-3-319-70096-0_21.

[25] Weng, B.—Ahmed, M. A.—Megahed, F. M.: Stock Market One-Day Ahead Movement Prediction Using Disparate Data Sources. Expert Systems with Applications, Vol. 79, 2017, pp. 153–163, doi: 10.1016/j.eswa.2017.02.041.

[26] Bisoi, R.—Dash, P. K.—Parida, A. K.: Hybrid Variational Mode Decomposition and Evolutionary Robust Kernel Extreme Learning Machine for Stock Price and Movement Prediction on Daily Basis. Applied Soft Computing, Vol. 74, 2019, pp. 652–678, doi: 10.1016/j.asoc.2018.11.008.

[27] Kia, A. N.—Haratizadeh, S.—Shouraki, S. B.: A Hybrid Supervised Semi-Supervised Graph-Based Model to Predict One-Day Ahead Movement of Global Stock Markets and Commodity Prices. Expert Systems with Applications, Vol. 105, 2018, pp. 159–173, doi: 10.1016/j.eswa.2018.03.037.

[28] Gong, J.—Sun, S.: A New Approach of Stock Price Prediction Based on Logistic Regression Model. 2009 International Conference on New Trends in Information and Service Science, IEEE, 2009, pp. 1366–1371, doi: 10.1109/NISS.2009.267.

[29] Kara, Y.—Boyacioglu, M. A.—Baykan, Ö. K.: 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, 2011, No. 5, pp. 5311–5319, doi: 10.1016/j.eswa.2010.10.027.

[30] Yahoo Finance. `https://finance.yahoo.com/quote/%5EGSPC/history`.

[31] Adebiyi, A. A.—Adewumi, A. O.—Ayo, C. K.: Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction. Journal of Applied Mathematics, Vol. 2014, 2014, Art. No. 614342, doi: 10.1155/2014/614342.

[32] Niu, T.—Wang, J.—Lu, H.—Yang, W.—Du, P.: Developing a Deep Learning Framework with Two-Stage Feature Selection for Multivariate Financial Time Series

Forecasting. Expert Systems with Applications, Vol. 148, 2020, Art. No. 113237, doi: 10.1016/j.eswa.2020.113237.

[33] Zhou, J.—He, Z.—Song, Y. N.—Wang, H.—Yang, X.—Lian, W.—Dai, H. N.: Precious Metal Price Prediction Based on Deep Regularization Self-Attention Regression. IEEE Access, Vol. 8, 2019, pp. 2178–2187, doi: 10.1109/ACCESS.2019.2962202.

[34] Alhnaity, B.—Abbod, M.: A New Hybrid Financial Time Series Prediction Model. Engineering Applications of Artificial Intelligence, Vol. 95, 2020, Art. No. 103873, doi: 10.1016/j.engappai.2020.103873.

[35] Tay, F. E. H.—Cao, L.: Application of Support Vector Machines in Financial Time Series Forecasting. Omega, Vol. 29, 2001, No. 4, pp. 309–317, doi: 10.1016/S0305-0483(01)00026-3.

[36] Naeini, M. P.—Taremian, H.—Hashemi, H. B.: Stock Market Value Prediction Using Neural Networks. 2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM), IEEE, 2010, pp. 132–136, doi: 10.1109/CISIM.2010.5643675.

[37] Mehtab, S.—Sen, J.: Stock Price Prediction Using Convolutional Neural Networks on a Multivariate Timeseries. TechRxiv, 2021, doi: 10.36227/techrxiv.15088734.v1.

[38] Nelson, D. M. Q.—Pereira, A. C. M.—De Oliveira, R. A.: Stock Market's Price Movement Prediction with LSTM Neural Networks. 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 1419–1426, doi: 10.1109/IJCNN.2017.7966019.

[39] Fortier, M.: TA-Lib – Technical Analysis Library. `https://ta-lib.org`.

[40] Shin, S.—Lee, Y.—Kim, M.—Park, J.—Lee, S.—Min, K.: Deep Neural Network Model with Bayesian Hyperparameter Optimization for Prediction of NOx at Transient Conditions in a Diesel Engine. Engineering Applications of Artificial Intelligence, Vol. 94, 2020, Art. No. 103761, doi: 10.1016/j.engappai.2020.103761.

[41] Singh, R.—Srivastava, S.: Stock Prediction Using Deep Learning. Multimedia Tools and Applications, Vol. 76, 2017, No. 18, pp. 18569–18584, doi: 10.1007/s11042-016-4159-7.

[42] Bengio, Y.: Practical Recommendations for Gradient-Based Training of Deep Architectures. In: Montavon, G., Orr, G. B., Müller, K. R. (Eds.): Neural Networks: Tricks of the Trade. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7700, 2012, pp. 437–478, doi: 10.1007/978-3-642-35289-8_26.

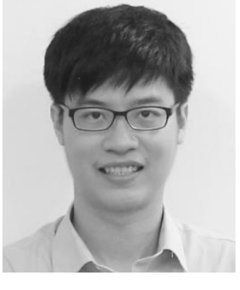
[43] Yang, H.—Zhu, Y.—Huang, Q.: A Multi-Indicator Feature Selection for CNN-Driven Stock Index Prediction. In: Cheng, L., Leung, A. C. S., Ozawa, S. (Eds.): Neural Information Processing (ICONIP 2018). Springer, Cham, Lecture Notes in Computer Science, Vol. 11305, 2018, pp. 35–46, doi: 10.1007/978-3-030-04221-9_4.

[44] Alonso-Monsalve, S.—Suárez-Cetrulo, A. L.—Cervantes, A.—Quintana, D.: Convolution on Neural Networks for High-Frequency Trend Prediction of Cryptocurrency Exchange Rates Using Technical Indicators. Expert Systems with Applications, Vol. 149, 2020, Art. No. 113250, doi: 10.1016/j.eswa.2020.113250.

Bey Kun Chan received his Bachelor's degree (with first-class honor) in the School of Computer Sciences, Universiti Sains Malaysia, in year 2022. Previously he joined a multinational company to perform his internship in the year 2021.

Olanrewaju Victor Johnson received his M.Sc. degree from the University of Benin, Benin City, Nigeria. He studied advanced ICT management at the renowned Galilee International Management Institute, Israel in 2014. He attended CIMPA research school both at the University of Ibadan in 2017 and Univeristy of Lome, Togo in 2018. He obtained a Research Project Management certificate at the Cambridge Moller Institute, University of Cambridge, UK in 2020. He is currently pursuing his Ph.D. at the Universiti Sains Malaysia. His research interests include machine learning, artificial intelligence, data mining, decision making analysis, big data and cloud computing

XinYing Chew received her Ph.D. degree in statistical quality control from the Universiti Sains Malaysia. She is currently a Senior Lecturer at the School of Computer Sciences, Universiti Sains Malaysia. She is also a Certified Trainer with the Human Resources Development Fund (HRDF). She is also an Adjunct Research Fellow of the Swinburne University of Technology Sarawak Campus and a Professional Technologist in the Malaysia Board of Technologists (MBOT). Her research interests include advanced analytics and statistical quality/process control.



Khai Wah Khaw received his Ph.D. degree from the School of Mathematical Sciences, Universiti Sains Malaysia (USM). He is currently a Senior Lecturer at the School of Management, USM. He is a Professional Technologist in the Malaysia Board of Technologists (MBOT). His research interests include statistical process control and advanced analytics.

Ming Ha Lee received her M.Sc. degree in applied statistics from the Universiti Putra Malaysia. Prior to her Ph.D. degree at the Universiti Sains Malaysia, she worked as a Lecturer in a private college, where she was teaching mathematics for matriculation and business statistics for the undergraduate level. She joined the School of Engineering, Swinburne University of Technology Sarawak Campus as a Lecturer in January 2007.

Alhamzah Alnoor works as a Senior Lecturer of strategic management at the Management Technical College, Southern Technical University, Basrah, Iraq. He received his B.Sc. degree with first-class honors from the Southern Technical University in 2013. After completing his B.Sc. in operation management, he was awarded his Master's degree in business administration from the University of Basrah, Iraq, 2016. He received his Ph.D. degree from the School of Management, Universiti Sains Malaysia (USM), Malaysia in 2022. His research interests lie in the areas of strategic management, business ethics, leadership, organizations studies, entrepreneurship, technology adaption, sustainability, and social commerce. He has published numerous scientific journal articles and conference papers. He is a regular reviewer of numerous prestigious journals. During his career, he was engaged in more than 50 projects.