Machine Learning with Applications 9 (2022) 100385

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Predicting NEPSE index price using deep learning models

Nawa Raj Pokhrel [a], Keshab Raj Dahal [b,*], Ramchandra Rimal [c], Hum Nath Bhandari [d], Rajendra K.C. Khatri [e], Binod Rimal [f], William Edward Hahn [f]

[a] Department of Physics and Computer Science, Xavier University of Louisiana, New Orleans, LA, USA
[b] Department of Statistics, Truman State University, Kirksville, MO, USA
[c] Department of Mathematical Sciences, Middle Tennessee State University, Murfreesboro, TN, USA
[d] Department of Mathematics, Roger Williams University, Bristol, RI, USA
[e] Department of Applied Mathematics, Philander Smith College, Little Rock, AR, USA
[f] Department of Mathematical Sciences, Florida Atlantic University, Boca Raton, FL, USA

ARTICLE INFO

Keywords:
Stock market index
LSTM
GRU
CNN
Prediction
Machine learning
Deep learning
Time series
NEPSE

ABSTRACT

Stock price prediction is a prevalent research field in both industry and academia. There is a pressing demand to develop a prediction model that captures the pattern of the financial activities with high precision to make an informed decision. Stock price prediction is challenging due to the complex, incomplete, fuzzy, nonlinear, and volatile nature of financial data. However, developing a robust model is possible due to advancements in artificial intelligence, availability of large-scale data, and increased access to computational capability. This study performs a comparative analysis of three deep learning models—the Long Short-term Memory (LSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Network (CNN)—in predicting the next day's closing price of the Nepal Stock Exchange (NEPSE) index. A set of sixteen predictors is carefully chosen under the domain of the fundamental market data, macroeconomic data, technical indicators, and financial text data of the stock market of Nepal. The performances of employed models are compared using the standard assessment metrics—Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Correlation Coefficient (R). The experimental results show that the LSTM model architecture provides a superior fit with high prediction accuracy. Moreover, statistical evidences are presented to validate the models' reliability and robustness.

1. Introduction

Stocks markets are naturally noisy, non-parametric, non-linear, and deterministic chaotic systems (Ahangar et al., 2010). Thus, predicting market performance is a complex and challenging task for companies, investors, and equity traders. There are many interconnected reasons behind the scenes, both in local and global contexts. Possible causes include, but are not limited to, global economic data, changes in unemployment, the monetary policy of an influential country, immigration policy, natural disasters, and public health conditions. All of this information is disseminated on multiple platforms in diverse forms. Both numerical and textual information may influence stock price forecasts in several ways. This study addresses the major challenges of organizations/individuals to collect multiple pieces of information, stack them into one basket, and build a reliable model for accurate predictions. In particular, investment firms and traders may utilize the developed model as an additional tool for better investment or trading decisions.

Feature selection for financial data is one of the most difficult tasks in stock price prediction, for which many methods have been proposed (Hoseinzade & Haratizadeh, 2019). There has been a trend in which some researchers use only technical indicators, whereas others use historical data (Di Persio & Honchar, 2016; Kara et al., 2011; Nelson et al., 2017; Patel et al., 2015; Qiu & Song, 2016; Wang & Kim, 2018). In the recent decade, users ranging from entrepreneurs to personal investors share their thoughts and opinions through various social media platforms. The stock market is largely driven by new information in the form of Facebook posts, tweets on Twitter, or discussions by people on various other websites that further increase the volatility in stock price prediction. This situation is most likely to continue in the foreseeable future as well. Thus, sentiment analysis of such news could play a vital role in stock price prediction. Current research shows a correlation and a predictive relationship between sentiment and stock prices (Coelho et al., 2019; Seng & Yang, 2017). On top of fundamental data, technical indicators, and macroeconomic data, the authors believe

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: npokhrel@xula.edu (N.R. Pokhrel), kdahal@truman.edu (K.R. Dahal), ramchandra.rimal@mtsu.edu (R. Rimal), hbhandari@rwu.edu (H.N. Bhandari), rkhatri@philander.edu (R.K.C. Khatri), brimal2014@fau.edu (B. Rimal), whahn@fau.edu (W.E. Hahn).

https://doi.org/10.1016/j.mlwa.2022.100385
Received 14 May 2022; Received in revised form 20 July 2022; Accepted 20 July 2022
Available online 25 July 2022

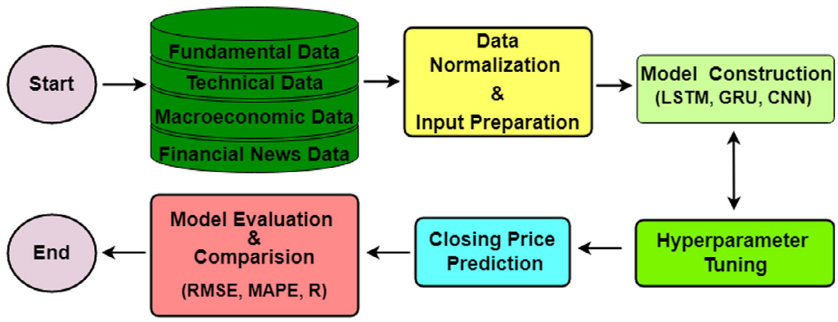


Fig. 1. Schematic diagram of the proposed research framework.

the positive contribution of the sentiment of social media news in stock price prediction (Wang & Wang, 2016). Thus, natural language processing and sentiment analysis techniques are implemented to tie up the stock market with news sentiment to improve prediction accuracy.

Various theories are available in the market to develop an efficient business model for better-informed investment decisions. With very few exceptions, almost all theories such as the efficient market hypothesis, random walk theory, mean reversion, martingales, and others help to cope with specific market situations. Mean reversion is a phenomenon where the time series reverse to the mean, retracing to their long-run average. Investors can utilize this phenomenon when the market has gone far away from the mean and expect the gravitational pull to return the stock price to the mean (Fama & French, 1988; Poterba & Summers, 1988; Wei, 2020; Zhao et al., 2013). The martingale strategy (Barnett & Serletis, 2000; Danthine, 1977) is a methodology based on mean reversion that is applied to amplify the chance of recovering from losing trades. These strategies are established in forex trading but also can be used by equity index investors.

There are two schools of thought in developing the predictive model to estimate the stock price. Classical thinking holds that future stock prices are predicted based on historical facts and indicators (Hansen et al., 1999). These methods include variations in Autoregressive Integrated Moving Average (ARIMA) models. These statistical models are well suited to describe the linear relationships but may not be equally efficient in capturing the noisy and nonlinear behavior. The ARIMA is used to handle only stationary time series data by default and one must convert non-stationary time series data to stationary. On the other hand, the modern theory presumes that no good pattern can be found from the historical data to reflect the exact upcoming structure due to the inherent nonlinearity in the stock market data (Wang & Leu, 1996). The rapid advancement in artificial intelligence and machine learning techniques, availability of large-scale data, and increased computational capabilities opens the door to developing robust machine learning models (Hill et al., 1996) to capture the nonlinear behavior to predict the stock price.

A plethora of research has been published to predict the stock price using variations of deep learning techniques. A varying degree of success is seen with the accuracy and robustness of the models. Most of the published research articles use single-layer to multi-layer ANN architectures and show reasonable results with or without sentiment analysis of the subject area (Shahi et al., 2020). The most widely used deep learning architecture is long short-term memory (LSTM), convolution neural network (CNN), gated recurrent unit (GRU), and their respective hybridization techniques (Althelaya et al., 2018; Chen et al., 2015; Cho et al., 2014; Jiawei & Murata, 2019; Ju et al., 2021, 2022; Li et al., 2017; Samarawickrama & Fernando, 2017; Shen et al., 2018). Research studies often employ the stated methods to speak of oneself with pride about the model's accuracy in every new publication. However, their working framework, underlying series of assumptions, and actual implementation differ from one to another. The implementation of various stock market databases and several attributable variables is considered. Thus, it is tough to make an unbiased comparison between previously published research articles that use the same model to predict the stock price. Furthermore, very little has been done to incorporate the financial news sentiments in stock price prediction and their implementation in a different scenario.

The primary objective of this article is to conduct a comparative study of LSTM, GRU, and CNN models under identical conditions by utilizing the multifaceted information of stock data. To accomplish the stated goal, we have incorporated the balanced set of input features consisting of fundamental data, macroeconomic data, technical indicators, and the sentiment score of the financial news data to predict the stock price. Stock data in most developed countries is available on heterogeneous platforms, and financial news is broadcast from various media outlets. Given the availability of mixed bag information of news sentiment of stock market data from numerous media platforms, it is more reasonable to select the stock market data from underdeveloped countries where media outlets are more consistent and accessible than in developed countries. Hence, the Nepal Stock Exchange Limited (NEPSE) dataset, the only stock exchange in Nepal, and its associated financial news are selected for this study. The primary objective of NEPSE is to promote the marketability and liquidity of listed corporate securities by providing an online trading platform via market intermediaries and regulating trading activities. Based on the most recent data, there are 215 companies, 50 broker companies, and 31 merchant bankers listed in NEPSE (Acharya & Koirala, 2021).

Our complete vision to achieve the stated goal can be conceptualized from an aerial perspective via the schematic diagram in Fig. 1. The proposed study carefully utilizes the selected features from fundamental, macroeconomic, technical, and financial news data to build the model. The collected data has been normalized using the min–max technique. The input sequence for the LSTM, GRU, and CNN model is created using a specific time step. The hyperparameters such as the number of neurons, number of filters, filter size, epochs, learning rate, batch size, and time step are tuned using regularization techniques to optimize the model's performance and to overcome the over-fitting problems. Once the hyperparameters are tuned, the final model is fit to predict the closing price of the stock market index. The quality of the proposed model is assessed through RMSE, MAPE, and R scores.

The main contributions of this study include (a) Gathering the multifaceted information of NEPSE index, putting them together into a common framework, and constructing a reliable model for accurate predictions. (b) Conducting extensive data driven experimentation using customized parameters of LSTM, GRU, and CNN models. (c) Performing comparative study of deep learning models (LSTM, GRU, and CNN) for the best fit and forecasting under the identical conditions. (d) Conducting statistical experiment to validate and verify the reliability and robustness of the model.

The rest of the paper is as follows. Section 2 explains the related work in this field. The data collection and feature selection procedure is explained in Section 3. Hyperparameter tuning and the comparison of models' outcomes are discussed in Section 4. Section 5 discuss the experiment and results. Section 6 discuss the ethics and implications. Finally, Section 7 presents the conclusion and future work, followed by acknowledgments, and a list of references.

Table 1
List of potential features for the model.

Data	Source	Frequency	Abbreviation
Fundamental:			
Open Price	Share Sansar	Daily	...
High Price	Share Sansar	Daily	...
Low Price	Share Sansar	Daily	...
Close Price	Share Sansar	Daily	...
Volume	Share Sansar	Daily	...
Macroeconomic:			
Remittance	Share Sansar	Monthly	RMT
Inflation Rate	Share Sansar	Monthly	IR
Commercial Bank Interest Rate	Share Sansar	Monthly	CBIR
Treasury Bill	Share Sansar	Monthly	TRB
Consumer Price Index	The Global Economy	Monthly	CPI
Exchange Rate to US Dollar	The Global Economy	Monthly	ER
Technical Indicator:			
Moving Average Convergence Divergence	...	Daily	MACD
Average True Range	...	Daily	ATR
Relative Strength Index	...	Daily	RSI
Money Flow Index	...	Daily	MFI
Financial News:			
Sentiment Score	Share Sansar	Daily	Score

2. Related work

In 1993, Pradhan studied the stock market behavior in a small capital market in the context of Nepal (Pradhan, 1993). He examined the relationship between market equity, market value to book value, price-earnings, and dividends related to liquidity, leverage, profitability, turnover, and interest coverage. The study was based on pooled cross-sectional data of 17 enterprises, and their stocks were listed in the Stock Exchange Center and traded in the stock market.

In 2004, Gurung studied the growth trend and performance of NEPSE (Gurung, 2004). The author analyzed the features such as the number of listed and traded companies and their securities, the number of transactions, trading turnovers, paid-up value, market capitalization, and the NEPSE index. He used the secondary data obtained from SEBON and NEPSE for the ten years of corporate securities. The study concluded that there was no synchronization between different securities market performance indicators, but it was true that they almost depicted an erratic trend during the observed period. In 2010, Neupane studied the Nepalese stock market, where they concluded that future stock prices could be estimated by analyzing the historical information and with the help of proper technical indicators of the market (Neupane, 2010).

In 2017, Bhusal focused on identifying the long-run behavior of the NEPSE index using the Markov chain model (Bhusal, 2017). He grouped NEPSE index data into three states —increment, constant, and decrease. He was particularly interested in determining the expected number of visits to a particular state and finding out the expected first return time of the states. Furthermore, the author concluded that the movement of the stock index to various states on a particular trading day is independent of the index of initial trading days but depends only on the index of the most recent day. After a year, Pun and Shahi implemented the Support Vector Regression and Artificial Neural Network to predict the next day's stock price (Pun & Shahi, 2018). The data obtained from NEPSE was divided into ten investment sectors, and the outcome of each sector was evaluated via various assessment metrics.

In 2019, Saud and Shakya compared stock price prediction accuracy obtained from GRU with momentum, RMSProp, and Adam optimization technique using two individual stocks traded in NEPSE (Saud & Shakya, 2019). The study concluded that the GRU with Adam showed more accurate and consistent prediction accuracy. After a year, they performed a comparative analysis to predict the next day's closing price based on the look-back period using different deep learning techniques —Vanilla RNN, LSTM, and GRU (Saud & Shakya, 2020). They considered attributable variables such as fundamental stock data, multiple variations of moving average, relative strength index, and William %R for two commercial banks. The study concluded that GRU and LSTM outperformed RNN in predicting stock price.

In 2020, Shahi et al. conducted a comparative study to predict the stock price using long short-term memory (LSTM) and gated recurrent unit (GRU) models (Shahi et al., 2020). Authors incorporated the financial news sentiments with the stock features as the input in stock market forecasting. The authors believed that one possible way of improving the deep learning model performance on stock market prediction could be by adding financial news sentiment data.

3. Data description and preparation

Nepal is a culturally diverse landlocked country, only covering about 0.1% of the world's total area (Paudel et al., 2012). It is a sandwich between two giant economies China and India. Due to geographic proximity and open border, Nepal's stock market is significantly influenced by the economic activities happening in these countries. Hence, the unique geographical landscape and its peculiar identity in the world, Nepal's stock market is equally influenced by domestic and external issues. After meticulously analyzing the local and global context, we broadly categorized the most affecting factors to the stock market in terms of fundamental, macroeconomic, technical, and financial news data listed in Table 1. We converted monthly data to daily data by reverse padding to ensure consistency between variables.

3.1. Fundamental data

The first set of variables presented in Table 1 is fundamental or historical data that provides the basic information required for stock trading. It consists of the open price, close price, high price, low price, and volume. Open price is the first transaction price on the opening of a market on a trading day, whereas the closing price is the last price at which the stock is traded during that day. Similarly, high and low prices are the highest and lowest price of the day. Volume refers to the number of shares traded during the trading day, indicating the trader's interest in the trading activity. High trading volume means greater interest and vice versa. All historical trading data accessed from the portal Share Sansar (https://www.sharesansar.com), one of the popular stock trading sources that provide detailed information about the Nepal stock market, is daily data.

3.2. Macroeconomic data

The second set of variables demonstrated in Table 1 is macroeconomic variables that significantly influence stock market performance. The representative features that affect the stock price prediction under the umbrella of macroeconomic factors are Remittance (RMT), Inflation Rate (IR), Commercial Bank Interest Rate (CBIR), Treasury Bill (TRB), Consumer Price Index (CPI), and Exchange Rate to US Dollar (ER).

- **Remittance:** The remittance is the money sent by Nepali workers from abroad. For the last ten years, the remittance has been serving about a quarter of the total gross domestic product of Nepal (Ojha, 2019). Nepal is increasingly dependent on remittance as being a significant labor exporting country. The central bank of Nepal, individual researchers, and various independent organizations are exploring the impact of remittance on the broader economy independently. All of them agree on the fact that the inflow of money via migrant remittance is playing an increasingly important role in economic growth (Katuwal, 2021; Salahuddin & Gow, 2015). The remittance inflow is less volatile, and it serves the recipient economy steadily even in times of economic downturns (Ahmed & Martinez-Zarzoso, 2013). Thus, it can have a significant impact on the broader stock market index of Nepal.
- **Inflation Rate:** Inflation refers to the increase in the price of commodities and a decrease in the purchasing power of money. When the inflation rate is high, consumer spending on Giffen goods increases, reducing investment and slowing economic growth. The inflation rate significantly impacts the economic growth of the country like Nepal (Bank, 2017; Bhusal & Silpakar, 2011; Chaudhary & Xiumin, 2018) thus motivating us to incorporate it as the predictor variable in the stock price prediction.
- **Exchange Rate:** An exchange rate is the value of one nation's currency to another. Nepal has been adopting a dual exchange rate arrangement since February 12, 1993; the Nepali currency (NC) is pegged with the Indian currency (IC), whereas it floats with the convertible currencies (Pant & Budha, 2016). The appreciation of the exchange rate hurts export competitiveness, impacting economic growth (Adhikari, 2018; Paudel & Burke, 2015). As the economy and the broader stock market are interrelated, the variable Exchange Rate to US Dollar is included as a predictor in the model.
- **Consumer Price Index:** The Consumer Price Index (CPI) is a measure that examines the weighted average prices of a basket of consumer goods and services. CPI is the most widely used measure of inflation and, by proxy, the government's economic policy. There is always long-run relationship of the effect of consumer price index with stock market data (Devkota, 2018; Panta, 2020; Shrestha & Pokhrel, 2019).
- **Treasury Bill:** The treasury bill is typically a promissory maturity note issued by a government as a primary instrument for regulating money supply and raising funds via open market operations. The Nepal Rastra Bank issues treasury notes with the short-term maturity (usually three months) and the long-term maturity (usually a year). There is long-run causality of the treasury bill rate with the NEPSE stock index and even a negative relationship between treasury bill rates to the stock index price (Gurung, 2019; Khatri, 2019; Rakhal, 2018) thus it is incorporated as a potential predictor in the model.
- **Commercial Bank Interest Rate:** The commercial bank interest rate is the bank rate at which Nepal Rastra Bank lends money to domestic financial institutions. With the increase/decrease in the interest rate, the customers' (or investors') cost of money also increases/decreases. There is an inverse relationship between investment to the interest rate (Devkota & Panta, 2018; Neupane, 2018; Thapa, 2019; Timsina, 2017) thus included as the plausible feature to predict the next day closing price.

3.3. Technical indicator

The third set of variables demonstrated in Table 1 is the technical indicators, including Moving Average Convergence Divergence (MACD), Average True Range (ATR), Relative Strength Index (RSI), and Money Flow Index (MFI). MACD is calculated by subtracting the 26-day exponential moving average (EMA) from the 12-day EMA (Murphy, 1999). ATR measures the market volatility and is defined as follows:

$$\text{True Range (TR)} = \max\{(H-L), |H-C_P|, |L-C_P|\}$$

$$\text{Average True Range (TR)} = \frac{1}{n}\sum_{i}^{n} TR_i$$

where, H, L, C_p represent current high, current low, previous close prices and TR_i, n represent a particular true range, the time period respectively.

The RSI is a momentum indicator that signals whether a security is overbought or oversold with current price levels, which is computed as follows:

$$RSI = 100 - \frac{100}{1 + \frac{Average\ gain}{Average\ loss}}$$

where average gain and average loss are the average percentage gain and loss calculated over the certain look-back period.

The MFI communicates a possible reversal in time and provides a signal for further investment. Its calculation starts by finding the typical price (TP), the average of high, low, and close prices for each trading day. If the current TP is higher than the previous one, then positive money flow is calculated by multiplying the current TP with its volume. Similarly, a negative money flow is obtained if the current typical price is lower than the previous one. If the TP does not change, both positive and negative money flow will be zero. Summing all positive money flow indexes leads to positive money flow for a particular period. Negative money flow is calculated similarly. Mathematically, MFI is defined as:

$$\text{MFI} = 100 - \frac{100}{1 + \text{Money Ratio}}.$$

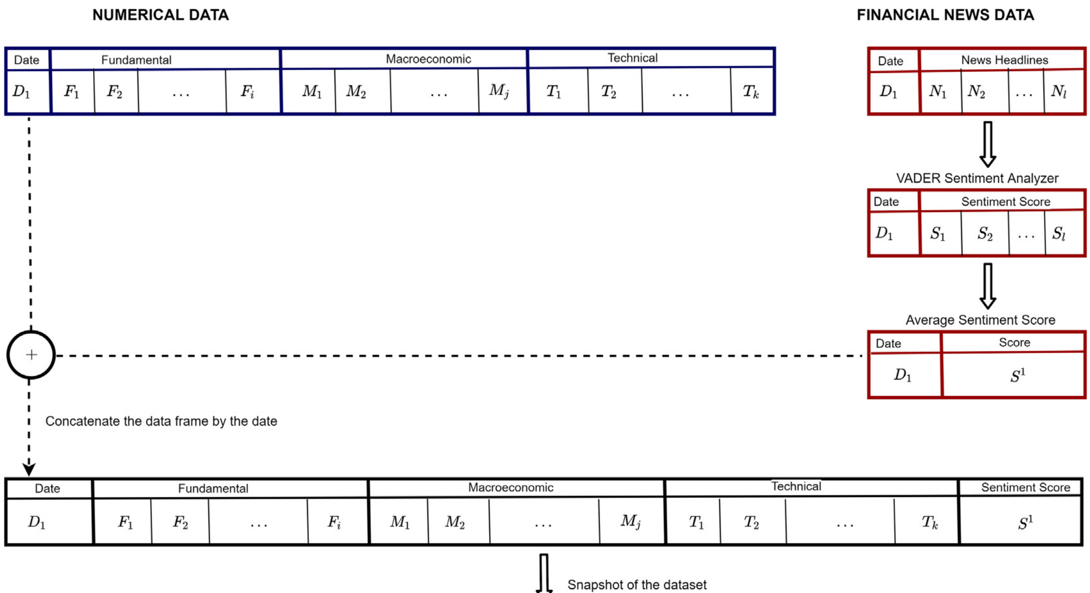
The money ratio is obtained by dividing the positive money flow by negative money flow during the look-back period.

The 14 days look-back period is used to calculate the technical indicators in general. Traders extensively use them in the market as they are designed to analyze short-term price movements (Anghel, 2015; Chong & Ng, 2008; Chong et al., 2014; Eric et al., 2009; Murphy, 1999; Rodríguez-González et al., 2011; Wang & Kim, 2018; Wilder, 1978).

3.4. Financial news data

The last set of variables listed in Table 1 is financial news data. The financial news data of NEPSE published from multiple news outlets was collected from Share Sansar through web scraping. After the necessary data prepossessing, the VADER package is used to calculate the news sentiment score (Hutto & Gilbert, 2014). The news score is the average of the multiple news scores for a day. Finally, the sentiment scores for the financial news data are aligned with the stock market data, as shown in Fig. 2.

The data prepossessing and a snapshot of the sample data are outlined in Fig. 2. The technical indicators defined in Section 3.3 are calculated using fundamental data. Since the fundamental data are the daily data, the technical indicators calculated using them also become daily data. All the macroeconomic variables are available monthly, and they are converted to daily data through the backward filling. Since no financial news data was recorded prior to 2016-07-17, we used information on all variables from that date to October 15, 2020, as data for sentiment scores. The observations on each variable are concatenated with the date as an index. In addition, there were no missing values and outliers of selected features within the time frame.



Date	Fundamental					Macroeconomic						Technical				Sentiment
	Close	Open	High	Low	Volume	CPI	ER	IR	TB	CBIR	RMT	MACD	RSI	ATR	MFI	Score
2016-07-17	1745.74	1718.15	1749.96	1715.14	2272045	112.9	107.52	8.6	0.44	0.82	51.94	-37.51	74.94	23.12	70.20	0.21
2016-07-18	1786.59	1745.74	1786.68	1745.74	2870497	112.9	107.52	8.6	0.44	0.82	51.94	-41.07	82.96	24.40	67.20	0.22
2016-07-19	1800.47	1786.59	1813.42	1785.33	2902127	112.9	107.52	8.6	0.44	0.82	51.94	-44.50	84.86	24.66	65.49	-0.03
2016-07-20	1786.84	1800.47	1813.44	1782.57	3474801	112.9	107.52	8.6	0.44	0.82	-45.60	75.34	25.10	55.04	51.94	0.15
2016-07-21	1798.83	1786.84	1804.75	1779.71	3154492	112.9	107.52	8.6	0.44	0.82	-46.89	77.86	25.10	67.33	51.94	0.13
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2020-01-14	1263.38	1271.11	1274.86	1254.30	3766982	132.38	113.95	6.82	3.17	1.76	447.26	-27.60	82.53	15.11	64.22	0.11
2020-01-15	1284.23	1263.32	1287.20	1263.32	2179674	132.38	113.95	6.87	3.91	4.59	513.21	-30.87	86.19	15.74	70.88	0.22

Fig. 2. Concatenation of fundamental, macroeconomic, technical and financial news data.

4. Modeling approach

4.1. Long short-term memory

LSTM is a popular deep learning technique in RNN for time series prediction. While standard RNNs outperform traditional networks in preserving information, they are not very effective in learning long-term dependencies due to the vanishing gradient problem (Hochreiter, 1998). LSTM uses memory cells to overcome the issue of vanishing gradients. It consists of an input layer, a hidden layer, a cell state, and an output layer (Gers et al., 2000, 2003; Hochreiter & Schmidhuber, 1997). The main component of LSTM architecture is the cell state, which runs through a sequence with only linear interaction, keeping information flow unchanged. The gate mechanism of LSTM deletes or modifies the information of the cell state. It passes the information selectively that consists of the sigmoid layer, hyperbolic tangent layer, and the pointwise multiplication operation.

Fig. 3 illustrates the architecture of LSTM at time t which is designed to model sequential input. In particular, four gates —output, change, input, and forget —are shown with their operations at time t.

For a given input sequence $\{x_1, x_2, \dots, x_n\}$, $x_t \in \mathbb{R}^{k\times 1}$ is the input sequence at time t. The memory cell c_t updates the information using three gates: input gate i_t, forget gate f_t, and change gate $\tilde{c}_t$. The hidden state h_t is updated using output gate o_t and the memory cell c_t. At time t, the respective gates and layers compute the following functions:

$$i_t = \sigma(W_i x_t + W_{hi} h_{t-1} + b_i),$$
$$f_t = \sigma(W_f x_t + W_{hf} h_{t-1} + b_f),$$
$$o_t = \sigma(W_o x_t + W_{ho} h_{t-1} + b_o),$$
$$\tilde{c}_t = \tanh(W_c x_t + W_{hc} h_{t-1} + b_c),$$
$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t,$$
$$h_t = o_t \otimes \tanh(c_t)$$

where, σ and $\tanh$ represent the sigmoid and hyperbolic tangent functions respectively, the operator $\otimes$ is the element-wise product, $W \in \mathbb{R}^{d\times k}, W_h \in \mathbb{R}^{d\times d}$ are weight matrices, and $b \in \mathbb{R}^{d\times 1}$ is bias vector. Moreover, n, k, d are sequence length, the number of features, and the hidden size respectively (Bhandari et al., 2022; Greff et al., 2017; Lei et al., 2019; Qiu et al., 2020).

LSTM cells require three different kinds of information —input sequence x_t, the short-term information from the previous cell h_{t-1}, and the long-term information from the previous cell state c_{t-1} at time t. The forget gate takes the information from x_t and h_{t-1} and produces the output between 0 and 1 through the sigmoid layer. Then, it identifies the information to forget from the former cell state c_{t-1}. It stores all the information in the cell if the output is 1. On the other hand, it forgets all the information from the previous cell state if the output is 0. Similarly, the input gate identifies the information that needs to update from the change gate. The output gate decides which information to be taken as an output from the present cell state.

4.2. Gated recurrent unit

Gated recurrent unit (GRU), a simplified version of the LSTM, was first developed by Chung et al. in 2014 (Chollet, 2017). The short-term (h_t) and long-term (c_t) information of LSTM are merged into a single vector h_t in GRU. As opposed to the four gates in LSTM, GRU has three gates: reset gate, change gate, and update gate. The update gate of GRU is equivalent to the forget gate and input gate of LSTM (Géron, 2019). Thus, a single gate decides what to forget and update in GRU instead of

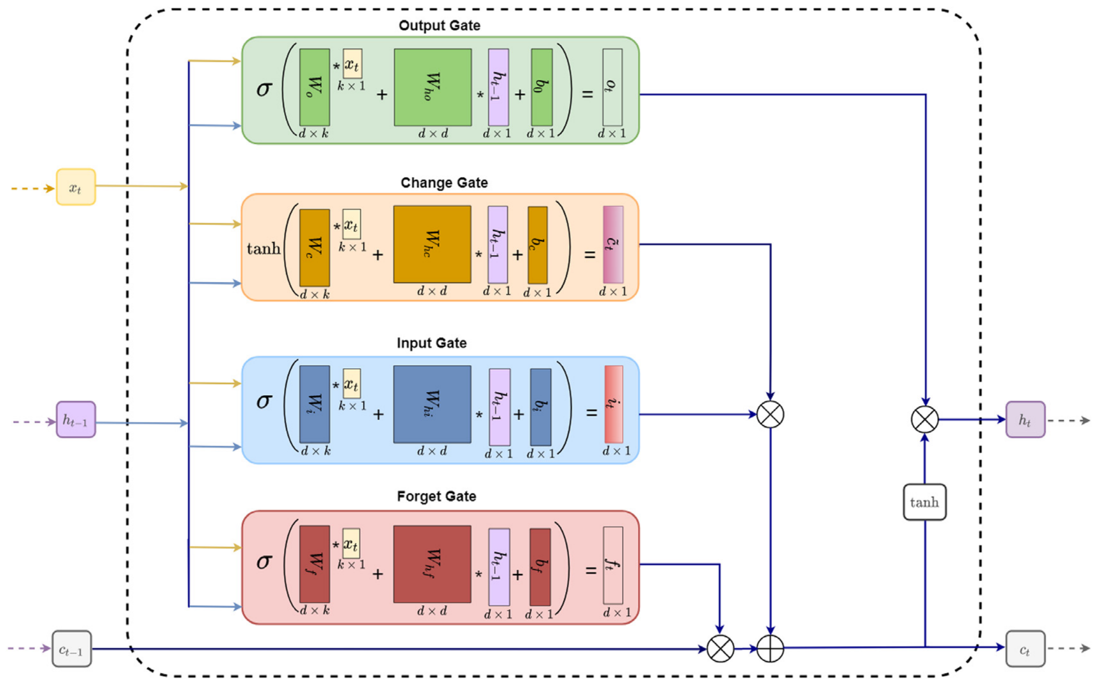


Fig. 3. Long short-term memory (LSTM) architecture.

two gates in LSTM. GRU cell takes two different pieces of information: the current input sequence x_t, the short-term memory from the previous cell h_{t-1}, at time t. The update gate carries the long-term dependencies in GRU. It determines the past information that needs to be passed into the next step. The reset gate takes the information from x_t and h_{t-1} and produces the output between 0 and 1 through the sigmoid layer and then it identifies which information to discard from the previous hidden state h_{t-1}. When the value is 1, it stores all the information in the cell while with a value of 0, it forgets all the information from the previous hidden state. Based on empirical evidence, LSTM and GRU have proven their effectiveness on many machine learning tasks (Agarap, 2018; Chorowski et al., 2015; Wen et al., 2015; Yang et al., 2016). The operations in each gate are shown in Fig. 4.

For a given input sequence $\{x_1, x_2, \dots, x_n\}$, $x_t \in \mathbb{R}^{k\times 1}$ is the input sequence at time t. It takes the input x_t and the hidden state h_{t-1} from the previous time step $t-1$ at time t. It outputs a new hidden state h_t and forwards it again to the next time step (Zhang et al., 2021). At time t, the respective gates and layers compute the following functions:

$$u_t = \sigma(W_z x_t + W_{hz} h_{t-1} + b_u),$$
$$r_t = \sigma(W_r x_t + W_{hr} h_{t-1} + b_r),$$
$$\tilde{h}_t = \tanh(W_c x_t + W_{hc}(r_t \otimes h_{t-1}) + b_c),$$
$$h_t = (1-u_t) \otimes h_{t-1} + u_t \otimes \tilde{h}$$

where, σ and $\tanh$ represent the sigmoid and hyperbolic tangent functions respectively, the operator $\otimes$ is the element-wise product, $W \in \mathbb{R}^{d\times k}, W_h \in \mathbb{R}^{d\times d}$ are weight matrices, and $b \in \mathbb{R}^{d\times 1}$ are bias vectors. Moreover, n, k, d are sequence length, the number of features, and the hidden size respectively.

4.3. Convolution neural networks (CNNs)

A Convolutional Neural Network (CNN) is a popular deep learning technique widely used in computer vision. It is inspired by Hubel and Wiesel's experiments on animals' natural visual perception mechanism (Hubel, 1959). Motivated by their experiments, Fukushima and Miyake developed a neural network model for the mechanism of visual pattern recognition (Fukushima & Miyake, 1982). In 1990, Lecun et al. developed a handwritten digit recognition model that is considered a base model for current CNN architecture (LeCun et al., 1989). There are many variations and benchmark models of CNN so far. CNN models are also used in natural language processing, voice recognition, and stock price prediction. In this study, we discuss the CNN architecture, particularly for time-series predictions.

CNN architecture has the following components: input, convolutional layer with the non-linear activation function, pooling layer, fully connected layer, and output. All layers of the CNN have training parameters except the pooling layer. The number of convolutional, pooling and fully connected layers varies based on the complexities of the task. Generally, a higher number of layers is used for a complex task.

For a given multivariate time series $\{x_1, x_2, \dots, x_n\}$, $x_t \in \mathbb{R}^{k\times 1}$, a matrix of size $T_s \times k$ is formed as an input image, where T_s is the time step and k is the number of input variables. The CNN views a time step as a sequence over which convolutional operations can be performed as a one-dimensional image. Since each series has observations at the same time step, the input time series are parallel. We can reshape these three arrays of data (no. of samples, time-step, no. of features) as a single dataset where each row is a time step and each column is a separate time series (Brownlee, 2018b, 2018c). Thus, from n observations, we will have $n - T_s$ (since we do not include the last observation in the input as the last row will be the output of the previous step input) many matrices of size $T_s \times k$ as in LSTM, each matrix is treated as an image of size $k \times T_s$ in CNN.

The convolution operations are performed on the time axis with filters of size $l \times k$, where l is the length of a filter, a hyperparameter to be tuned. The number of filters m and convolution stride s are other hyperparameters to be chosen based on experiments or domain knowledge. Then, from m many filters, we get m many univariate series of length $T_{s1} = \lfloor \frac{T_s - l}{s} + 1 \rfloor$.

After convolution operations with m filters, we apply non-linear activation function on feature map, then apply pooling operation of size $p_s \times 1$ for each univariate feature with a stride of p_s. Then the size of the feature map after pooling is $\lfloor \frac{T_{s1} - p_s}{p_s} + 1 \rfloor \times m$.

A series of feature maps obtained from multiple convolutions and pooling operations represent the original time series. In a fully connected layer, we connect all the feature maps to generate a new long

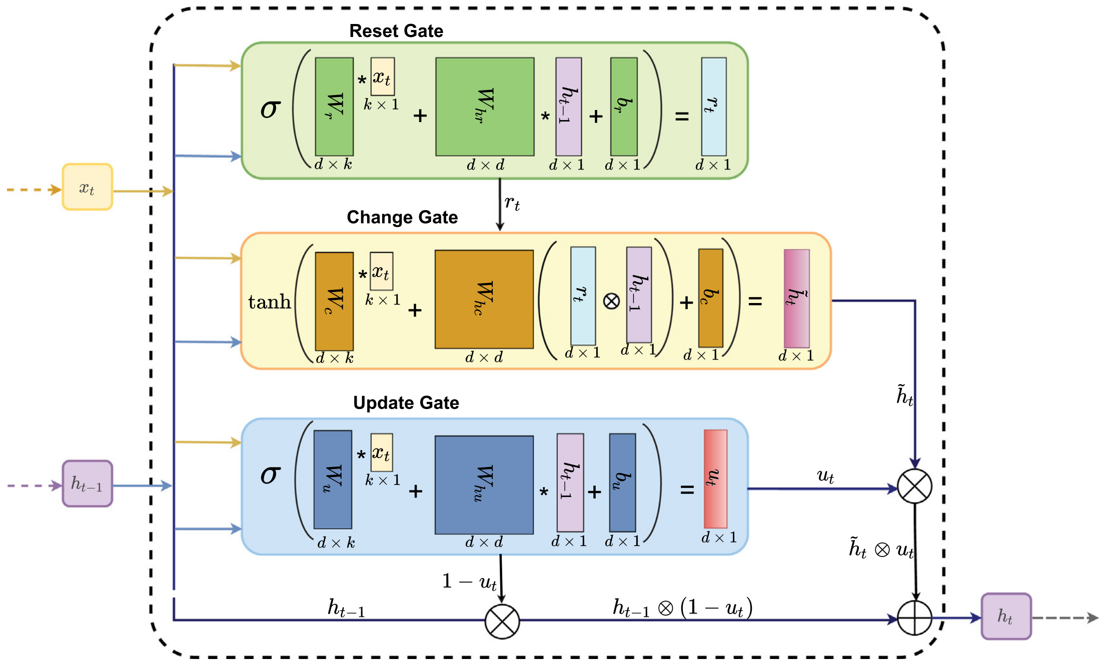


Fig. 4. Gated Recurrent Unit (GRU) architecture.

time series as the final representation of the original input. We use a linear activation function on the time series from fully connected layers to predict closing prices. The network compares the output with the true value to calculate the error.

In Fig. 5, we take a data frame of size $n \times k$ with n observations and k many input variables. For the input to the CNN, we take time steps 5 ($T_s = 5$), so the size of an input image is $k \times 5$. For each image, we use m filters of size $2 \times k$ and slide the filter on the time axis with stride 1. Then, after the convolution operation, we get m feature maps from m filters. After convolution operations, we use non-linear activation functions such as ReLU or Leaky ReLU. A pooling of size 2×1 with stride 2 is performed for downsampling. Then, the feature maps from each filter are vectorized into a single sequence and formed a fully connected layer. Finally, the output $\hat{y}_1$ is predicted using the linear activation function.

5. Experimental design and results

The primary objective of this study is to conduct a comparative analysis of the performance of LSTM, GRU, and CNN models for stock price prediction. Fig. 6 shows the overall trend of the response variable, the closing price of the NEPSE. It exhibits complex, noisy, and volatile behavior. The black curve represents the original time series of the closing price (vertical axis) in the interval of 07/17/2016 —01/15/2020 (horizontal axis). Similarly, green and blue curves represent 20-day and 50-day moving averages to visualize the short-term and long-term trends of the closing price. We see a unique closing price pattern from an overall perspective, showing a downtrend in the given time frame despite various irregularities.

In order to achieve the stated goal, the overall experiment is divided into the five different phases: (a) setting up environment, input/output sequences, and performance metrics, (b) model construction and hyperparameter tuning, (c) identifying the best performing models from respective architectures, (d) identifying overall best model, and (e) performing statistical analysis.

5.1. Setting up environment, input/output sequences, and performance metrics

Table 2 summarizes the computational framework of the experiments, making the comparison of model results consistent and reliable.

Table 2
Computing environmental setup.

Machine Configuration	Google Colab with NVIDIA-SMI 495.44 GPU
Environment	Python 3.6.0, TensorFlow, and Keras APIs
Architecture	LSTM, GRU, CNN

The experiments use the python programming environment with the TensorFlow and Keras APIs. Also, the Machine configuration and employed architecture used in the experiments are listed in the Table.

As part of input/output preparation, the original dataset shown at the bottom of the Fig. 2 is split into a training set and a test set in the 4:1 ratio. Among the training data, 20% of the data is separated for validation, accounting for 16% of the total data. Furthermore, the datasets are normalized using a min–max normalization technique which scales each feature into the range of [0,1]. The transformed datasets are in the form of 2D arrays (number of observations, number of features). However, our model architecture requires 3D input data. Thus, the data are converted into 3D arrays (number of observations, time step, number of features) by incorporating the time step before feeding into the model. The validation set is used during the hyperparameter tuning process. Once we tuned the hyperparameters, the validation set is included back in the training data. Then, the models are trained on the complete training data with optimized hyperparameters. The test data is used to compare the performance of the trained models. Prediction accuracy from the constructed model is assessed through the three different performance metrics —RMSE, MAPE, and R (Bhandari et al., 2022). These metrics help us determine the best model in terms of accuracy and reliability.

5.2. Model construction and hyperparameter tuning

We construct deep learning models where each model consists of an input layer, an LSTM/GRU/CNN layer, and a dense output layer with linear activation. Early stopping criteria are implemented to address the consequences of overfitting that can occur when training neural networks. It allows us to specify a large number of epochs and stop training when the model's performance stops improving on the validation data (Brownlee, 2018a).

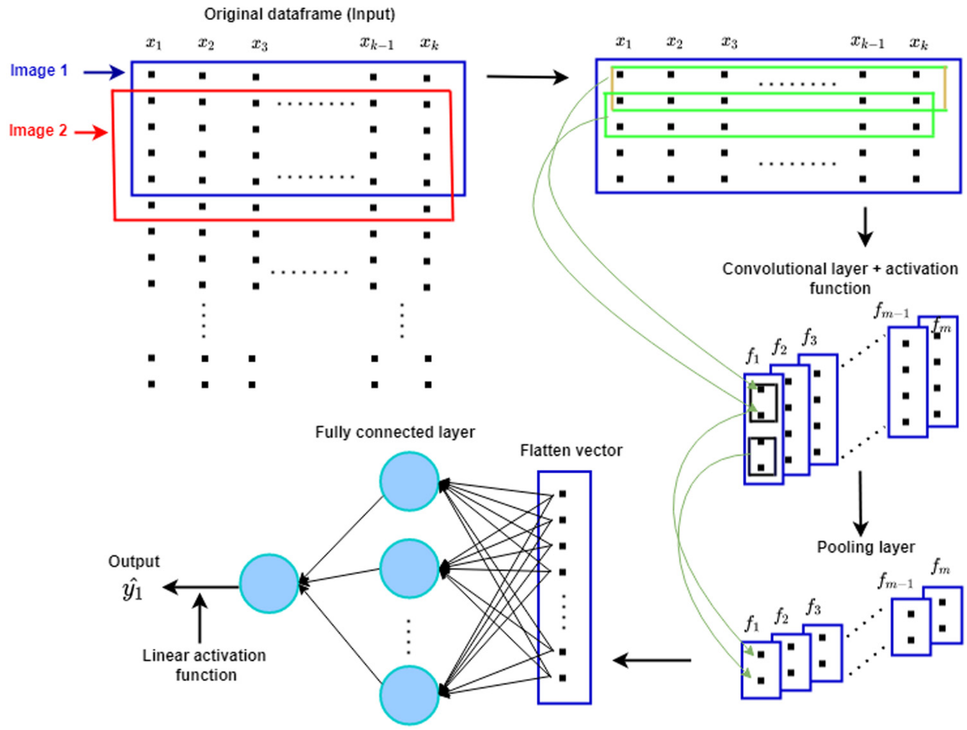


Fig. 5. CNN architecture with m filters for multivariate time series prediction.

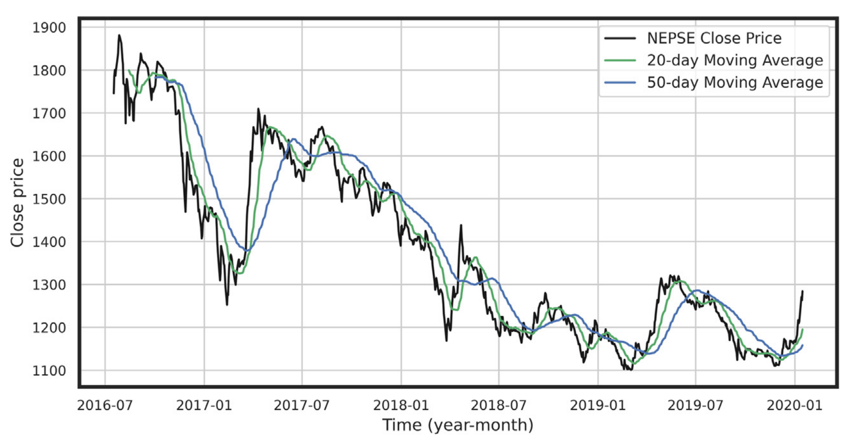


Fig. 6. NEPSE closing price along with moving averages.

After constructing a model, we perform a hyperparameter tuning process where each model identifies its best set of hyperparameters from multiple avenues. This includes three different optimizers (Adam, Adagrad, and Nadam); three different learning rates (0.1, 0.01, and 0.001); and three options for batch sizes (4, 8, and 16). Therefore, 3*3*3 = 27 possible choices are available for each model to identify its best combination. We perform ten independent replicates of each model before calculating the average scores to address the model's stochastic behavior. The selection of the best model relies on the smallest possible average RMSE score calculated in the validation data set. Thus, we execute a total of 3 (architectures)* 7 (models for each architecture)* 27 (possible combinations for each model) = 567 instances during the complete hyperparameter tuning process. The optimal set of hyperparameters for each model architecture is presented in Table 3.

5.3. Identifying the best performing models from respective architectures

Once the hyperparameters are tuned, the models are set with their corresponding optimized hyperparameters. Finally, a total of (7*3 =21) models are trained in full scale and then fitted on the test data to verify their performances and reliability. To address the stochastic behavior of the deep learning models, we have replicated each of the models 25 times. Fig. 7 illustrates the graphical representation of the average scores produced from the employed model architectures (LSTM, GRU, CNN). The subplots (a), (b), and (c) reveal the overall patterns of the average RMSE, MAPE, and R scores of each model architecture.

Looking at the performance scores in a holistic approach, the LSTM and GRU models are relatively close, but CNN shows no similar behavior to the LSTM and GRU. The LSTM outcome shows that the average RMSE and MAPE scores are low with 30 neurons. Furthermore, after 30 neurons, we can see the increasing trends until 200 and no significant decreasing trends afterward. Similarly, the most considerable value of the R score is seen at 30 neurons. The GRU model with 50 neurons provided the smallest average RMSE, MAPE, and the most significant average R score. The CNN model with 30 neurons has the smallest average RMSE and MAPE with the highest R score. From the comparative analysis, it can be concluded that 30 neurons LSTM, 50 neurons GRU, and 30 neurons CNN are the best in their respective categories. Table 4

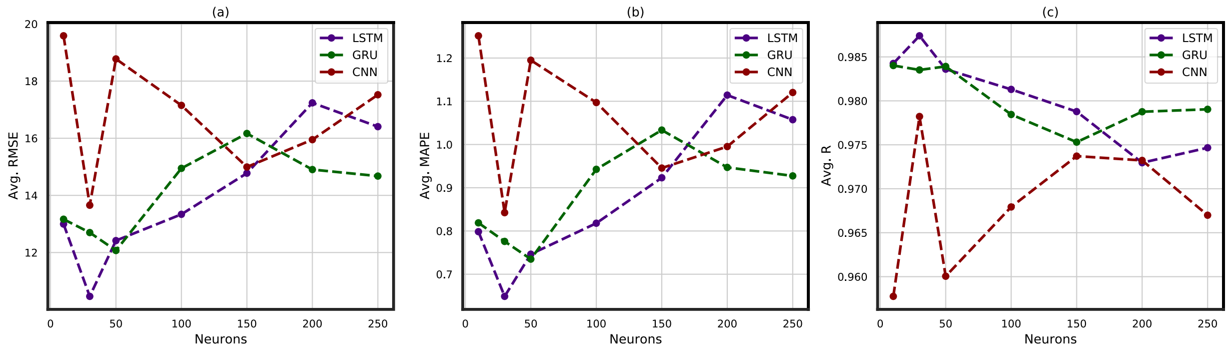


Fig. 7. Average scores obtained from LSTM, GRU, and CNN models: (a) RMSE, (b) MAPE, and (c) R on test dataset.

Table 3
List of the best hyperparameters for LSTM, GRU, and CNN models.

No. of Neurons/Filters	LSTM			GRU			CNN		
	Optimizer	Learning rate	Batch size	Optimizer	Learning rate	Batch size	Optimizer	Learning rate	Batch size
10	Adam	0.01	16	Adam	0.001	4	Adagrad	0.01	8
30	Adam	0.1	16	Adagrad	0.01	8	Adagrad	0.1	8
50	Adam	0.001	16	Adagrad	0.01	16	Adagrad	0.01	16
100	Adagrad	0.01	16	Adagrad	0.001	8	Adagrad	0.01	16
150	Adagrad	0.001	4	Adagrad	0.001	16	Adagrad	0.01	16
200	Adagrad	0.001	16	Adagrad	0.001	16	Adagrad	0.01	16
250	Adagrad	0.001	16	Adagrad	0.001	16	Adagrad	0.001	8

Table 4
Best performing LSTM, GRU, and CNN models with their best hyperparameters.

Models	No. of neurons	Optimizer	Learning rate	Batch size
LSTM	30	Adam	0.1	16
GRU	50	Adagrad	0.01	16
CNN	30	Adagrad	0.1	8

Table 5
The performance scores of the models in the test data.

Models	Metrics →	RMSE	MAPE	R
LSTM	Mean ± Std	**10.4660 ± 0.6836**	**0.6488 ± 0.0502**	**0.9874** ± **0.0009**
	Minimum	9.4338	0.5826	0.9854
	Maximum	12.0675	0.7574	0.9887
GRU	Mean ± Std	12.0706 ± 0.8996	0.7350 ± 0.0560	0.9839 ± 0.0012
	Minimum	10.8261	0.6527	0.9817
	Maximum	14.0081	0.8630	0.9859
CNN	Mean ± Std	13.6554 ± 1.1269	0.8424 ± 0.0711	0.9782 ± 0.0038
	Minimum	10.9845	0.7058	0.9697
	Maximum	15.3935	0.9589	0.9840

provides the list of the best performing models with their corresponding hyperparameters.

5.4. Identifying overall best model

After identifying the best-performing models from respective architectures, we further compare the performance scores to find the absolute best among these three models. Table 5 provides the statistics of the performance scores obtained from the three best-performing models. As we can see, the LSTM model with 30 neurons shows the smallest RMSE (10.4660) and MAPE (0.6488) with the highest R score (0.9874). Furthermore, the LSTM model has the lowest standard deviations in all the performance score metrics (0.6836 for RMSE, 0.0502 for MAPE, and 0.0009 for R). The superior performance of the LSTM model is further supported by Fig. 8 which provides the boxplots of the performance scores obtained from the three best-performing models. The overall distributions of the scores are approximately symmetric, which is evidence of consistent performance obtained from these models. The LSTM model has relatively better RMSE, MAPE, and R scores and has lower variability. Thus, Table 5 and Fig. 8 suggest that the LSTM model with 30 neurons is the ultimate winner, followed by GRU with 50 neurons and CNN with 30 neurons.

Fig. 9 and Fig. 10 provide the actual vs. predicted plots that gauge the goodness of fit to determine the quality of the prediction obtained on the training and the test data, respectively. Blue dots represent actual versus predicted closing price values for the training and test data set. Furthermore, the red dotted line shows each plot's best-fit line ($y = x$). The overall fit of the training data is almost indistinguishable in all three subplots. However, there is a slight variation after 1600, where all models have a slight deviation in both directions. The LSTM has relatively better performance. In the test data, the predicted closing price deviates more from the actual closing price compared to the train data, which is as expected. Among the subplots (a), (b), and (c) in Fig. 10, the LSTM shows a superior fit compared to GRU and CNN.

Fig. 11 represents the actual time series together with the predicted closing price obtained from the three best-performing models. The blue curve represents actual values, whereas the red one represents the predicted values. As shown in the subplots, each model underestimates the true value until June 2019 but becomes closer to the true value afterward. In subplot (a), the prediction curve obtained from the LSTM model captures the fluctuations more accurately in almost every situation. However, the GRU and CNN struggle to capture the actual values after January 2020, as shown in subplots (b) and (c). Overall, the plot demonstrates the superiority of the LSTM model in predicting the index price compared to other models.

5.5. Performing statistical analysis

To validate the reliability of the model's outcome, we conduct a statistical analysis to identify whether the performances of the three best models are significantly different from each other. First, we perform the normality test of RMSEs based on D'Agostino and Pearson (D'agostino & Pearson, 1973) which ensures that the RMSEs of the three models follow the normal distributions as p-values are significantly higher than the significance level $\alpha = 0.05$ as presented in Table 6. The Fig. 12 further validates the error produced by the employed architecture following the normal distribution via quantile–quantile (QQ) plots.

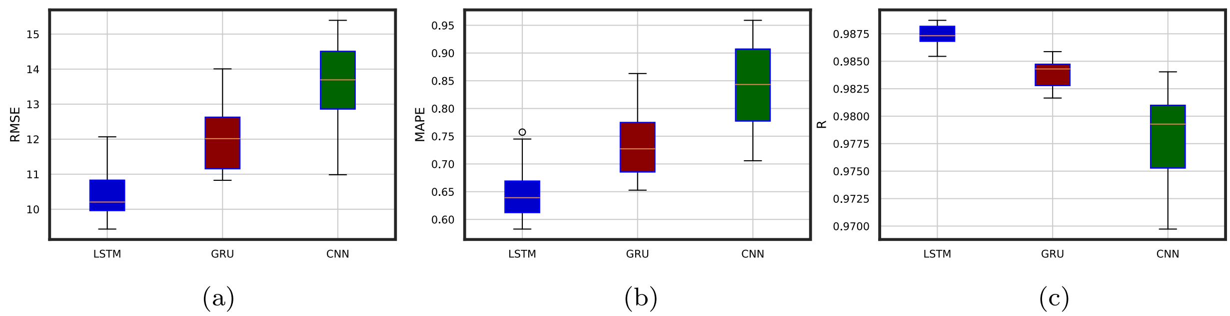


Fig. 8. Boxplots of metrics: (a) RMSE, (b) MAPE, and (c) R of the LSTM, GRU, and CNN models.

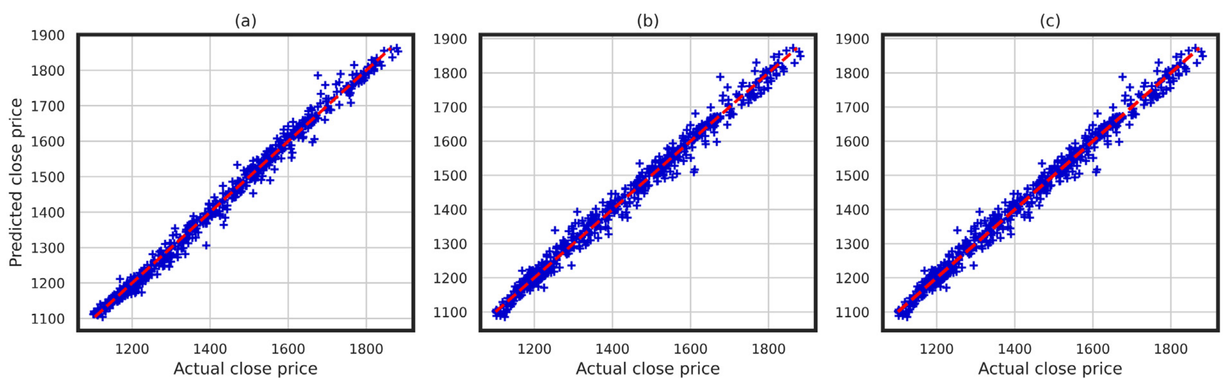


Fig. 9. True versus predicted plots of the models: (a) LSTM, (b) GRU, and (c) CNN on train data.

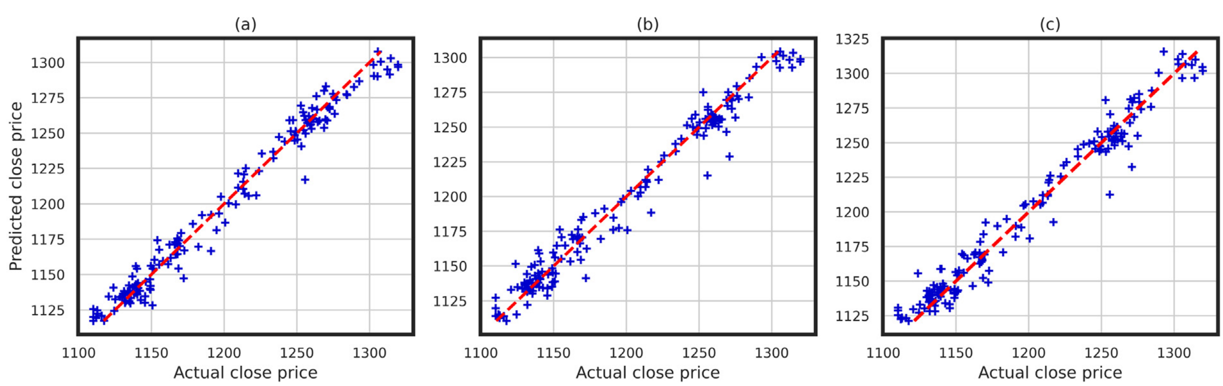


Fig. 10. True versus predicted plots of the models: (a) LSTM, (b) GRU, and (c) CNN on test data.

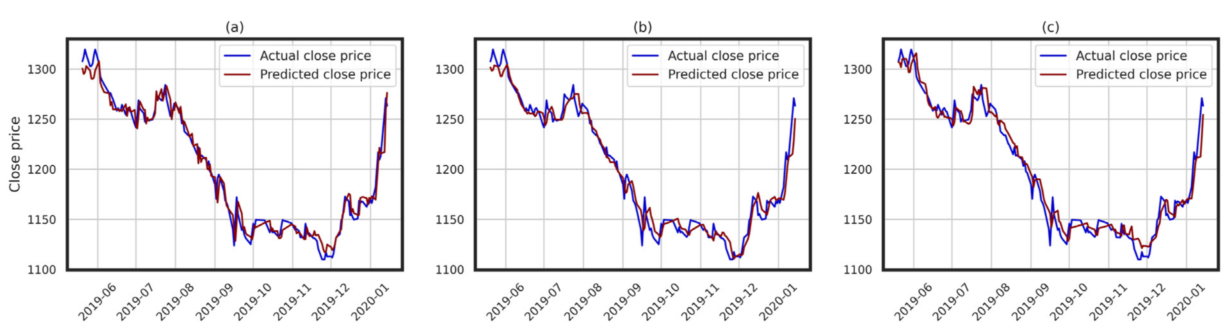


Fig. 11. Time series plots of the true and predicted values obtained from three models: (a) LSTM, (b) GRU, and (c) CNN on test dataset.

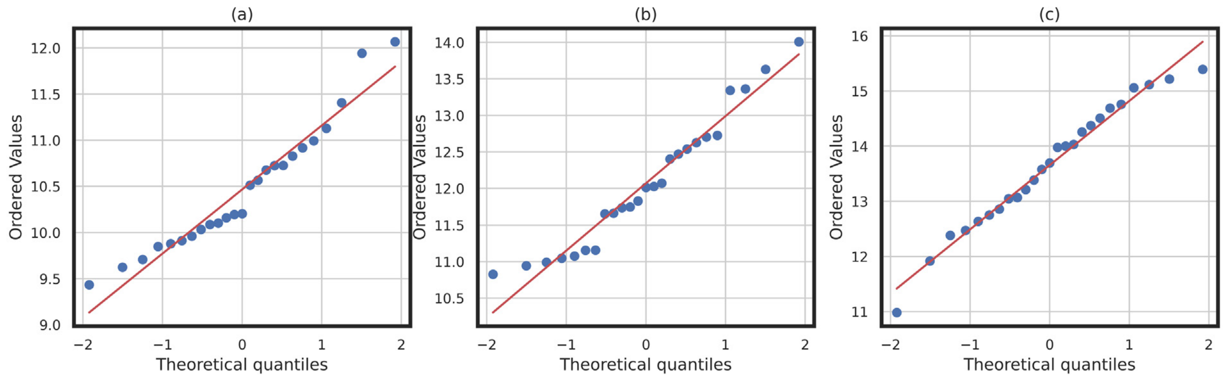


Fig. 12. Normality plots of RMSEs of the LSTM, GRU, and CNN models.

Table 6
The test statistics and P-values from normality test of RMSEs of the models performance.

	LSTM	GRU	CNN
Test statistics	3.2882	1.5082	0.7701
P-value	0.1932	0.4704	0.6804

Table 7
The test statistics and P-values from two samples t-test for pairwise comparison of model performance.

	(LSTM, GRU)	(LSTM, CNN)	(GRU, CNN)
Test statistics	−7.1010	−12.0992	−5.4951
P-value	7.4038×10^{-9}	7.1701×10^{-15}	1.6655×10^{-6}

After conforming the normality of the distributions of RMSEs, we perform the two-sample t-test between the distributions of RMSEs. The test statistics and p-values from the tests are listed in Table 7. There is a significant difference between the means of RMSEs of pairs (LSTM, GRU), (LSTM, CNN), and (GRU, CNN). The stated pairwise model's comparison produces the outcome in favor of the LSTM model. Hence, we conclude that the LSTM model with 30 neurons best serves to predict the closing price of NEPSE.

6. Ethics and implications

High ethical standards, transparency, integrity, and candor are critical attributes to maintaining investors' trust in the financial market. The study uses a publicly available dataset without manipulation at all. Also, the reported performance of the model is an average performance on out-of-sample data based on several replications. Thus, one can use the results as an additional piece of information to make an investment decision that upholds investors' confidence. However, the investment decision should not rely entirely on the research outcome. Investors are expected to perform their due diligence and consider their risk tolerance in various market conditions. The reasonable forecast depends not only on the outcome of the specific model but also on the volatile nature of the stock market, especially during geopolitical tension, the global supply chain disturbance, war, pandemic, and various other market risks. Thus, the stakeholders can benefit if the market's current behavior is appropriately analyzed and amalgamated with the model's outcome.

The equity traders, individual investors, and portfolio managers intrinsically want to predict the stock price with the projected return. This research shows the promising possibility of neural network architecture to delineate the cone of uncertainty in stock price prediction. Moreover, academic researchers can build the proposed model framework to expand the horizon of the field in sequential data modeling.

7. Conclusion

Predicting stock price is of high interest for finance practitioners to best allocate their assets and academics to build an optimal model for consistent predictions with a high level of accuracy. Stock prediction is a challenging task due to its noisy and nonlinear behavior. Multifaceted factors, both local and global, may affect the prediction directly or indirectly. This study builds the predictive models using sixteen predictors that fall under the fundamental, macroeconomic, technical, and financial news data. A comparative analysis of the NEPSE index price prediction is performed using the deep learning architectures, namely LSTM, GRU, and CNN. An extensive data-driven approach has been implemented to optimize the model hyperparameters. The performance of the model is evaluated using RMSE, MAPE, and R. The experimental results show that the LSTM model with 30 neurons provides a superior fit and high prediction accuracy, followed by GRU with 50 neurons and CNN with 30 neurons. The outcome is further validated by statistical analysis of the performance metrics. The proposed model can be tailored across other broad market indices where the data show similar characteristics.

In the near future, we plan to develop hybrid predictive models by combining the implemented models with some other neural network architectures such as transformers. Another potential direction is the amalgamation of classical and deep learning model architecture to build a new predictive model. We also plan to implement a hybrid optimization algorithm that trains model parameters by combining local and global optimizers. In addition to the financial news, other media sentiments can be included in the model development process. Finally, the implementation of evolutionary algorithms to achieve a state-of-the-art performance is also a matter of future work.

CRediT authorship contribution statement

Nawa Raj Pokhrel: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Keshab Raj Dahal:** Conceptualization, Methodology, Validation, Visualization, Investigation, Writing – original draft, Writing – review & editing. **Ramchandra Rimal:** Methodology, Validation, Visualization, Investigation, Writing – original draft, Writing – review & editing. **Hum Nath Bhandari:** Methodology, Software, Writing – original draft, Writing – review & editing. **Rajendra K.C. Khatri:** Visualization, Writing – review & editing. **Binod Rimal:** Methodology, Software, Visualization, Writing – review & editing. **William Edward Hahn:** Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would like to thank the Association of Nepalese Mathematicians in America for creating a collaborative research opportunity that resulted in this work. We are also thankful to Google LLC for providing GPU supported open-source cloud computing platform, Google Colab.

References

Acharya, S., & Koirala, J. (2021). Legal aspect of money market and capital market in Nepal. Available At SSRN 3835358.

Adhikari, D. (2018). Impact of exchange rate on trade deficit and foreign exchange reserve in Nepal: An empirical analysis. *NRB Economic Review*, *30*(1), 36–48.

Agarap, A. F. M. (2018). A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data. In *Proceedings of the 2018 10th International conference on machine learning and computing* (pp. 26–30).

Ahangar, R. G., Yahyazadehfar, M., & Pournaghshband, H. (2010). The comparison of methods artificial neural network with linear regression using specific variables for prediction stock price in tehran stock exchange. URL.

Ahmed, J., & Martinez-Zarzoso, I. (2013). Blessing or curse: The stabilizing role of remittance, foreign aid and FDI to Pakistan. Foreign Aid and FDI to Pakistan (May 7, 2013).

Althelaya, K. A., El-Alfy, E.-S. M., & Mohammed, S. (2018). Evaluation of bidirectional LSTM for short-and long-term stock market prediction. In *2018 9th International conference on information and communication systems* (pp. 151–156). IEEE.

Anghel, G. D. I. (2015). Stock market efficiency and the MACD. Evidence from countries around the world. *Procedia Economics and Finance*, *32*, 1414–1431.

Bank, N. R. (2017). Optimal inflation rate for nepal. *Economic Review*, *29*, 1–21.

Barnett, W. A., & Serletis, A. (2000). Martingales, nonlinearity, and chaos. *Journal of Economic Dynamics & Control*, *24*(5–7), 703–724.

Bhandari, H. N., Rimal, B., Pokhrel, N. R., Rimal, R., Dahal, K. R., & Khatri, R. K. (2022). Predicting stock market index using LSTM. *Machine Learning with Applications*, *9*, Article 100320.

Bhusal, M. K. (2017). Application of Markov chain model in the stock market trend analysis of nepal. *International Journal of Scientific & Engineering Research*, *8*(10), 1733–1745.

Bhusal, T. P., & Silpakar, S. (2011). Growth and inflation: Estimation of threshold point for nepal. *Economic Journal of Development Issues*, *13*, 131–138.

Brownlee, J. (2018a). *Better deep learning: Train faster, reduce overfitting, and make better predictions*. Machine Learning Mastery.

Brownlee, J. (2018b). *Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery.

Brownlee, J. (2018c). How to develop convolutional neural network models for time series forecasting.

Chaudhary, S. K., & Xiumin, L. (2018). Analysis of the determinants of inflation in Nepal. *American Journal of Economics*, *8*(5), 209–212.

Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE International conference on big data (Big Data)* (pp. 2823–2824). IEEE.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Chollet, F. (2017). *Deep learning with Python*. Simon and Schuster.

Chong, T. T.-L., & Ng, W.-K. (2008). Technical analysis and the London stock exchange: testing the MACD and RSI rules using the FT30. *Applied Economics Letters*, *15*(14), 1111–1114.

Chong, T. T.-L., Ng, W.-K., & Liew, V. K.-S. (2014). Revisiting the performance of MACD and RSI oscillators. *Journal of Risk and Financial Management*, *7*(1), 1–12.

Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. arXiv preprint arXiv:1506.07503.

Coelho, J., D'almeida, D., Coyne, S., Gilkerson, N., Mills, K., & Madiraju, P. (2019). Social media and forecasting stock price change. In *2019 IEEE 43rd Annual computer software and applications conference, vol. 2* (pp. 195–200). IEEE.

D'agostino, R., & Pearson, E. S. (1973). Tests for departure from normality. Empirical results for the distributions of b^2 and $\sqrt{b}$. *Biometrika*, *60*(3), 613–622.

Danthine, J.-P. (1977). Martingale, market efficiency and commodity prices. *European Economic Review*, *10*(1), 1–17.

Devkota, M. L. (2018). The dynamic causality between stock prices and macroeconomic variables: evidence from nepal. *Annals of'Constantin Brancusi'University of Targu-Jiu. Economy Series*.

Devkota, M. L., & Panta, H. (2018). An inquiry into the effect of the interest rate, gold price, and the exchange rate on stock exchange index: Evidence from Nepal. *Dynamic Econometric Models*, *18*, 49–65.

Di Persio, L., & Honchar, O. (2016). Artificial neural networks architectures for stock price prediction: Comparisons and applications. *International Journal of Circuits, Systems and Signal Processing*, *10*(2016), 403–413.

Eric, D., Andjelic, G., & Redzepagic, S. (2009). Application of MACD and RVI indicators as functions of investment strategy optimization on the financial market. *Zbornik Radova Ekonomskog Fakulteta U Rijeci: ČAsopis Za Ekonomsku Teoriju I Praksu*, *27*(1), 171–196.

Fama, E. F., & French, K. R. (1988). Permanent and temporary components of stock prices. *Journal of Political Economy*, *96*(2), 246–273.

Fukushima, K., & Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets* (pp. 267–285). Springer.

Géron, A. (2019). *Hands-on machine learning with scikit-learn, keras, and tensorflow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471.

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2003). Learning precise timing with lstm recurrent networks. *Journal of Machine Learning Research*, [ISSN: 1532-4435] *3*(null), 115–143. http://dx.doi.org/10.1162/153244303768966139.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232. http://dx.doi.org/10.1109/TNNLS.2016.2582924.

Gurung, J. B. (2004). Growth and performance of securities market in Nepal. *Journal of Nepalese Business Studies*, *1*(1), 85–92.

Gurung, R. (2019). Relationship between treasury bill rate and NEPSE index in Nepal. *Pravaha*, *25*(1), 159–166.

Hansen, J. V., McDonald, J. B., & Nelson, R. D. (1999). Time series prediction with Genetic-Algorithm designed neural networks: An empirical comparison with modern statistical models. *Computational Intelligence*, *15*(3), 171–184.

Hill, T., O'Connor, M., & Remus, W. (1996). Neural network models for time series forecasts. *Management Science*, *42*(7), 1082–1092.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02), 107–116.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, *129*, 273–285.

Hubel, D. H. (1959). Single unit activity in striate cortex of unrestrained cats. *The Journal of Physiology*, *147*(2), 226–238.

Hutto, C., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*.

Jiawei, X., & Murata, T. (2019). Stock market trend prediction with sentiment analysis based on lstm neural network. In *International multiconference of engineers and computer scientists* (pp. 475–9).

Ju, X., Chen, V. C., Rosenberger, J. M., & Liu, F. (2021). Fast knot optimization for multivariate adaptive regression splines using hill climbing methods. *Expert Systems with Applications*, *171*, Article 114565.

Ju, X., Rosenberger, J. M., Chen, V. C., & Liu, F. (2022). Global optimization on non-convex two-way interaction truncated linear multivariate adaptive regression splines using mixed integer quadratic programming. *Information Sciences*, *597*, 38–52.

Kara, Y., Boyacioglu, M. A., & Baykan, O. K. (2011). 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*, *38*(5), 5311–5319.

Katuwal, N. (2021). Contribution of remittance to rural economy in nepal. *International Advanced Research Journal in Science, Engineering and Technology (IARJSET)*, *7*(8), 13–18.

Khatri, M. B. (2019). Macroeconomic influence on the nepalese stock market. *NRB Economic Review*, *31*(1), 47–64.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, *2*.

Lei, J., Liu, C., & Jiang, D. (2019). Fault diagnosis of wind turbine based on long short-term memory networks. *Renewable Energy*, [ISSN: 0960-1481] *133*, 422–432. http://dx.doi.org/10.1016/j.renene.2018.10.031, URL https://www.sciencedirect.com/science/article/pii/S0960148118312151.

Li, J., Bu, H., & Wu, J. (2017). Sentiment-aware stock market prediction: A deep learning method. In *2017 International conference on service systems and service management* (pp. 1–6). IEEE.

Murphy, J. J. (1999). *Technical analysis of the financial markets: a comprehensive guide to trading methods and applications*. Penguin.

Nelson, D. M., Pereira, A. C., & De Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. In *2017 International joint conference on neural networks* (pp. 1419–1426). IEEE.

Neupane, B. P. (2010). *Stock market efficiency: a case of nepal* (Ph.D. thesis), Tribhuvan University.

Neupane, S. (2018). Impact of interest rate on stock market in nepal. (Ph.D. thesis), Central Department of Management.

Ojha, K. P. (2019). Remittance status and contribution to GDP of Nepal. *NCC Journal*, *4*(1), 101–112.
Pant, B., & Budha, B. B. (2016). Remittances and exchange rate linkages: Experiences of Nepal. *Nepal Rastra Bank Working Paper*, 33.
Panta, B. P. (2020). Macroeconomic determinants of stock market prices in Nepal. *Panta, B.(2020). Macroeconomic Determinants of Stock Market Prices in Nepal. Quest Journal of Management and Social Sciences*, *2*(1), 64–79.
Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259–268.
Paudel, P. K., Bhattarai, B. P., & Kindlmann, P. (2012). An overview of the biodiversity in nepal. *Himalayan Biodiversity in the Changing World*, 1–40.
Paudel, R. C., & Burke, P. J. (2015). Exchange rate policy and export performance in a landlocked developing country: The case of Nepal. *Journal of Asian Economics*, *38*, 55–63.
Poterba, J. M., & Summers, L. H. (1988). Mean reversion in stock prices: Evidence and implications. *Journal of Financial Economics*, *22*(1), 27–59.
Pradhan, R. S. (1993). Stock market behavior in a small capital market: A case of Nepal. *Pradhan, RS (1993), Stock Market Behavior in A Small Capital Market: A Case of Nepal, Nepalese Management Review, Vol. Nn*, 20–32.
Pun, T. B., & Shahi, T. B. (2018). Nepal stock exchange prediction using support vector regression and neural networks. In *2018 Second international conference on advances in electronics, computers and communications* (pp. 1–6). IEEE.
Qiu, M., & Song, Y. (2016). Predicting the direction of stock market index movement using an optimized artificial neural network model. *PLoS One*, *11*(5), Article e0155133.
Qiu, J., Wang, B., & Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLoS One*, *15*(1), 1–15. http://dx.doi.org/10.1371/journal.pone.0227222.
Rakhal, R. (2018). Determinants of stock market performance. *NCC Journal*, *3*(1), 134–142.
Rodríguez-González, A., García-Crespo, A., Colomo-Palacios, R., Iglesias, F. G., & Gómez-Berbís, J. M. (2011). CAST: Using neural networks to improve trading systems based on technical analysis by means of the RSI financial indicator. *Expert Systems with Applications*, *38*(9), 11489–11500.
Salahuddin, M., & Gow, J. (2015). The relationship between economic growth and remittances in the presence of cross-sectional dependence. *The Journal of Developing Areas*, 207–221.
Samarawickrama, A., & Fernando, T. (2017). A recurrent neural network approach in predicting daily stock prices an application to the Sri Lankan stock market. In *2017 IEEE International conference on industrial and information systems* (pp. 1–6). IEEE.
Saud, A. S., & Shakya, S. (2019). Analysis of gradient descent optimization techniques with gated recurrent unit for stock price prediction: A case study on banking sector of Nepal stock exchange. *Journal of Institute of Science and Technology*, *24*(2), 17–21.
Saud, A. S., & Shakya, S. (2020). Analysis of look back period for stock price prediction with RNN variants: A case study on banking sector of NEPSE. *Procedia Computer Science*, *167*, 788–798.
Seng, J.-L., & Yang, H.-F. (2017). The association between stock price volatility and financial news–a sentiment analysis approach. *Kybernetes*.
Shahi, T. B., Shrestha, A., Neupane, A., & Guo, W. (2020). Stock price forecasting with deep learning: A comparative study. *Mathematics*, *8*(9), 1441.
Shen, G., Tan, Q., Zhang, H., Zeng, P., & Xu, J. (2018). Deep learning with gated recurrent unit networks for financial sequence predictions. *Procedia Computer Science*, *131*, 895–903.
Shrestha, P. K., & Pokhrel, S. (2019). Factors affecting stock index in Nepal. *Sebon Journal*.
Thapa, K. B. (2019). Influencing factors of stock price in Nepal. *NCC Journal*, *4*(1), 113–120.
Timsina, N. (2017). Determinants of bank lending in Nepal. *NRB Economic Review*, *20*, 22–42.
Wang, J., & Kim, J. (2018). Predicting stock price trend using MACD optimized by historical volatility. *Mathematical Problems in Engineering*, *2018*.
Wang, J.-H., & Leu, J.-Y. (1996). Stock market trend prediction using ARIMA-based neural networks. In *Proceedings of international conference on neural networks, vol. 4* (pp. 2160–2165). IEEE.
Wang, Y., & Wang, Y. (2016). Using social media mining technology to assist in price prediction of stock market. In *2016 IEEE International conference on big data analysis* (pp. 1–4). IEEE.
Wei, C. (2020). Estimation for partially observed mean-reversion type stochastic systems. *Engineering Letters*, *28*(1).
Wen, T.-H., Gasic, M., Mrksic, N., Su, P.-H., Vandyke, D., & Young, S. (2015). Semantically conditioned lstm-based natural language generation for spoken dialogue systems. arXiv preprint arXiv:1508.01745.
Wilder, J. W. (1978). *New concepts in technical trading systems*. Trend Research.
Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the north american chapter of the association for computational linguistics: human language technologies* (pp. 1480–1489).
Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. arXiv preprint arXiv:2106.11342.
Zhao, T., Zhao, J., Yang, D., & Wang, H. (2013). Generalized martingale model of the uncertainty evolution of streamflow forecasts. *Advances in Water Resources*, *57*, 41–51.