



Stock Market Index Prediction Using Machine Learning and Deep Learning Techniques

Abdus Saboor^{1,4}, Arif Hussain², Bless Lord Y. Agbley³, Amin ul Haq^{3,*}, Jian Ping Li³ and Rajesh Kumar^{1,*}

¹Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Huzhou, 313001, China

²Abdul Wali Khan University Mardan, Mardan, 23200, Pakistan

³School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China

⁴Brain Institute Peshawar, Peshawar, 25130, Pakistan

*Corresponding Authors: Amin ul Haq. Email: khan.amin50@yahoo.com; Rajesh Kumar. Email: rajakumarlohano@gmail.com

Received: 31 December 2022; Accepted: 24 February 2023; Published: 23 June 2023

Abstract: Stock market forecasting has drawn interest from both economists and computer scientists as a classic yet difficult topic. With the objective of constructing an effective prediction model, both linear and machine learning tools have been investigated for the past couple of decades. In recent years, recurrent neural networks (RNNs) have been observed to perform well on tasks involving sequence-based data in many research domains. With this motivation, we investigated the performance of long-short term memory (LSTM) and gated recurrent units (GRU) and their combination with the attention mechanism; LSTM + Attention, GRU + Attention, and LSTM + GRU + Attention. The methods were evaluated with stock data from three different stock indices: the KSE 100 index, the DSE 30 index, and the BSE Sensex. The results were compared to other machine learning models such as support vector regression, random forest, and k-nearest neighbor. The best results for the three datasets were obtained by the RNN-based models combined with the attention mechanism. The performances of the RNN and attention-based models are higher and would be more effective for applications in the business industry.

Keywords: Machine learning; deep learning; stock market; prediction; data analysis

1 Introduction

Prediction of stock market performance in terms of stock returns, closing prices, and volume traded is the most difficult activity in economic time series analysis. It is due to the unpredictable, dynamic, and nonlinear nature of stock market activities and data. The political conditions, global



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

economic shocks, and firm financial performance add to the dynamic nature of stock market activities. Prediction of stock market returns is highly beneficial for investment decisions. Various techniques were adopted in the past few years to predict movements in stock prices. These include technical analysis and fundamental analysis methods. Trend patterns and charts are used in technical analysis based on historical price data for the identification of hidden points that may be used by investors for decision-making [1].

The fundamental analysis includes estimations based on investor moods and perceptions, financial and political news, and events that may influence the market. In recent times, advanced intelligent techniques have been used to forecast stock market performance in order to deal with nonlinear and huge-sized data. In this respect, machine learning techniques are found to be more efficient as compared to traditional methods of stock market predictions and performance analysis [2].

In addition, forecasting market returns in the stock exchange are challenging due to its nonlinear and volatile nature. Various classical algorithms and machine learning techniques are used for technical and fundamental analysis to predict market returns.

Hu et al. [3] structured a hybrid attention network (HAN) on the basis of related news patterns for predicting various trends in stock prices. Li et al. [4] are of the opinion that traditional quantitative models use time series data for predicting stock returns but these models cannot incorporate investor sentiments in such predictions. He suggested that predictions in stock returns can further be augmented by using time series along with deep learning.

The study proposed using a convolutional neural network for extracting emotional information while replacing the basic emotional characteristics of the emotional extraction level. Results suggested that this algorithm is effective and feasible for predicting variations in stock indices.

Zhong et al. [5] used artificial neural networks to predict trends in the S and P 500 index. Various dimensionality reduction techniques were used for streamlining the data set including fuzzy Robust Principal Component Analysis (FPCA), PCA, and Kernel-based Principal Component Analysis (KPCA). It was asserted that selecting an appropriate kernel function affects KPCA performance while combining PCA with Artificial Neural Networks (ANN).

Guresen et al. [6] made predictions of stock prices in NASDAQ by making a comparison among three Artificial Neural Networks (ANN) models. These ANN included dynamic artificial neural networks (DAN2), Multilayer perception (MLP), and autoregressive conditional heteroscedasticity (GARCH). A comparison of these three models was made using mean square error (MSE) and mean absolute deviation (MAD). It was concluded that the MLP method outperformed GARCH and DAN2. Their study further recommended investigating whether GARCH has remedying impact on a stock price forecast or whether other correlated variables that have such remedial effects on a stock price forecast.

Bing et al. [7] adopted MLP and Generalized Feed Forward (GEF) to predict the market index of the Istanbul Stock Exchange. Their study made predictions using artificial neural networks and moving averages and changes the number of hidden layers. The coefficient of determination was used to measure the accuracy of predictions while the highest accuracy was achieved by using one hidden layer for both GEF and MLP.

Rajab et al. [8] used a hybrid fuzzy logic approach for analyzing sentiments given on social media and predicting stock indices.

Sedighi et al. [9] further adopted a novel model for predicting various stock indices using ANFIS, SVM, and Artificial Bee Colony (ABC). Their study used twenty technical indices for fifty US companies as input over a period of 2008 to 2018. Quality and accuracy are taken as criteria for performance measures and concluded that this model has higher forecasting accuracy compared to others.

Deep learning techniques have become increasingly popular for stock market analysis due to their ability to capture complex non-linear relationships within large amounts of data. They can be used to identify patterns, trends, and dependencies in historical stock market data to make predictions about future stock prices. This is achieved through the use of artificial neural networks, which are capable of learning and generalizing from large amounts of data to make predictions. One of the key advantages of using deep learning techniques for stock market analysis is their ability to handle high dimensional data, such as financial data which often includes multiple features such as stock prices, trading volume, and news articles. This allows deep learning models to capture not only the historical patterns in the data but also the relationships between different features that can impact stock prices. Another advantage is the ability of deep learning models to handle time-series data, where stock prices can be considered as a sequence of values over time. This is particularly useful for stock market prediction, where historical trends and patterns in the data can be used to make predictions about future trends. Recurrent Neural Networks (RNNs) Hushani [10], such as LSTM and GRU, are particularly suited for time-series data and have been widely used for stock market prediction.

The stock market's accurate index prediction is significantly necessary. In this work, we proposed using a recurrent neural network (RNN) to predict close prices for three stock indices. The RNNs have been observed to perform well on tasks involving sequence-based data in many research domains. With this motivation, we investigated the performance of long-short term memory (LSTM) and gated recurrent units (GRU) and their combination with stock data from three different stock indices: the KSE 100 index, the DSE 30 index, and the BSE Sensex. The results were compared to other machine learning models such as support vector regression, random forest, and k-nearest neighbor. The best results for the three datasets were obtained by the RNN-based models. The performances of the RNN-based models are higher and the proposed method could be easily incorporated into stock market index prediction.

We thus summarized our contributions in the following points:

- i) We proposed applying three AI methods based on RNN for the task of close price predicting of three stock market indices. We also enhanced the models with the attention mechanism.
- ii) Other machine learning techniques, including support vector regressor, random forest, and k-nearest neighbor, were also used to predict the closing prices.
- iii) The proposed RNN methods were compared with the machine learning techniques using metrics such as RMSE, explained variance, and mean gamma deviance, among others.
- iv) To measure the effect of the data range on the predictions, we conducted additional studies with a slice of the data, ignoring all instances before the year 2015.
- v) The results displayed in the tables and figures show that our proposed RNN and attention-based methods minimized errors better and, hence, obtained the best predictions.

This work is organized as follows: Introduction is provided in Section 1. In Section 2, we explored the three datasets used. The algorithms and metrics are also explained in this section. The experimental results are discussed in Section 3. Finally, the conclusion is given in Section 4.

2 Materials and Method

2.1 Data Exploration

Historical data for various stock indices for this research study is collected from a website [11]. Data for three South Asian countries are collected from their major stock exchange with a major representative stock index of the country. These countries and their major stock indices are represented in Table 2. Weekly stock indices data is extracted including the weekly opening price of a relevant stock index, the opening stock index price on the given date, the highest stock index price on the given date, the lowest stock index price on the given date, the volume of shares traded on the given date and weekly percentage change in the stock price index. The Stock Return (SR) in the last column of the table is calculated by subtracting the Previous week's price of the stock index (SI) from the current week's price and dividing by the current week's price, as shown in Eq. (1):

$$SR = \frac{Week_n SI - Week_{(n-1)} SI}{Week_{(n-1)} SI} \times 100 \quad (1)$$

Table 1: Month-wise mean of open and closed prices for BSE sensx

Date	Open	Close
January	29807.611207	29864.055517
February	27996.313542	27917.441458
March	27792.623019	27667.041321
April	27506.501961	27592.542549
May	27569.943273	27649.696364
June	27510.699273	27580.660182
July	27877.710526	27998.333333
August	28534.904310	28548.944138
September	28774.588214	28858.994107
October	29420.109474	29422.472807
November	29535.631071	29429.058929
December	29732.855614	29857.539649

Table 2: The countries and their respective stock index considered in this work

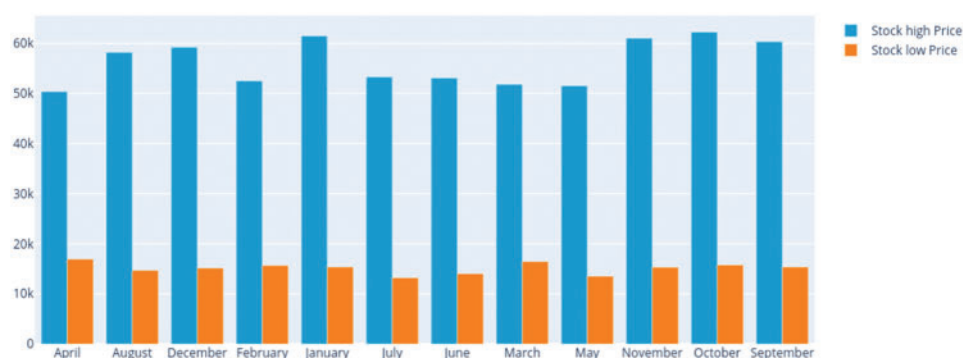
S. no.	Country	Stock exchange	Representative stock index
1	Pakistan	Karachi stock exchange	KSE 100 index
2	Bangladesh	Dhaka stock exchange	DSE 30
3	India	Bombay stock exchange	BSE sensx

2.1.1 Bombay Stock Exchange (BSE)

This data was gathered for the period 2009-05-31 to 2022-01-23, making a total of 4620 days. The mean for each month of this period is shown in Table 1. The lowest monthly mean for the open price is 27,506.50, recorded in April. The highest open price means is 29,807.61 recorded in January. Likewise, the lowest mean closing price is 27,998.33, recorded in June, while the highest mean closing price is 29,864.05, recorded in January.

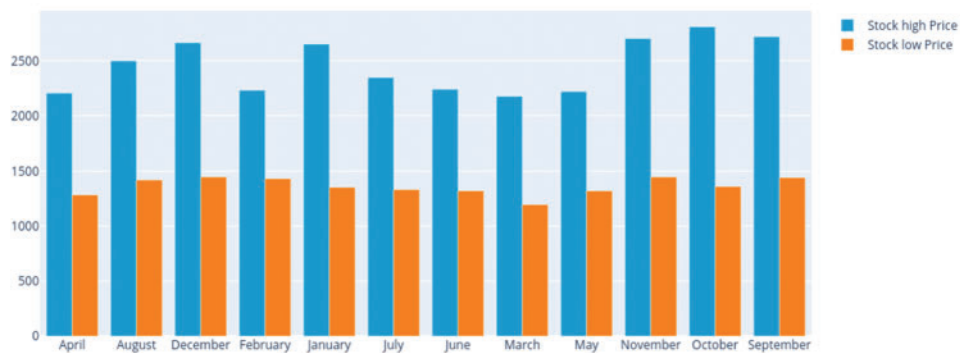
The low and high stock prices for BSE are also explored to identify the highest and lowest recordings based on the month, as shown in Fig. 1a. For the high stock prices, the highest value of 61,475.15 was recorded in January, and the lowest value of 50372.23 in April, as shown by the figure.

BSE: Monthwise High and Low stock price



(a)

Dhaka: Monthwise High and Low stock price



(b)

Figure 1: (Continued)

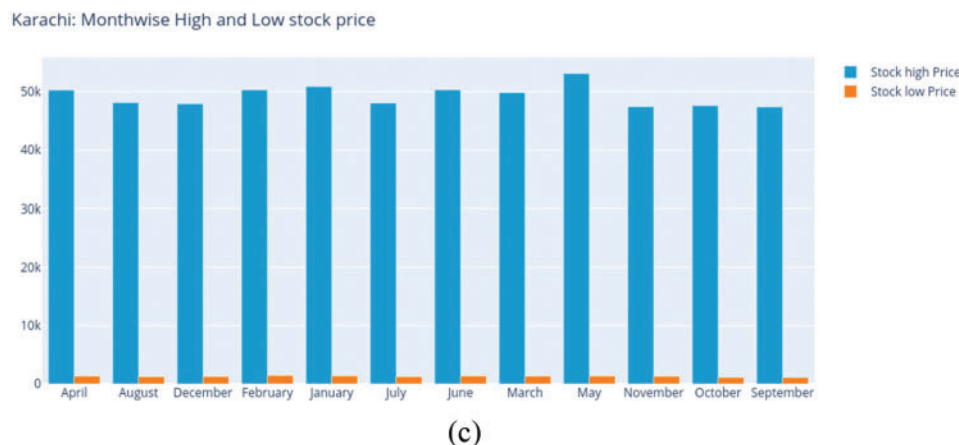


Figure 1: Month-wise high and low prices for (a) BSE, (b) DSE, and (c) KSE

2.1.2 Dhaka Stock Exchange (DSE)

The data collected for DSE covers the period 2013-02-03 to 2022-01-23, making a total of 3276 days. The mean for each month of this period is shown in Table 3. The lowest monthly mean for the open price is 1,766.88, recorded in the month of March. The highest open price means is 1,954.16 recorded in January. Likewise, the lowest mean closing price is 1,759.80, recorded in March, while the highest mean closing price is 1,962.49, recorded in January. The low and high stock prices for DSE are also explored to identify the highest and lowest recordings based on the month, as shown in Fig. 1b. For the high stock prices, the highest value of 2,807.80 was recorded in October, and the lowest value of 2174.98 in March, as shown by the figure.

Table 3: Month-wise mean of open and closed prices for DSE 30

Date	Open	Close
January	1954.164359	1962.486154
February	1844.338889	1841.792500
March	1766.884146	1759.788537
April	1795.665556	1799.210278
May	1768.026842	1770.220000
June	1781.358462	1781.942564
July	1825.288108	1830.443243
August	1858.223750	1869.817500
September	1900.312051	1904.150769
October	1930.242432	1926.463514
November	1871.264872	1873.670000
December	1869.292500	1871.652750

2.1.3 Karachi Stock Exchange (KSE)

The data collected for KSE begins from 2000-01-09 and ends at 2022-01-16 accounting for 8043 days. Table 4 contains the averages of open and close prices for the months. The lowest month-wise means for the open and close prices are 19,180.56 and 19,121.15 respectively, observed in March. However, the highest mean open price, 20,195.80 is observed in December, while the highest mean close price, 20,329.80 is observed in January regarding the stock prices for KSE, see Fig. 1c. The largest occurred high price is 53,127.24 in the month of May, while the lowest is 47,387.34, occurring in September. Also, the low price is highest in February at 1,415.51 and lowest in September at 1,069.55.

Table 4: Month-wise mean of open and closed prices for the KSE 100 index

Date	Open	Close
January	20190.698283	20329.797475
February	19270.073596	19212.736180
March	19180.564227	19121.145979
April	19414.548421	19478.297474
May	19632.227526	19642.608454
June	19420.506170	19406.781277
July	19547.084898	19589.522347
August	19726.102887	19619.107835
September	19604.523298	19624.731277
October	19703.748125	19794.886875
November	20131.769255	20174.665638
December	20195.796224	20314.478367

2.2 Algorithms

We investigated the performance of several machine learning algorithms with each of the different sources of data. Aside from the typical ML methods, we also used two deep-learning methods popularly used on sequential data; LSTM and GRU, a combination of these two methods, and their attention-based enhancements. We describe the algorithms as follows:

2.2.1 Support Vector Regressor (SVR)

An approach for supervised learning called support vector regression is used for predicting discrete values. The SVMs and Support Vector Regression both operate on the same theory. Finding the optimum fit line is the fundamental tenet of SVR. The hyperplane with the most points is the best-fitting line in SVR. The SVR seeks to match the best line within a threshold value, in contrast to other regression models that aim to reduce the error between the true and their predictions. The distance between the boundary line and the hyperplane is the threshold value. SVR is difficult to scale to datasets with more than a few ten thousand samples since the fit time complexity is more than quadratic with the number of samples. In our implementation, the radial basis function was used as the kernel. The gamma value is set to 0.1, and the regularization parameter, C is set to 1e2.

2.2.2 *Random Forest Regressor (RF)*

Decision trees (DT) are one of the most popular classifiers used in machine learning. A decision tree is a decision-making tool that employs a tree-like framework of choices and their possible results, such as chance event outcomes, resource costs, and utility. One way to approach it is to illustrate an algorithm with conditional control statements. DTs are usually designed as flow diagrams made up of the root nodes, the internal nodes, and the leaf nodes. A decision tree begins with a root node that is not formed by incoming branches. The branches from the root nodes are fed to the internal nodes for decision-making. These nodes evaluate the given features to generate homogeneous subsets that are indicated by leaf nodes or end nodes. The end nodes represent all the dataset's conceivable outcomes. Random forest is simply an ensemble of decision trees, and they can either be used for classification; where the output is the class selected by most of the trees, or for regression problems, where the output is the mean of the results of the decision trees.

Scikit Learn library was used for our implementation. The random forest contains a total of 100 decision trees with the "squared error" criterion. The maximum depth of the trees is set to "none" which means that the trees will continue to grow until all the leaves are pure or until the leaves contain less than "min samples split" samples. The number of features to be considered for splitting a node is determined by the parameter "max features" set to "auto". The minimum number of samples required to split an internal node and the minimum number of samples required to be a leaf node are set to 2 and 1 respectively. The implementation also has a random state of 0 and is not set to use out-of-bag samples.

2.2.3 *K-Nearest Neighbour (K-NN)*

K-NN is a supervised learning technique that is non-parametric. It can be used for either regression or classification tasks. The k closest training samples of the data are fed as input and the result when performing classification is a class membership. An object is allocated to the class most frequently chosen by its k closest neighbors based on a majority vote among the item's neighbors. When using k-NN for regression, the output is the object's property value, which is obtained by taking the average of the values of the k closest neighbors. The k value in our experiments for the three datasets was set to 15.

2.3 *RNN-Based Models*

A recurrent neural network (RNN) is a type of artificial neural network in which node connectivity can form a loop, enabling outputs from one node to influence future inputs to the very same node. This enables it to display temporal dynamic characteristics. RNNs, which are based on feedforward neural networks, can analyze input sequences of variable length using the internal state (memory). As a result, they can be used for tasks like handwriting or speech recognition. RNNs can be distinguished from convolutional neural networks (CNNs) in that RNNs are networked with infinite impulse responses, while CNNs are networking with finite impulse responses. Both types of networks display temporal dynamics.

An infinite impulse recurrent network is a directed cyclic graph that cannot be unrolled and substituted with a complete feedforward neural network, whereas a finite impulse recurrent network can be unrolled. The inputs and outputs of RNNs can vary in length so that configurations such as one-to-one, one-to-many, many-to-one, and many-to-many are possible. Like in CNNs, activation functions such as Sigmoid, Tanh, and Relu can be used in RNNs. This paper experiments with two of the variants of RNN: long short-term memory (LSTM) and gated recurrent units (GRU).

2.3.1 Long-Short-Term Memory (LSTM)

LSTM [12–14] is proposed to extend the recurrent neural network in capturing information in both the long-term and short-term. Long-term memory can be compared to the way changes in synaptic strengths keep memories for the long term. In a similar fashion, for a network, the weights and biases are changed once for every training episode.

To keep short-term memory, the activations are updated every time step, which is also similar to the way short-term memory is stored in the brain as a result of moment-to-moment changes in the pattern of the electrical firings. We constructed the LSTM network with the Keras sequential module. First, an LSTM layer with 32 internal units is added, and we set the return sequences argument to true. The input shape argument was set to (15,1). The second LSTM layer is added, again with 32 internal units, and the return sequence is set to true. Finally, we added the last LSTM layer with 32 internal units whose output is transferred to a dense layer. We used the mean squared error as the loss function and Adam as the optimizer. Each of the datasets was split into train-test sets, and training was carried out for 200 epochs.

2.3.2 Gated Recurrent Unit (GRU)

Another common recurrent neural network method is the GRU [15]. The GRU is similar to the LSTM but has a forget gate and fewer parameters than the LSTM. The GRU is constructed in a similar configuration as the LSTM network. The first GRU layer consists of 32 internal units with the return sequences argument set to true and the input shape arguments set to (15, 1).

The second GRU layer also consists of 32 internal units with the return sequence argument set to true. Finally, the last LSTM layer consists of 32 internal units whose output is transferred to a dense layer. The mean squared error was used to compute the loss, and the Adam optimizer was used. The training set was trained for 200 epochs.

2.3.3 LSTM + GRU

A hybrid model involving LSTM and GRU was designed by stacking these two methods. We composed this model with two LSTM layers followed by two GRU layers and then a dense layer. In detail, the first LSTM consists of 32 internal units with the return sequences set to true and the input shape set to 15.

Another LSTM layer of 32 internal units was added. We again returned the sequences as input to the third recurrent layer. This third layer is a GRU with 35 internal units. The sequences are again returned to the last GRU layer with 32 internal units. The final layer is a dense layer. We used the mean squared error as our loss function and Adam as the optimizer. We also trained this model for 200 epochs.

2.4 Attention Mechanism

In recent years, the attention mechanism has gained popularity in various natural language processing tasks such as machine translation and text summarization. The attention mechanism allows the model to focus on certain parts of the input while disregarding irrelevant information.

In this paper, we explore the use of the attention mechanism in time series forecasting. The attention mechanism is incorporated into the RNN models described previously.

2.4.1 LSTM or GRU with Attention

We begin by reshaping the input to have the shape (samples, time steps, features) which is required for LSTM/GRU. The LSTM/GRU layer has 32 units and returns sequences to be used as input for the attention mechanism. The attention mechanism is implemented using the Dot product method, where the dot product of the query and key tensors is taken and used to calculate the attention weights. These attention weights are then multiplied with the value tensor to get the attention output. The attention output is concatenated with the LSTM/GRU output. This merged output becomes an input to a second LSTM/GRU layer with another 32 units. The attention is again implemented and the weights are concatenated with the output of the LSTM/GRU layer. A final LSTM/GRU layer with 32 units takes the results. Now the return sequences argument is set to false. The output from this last layer is passed to the dense layer for prediction.

2.4.2 LSTM-GRU with Attention

Finally, we combine the LSTM and GRU models with the attention mechanism to create an LSTM-GRU model. Similarly to the previous description, the input is reshaped to have the shape (samples, time steps, features) which is required for both LSTM and GRU. The first two layers of the model are the LSTM layers with 32 units each and the return sequences options set to True. The attention mechanism is implemented in the same way explained previously; using the Dot product method, where the dot product of the query and key tensors is taken and used to calculate the attention weights. These attention weights are then multiplied with the value tensor to get the attention output. The attention output is concatenated with the second LSTM output and a third and fourth layers are added which are GRU layers with 32 units each. Another attention is computed and its output is concatenated with the output of the GRU layer before forwarding to the dense layer for the prediction.

2.5 Steps

The overall architectures of the proposed RNN-based models are presented in [Fig. 2](#). For each of the networks, namely; LSTM, GRU, and LSTM + GRU, and their attention-based variations, the following steps were performed:

Step 1: The data corresponding to BSE, DSE, and KSE were explored, and the closing prices were reshaped into samples, time steps, and features as required by RNN-based models.

Step 2: The data is then passed through the networks. Three layers of LSTM layers are followed by a dense layer for the LSTM model. Likewise, for the GRU model, three layers of the GRU are stacked and followed by a dense layer. For the LSTM + GRU model, two GRU layers were stacked on two LSTM layers, followed by the dense layer. For the attention-based variations for LSTM and GRU, we added computed the attention twice. One after the first LSTM/GRU layer and again after the second LSTM/GRU layer. For the hybrid model, LSTM + GRU + Att, we computed the attention twice, the first after the two LSTM layers and the second after the two GRU layers.

Step 3: The predictions obtained from each of the models were used with the original closing prices to compute the evaluation metrics.

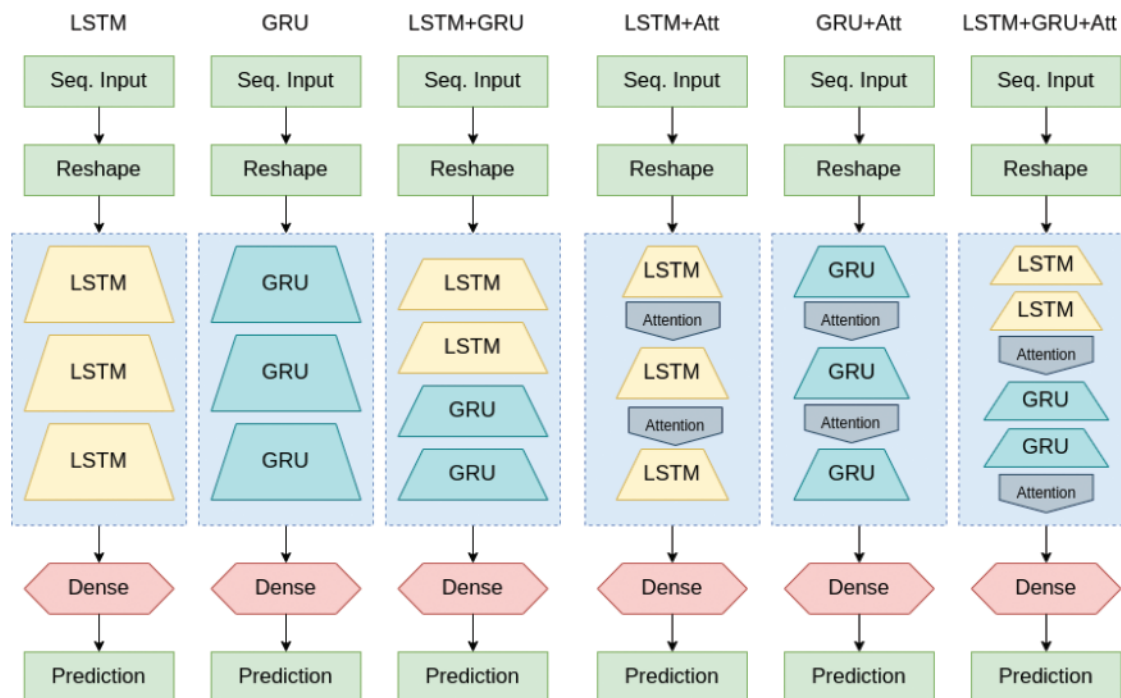


Figure 2: Proposed flow for prediction with LSTM, GRU, LSTM + GRU, LSTM + Attention, GRU + Attention, and LSTM + GRU + Attention models

2.6 Evaluation Metrics

Model evaluation is a crucial aspect of every research. In measuring the performance of the various methods used, we adopted the following evaluation metrics: mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), explained variance regression score, R2 score), mean gamma deviance regression loss (MGD), and mean Poisson deviance regression loss (MPD) for both the training data set and the testing data set.

3 Experimental Results and Discussion

The original data on the close prices and the predictions for all the stock indices are shown in Fig. 3. The following explains the results obtained on the various metrics.

3.1 BSE

The results, as presented in Tables 5 and 6, show that the GRU + Att algorithm performed the best overall, with the lowest RMSE, MSE, MAE, MGD, and MPD and the highest explained variance, R2 score on the test set. This suggests that the GRU + Att algorithm is able to provide the most accurate and consistent predictions for the BSE index.

On the other hand, the SVR algorithm performed the worst, with the highest RMSE, MSE, and MAE, and the lowest explained variance, R2 score. SVR also attained too high values for MGD and MPD on the test set. This suggests that the SVR algorithm is not well suited for the task of stock index prediction.

It is worth noting that the LSTM, LSTM + Att, GRU, LSTM+GRU, and LSTM + GRU + Att algorithms all performed relatively well, with RMSE, MSE, MAE, and R2 score values that were close to those of the GRU + Att algorithm. However, these algorithms did not perform as well in terms of explained variance, MGD, and MPD. From the results, the GRU + Att algorithm is the best choice for stock index prediction when considering the BSE. However, the other algorithms, LSTM, LSTM + Att, GRU, LSTM + GRU, and LSTM + GRU + Att are also good options, as they have similar performance in terms of RMSE, MSE, MAE, and R2 score, but not as well in terms of explained variance, MGD, and MPD.

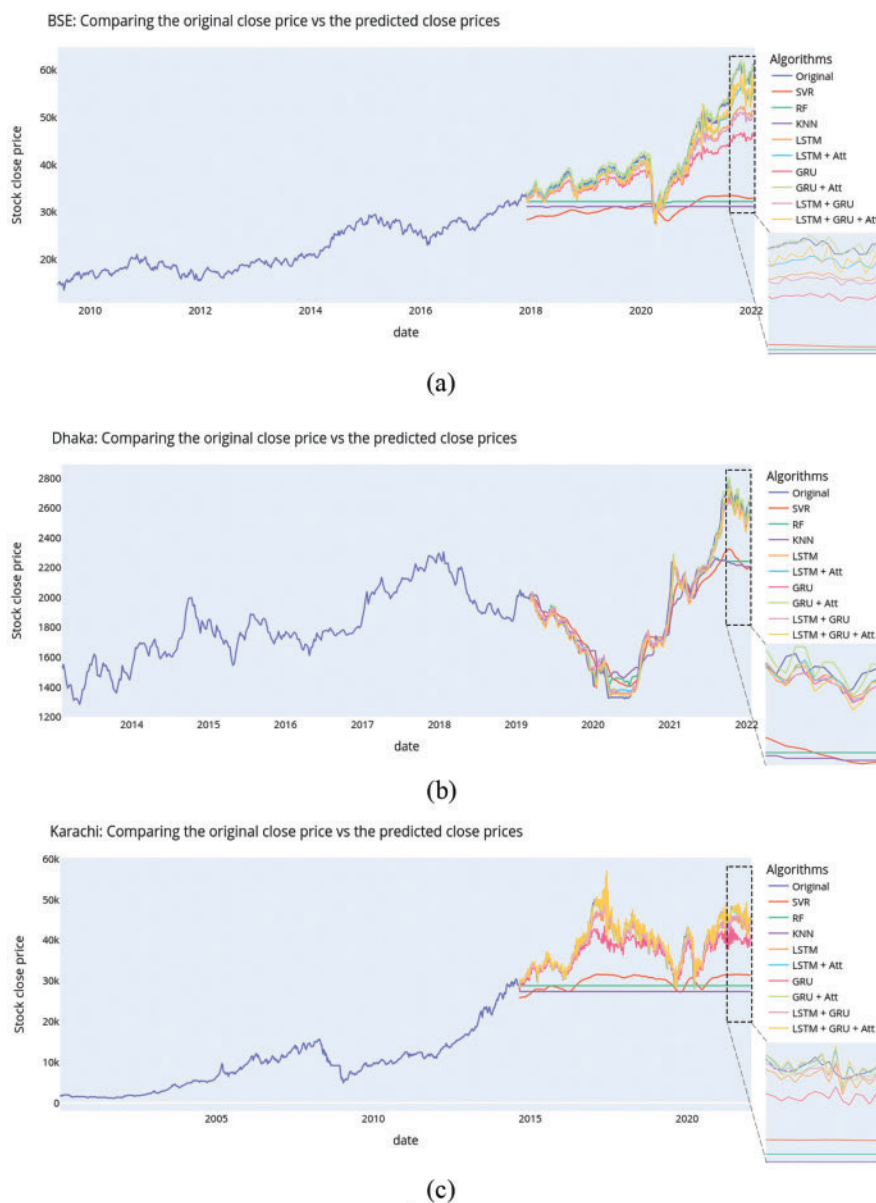


Figure 3: Comparing the original close prices and the predicted close prices for (a) BSE, (b) Dhaka, and (c) Karachi

Table 5: Results for RMSE, MSE, and MAE

Stock index	Algorithm	RMSE		MSE		MAE	
		Train	Test	Train	Test	Train	Test
BSE	SVR	2,410.31	12,569.63	5,809,601.11	157,995,702.13	2,158.71	10,585.75
	RF	188.19	12,356.13	35,416.05	152,674,013.92	148.76	9,376.29
	KNN	588.07	13,155.32	345,830.38	173,062,323.77	469.78	10,369.15
	LSTM	468.92	3,295.93	219,887.46	10,863,140.21	371.32	2,232.32
	LSTM + Att	571.63	2,028.11	326,756.93	4,113,238.56	444.03	1,429.84
	GRU	488.73	5,520.41	238,856.01	30,474,907.22	388.96	3,891.66
	GRU + Att	574.12	1,232.79	329,609.65	1,519,762.73	458.58	946.61
	LSTM + GRU	466.61	3,698.20	217,723.45	13,676,666.61	368.56	2,499.75
	LSTM + GRU + Att	586.68	2,005.89	344,195.76	4,023,610.00	477.21	1,515.58
DSE	SVR	62.39	170.89	3,892.60	29,203.72	50.18	122.71
	RF	13.47	163.01	181.31	26,573.50	10.57	97.86
	KNN	50.78	186.20	2,578.24	34,668.94	39.16	131.43
	LSTM	34.34	48.42	1,179.07	2,344.12	27.21	37.44
	LSTM + Att	38.08	56.67	1,450.25	3,211.07	29.56	43.96
	GRU	32.52	52.28	1,057.79	2,733.58	24.72	39.11
	GRU + Att	33.73	45.57	1,137.88	2,076.58	26.38	35.55
	LSTM + GRU	32.00	53.16	1,023.73	2,826.06	24.93	40.31
	LSTM + GRU + Att	39.07	52.91	1,526.10	2,799.75	30.66	40.00
KSE	SVR	2,783.07	10,615.67	7,745,463.39	112,692,496.04	2,294.41	9,713.17
	RF	125.74	12,113.12	15,811.21	146,727,718.15	81.04	10,711.75
	KNN	510.61	13,404.33	260,718.23	179,676,046.09	335.74	12,148.21
	LSTM	311.05	1,844.47	96,751.11	3,402,069.24	205.54	1,437.96
	LSTM + Att	316.20	1,437.82	99,981.77	2,067,340.44	215.33	1,024.57
	GRU	467.85	3,751.28	218,884.97	14,072,087.59	361.61	2,908.88
	GRU + Att	312.78	1,133.69	97,831.79	1,285,245.89	210.37	855.20
	LSTM + GRU	339.62	1,654.64	115,342.53	2,737,818.29	224.07	1,262.65
	LSTM + GRU + Att	320.09	1,645.11	102,456.38	2,706,381.52	216.98	1,162.08

Table 6: Results for explained variance, R^2 score, MGD, and MPD

Stock index	Algorithm	Explained variance		R^2 score		MGD		MPD	
		Train	Test	Train	Test	Train	Test	Train	Test
BSE	SVR	0.75	0.30	0.73	-1.35	0.012849	0.112087	269.84	4,165.70
	RF	1.00	0.01	1.00	-1.27	0.000077	0.105280	1.61	3,974.12
	KNN	0.98	0.01	0.98	-1.57	0.000752	0.125643	15.79	4,618.42
	LSTM	0.99	0.90	0.99	0.84	0.000483	0.004528	10.10	219.41
	LSTM + Att	0.99	0.96	0.99	0.94	0.000744	0.001806	15.28	84.66
	GRU	0.99	0.76	0.99	0.55	0.000528	0.013996	11.00	648.55
	GRU + Att	0.99	0.98	0.98	0.98	0.000675	0.000961	14.60	37.50
	LSTM + GRU	0.99	0.88	0.99	0.80	0.000481	0.005776	10.02	278.35
	LSTM + GRU + Att	0.99	0.97	0.98	0.94	0.000700	0.002060	15.20	88.97

(Continued)

Table 6 (continued)

Stock index	Algorithm	Explained variance		R ² score		MGD		MPD	
		Train	Test	Train	Test	Train	Test	Train	Test
DSE	SVR	0.92	0.84	0.92	0.82	0.001171	0.006245	2.12	13.31
	RF	1.00	0.85	1.00	0.84	0.000058	0.005348	0.10	11.77
	KNN	0.95	0.81	0.95	0.79	0.000797	0.007763	1.42	16.13
	LSTM	0.98	0.99	0.98	0.99	0.000372	0.000684	0.66	1.23
	LSTM + Att	0.97	0.98	0.97	0.98	0.000471	0.000922	0.82	1.68
	GRU	0.98	0.98	0.98	0.98	0.000338	0.000709	0.59	1.36
	GRU + Att	0.98	0.99	0.98	0.99	0.000361	0.000615	0.64	1.10
	LSTM + GRU	0.98	0.98	0.98	0.98	0.000327	0.000757	0.57	1.42
	LSTM + GRU + Att	0.98	0.99	0.97	0.98	0.000482	0.000743	0.85	1.41
KSE	SVR	0.88	0.42	0.80	-2.51	0.298827	0.094718	1,364.06	3,248.73
	RF	1.00	0.00	1.00	-3.57	0.000185	0.132139	1.44	4,377.25
	KNN	0.99	0.00	0.99	-4.60	0.003213	0.173947	23.96	5,550.79
	LSTM	1.00	0.93	1.00	0.89	0.001449	0.001993	9.13	81.69
	LSTM + Att	1.00	0.94	1.00	0.94	0.001413	0.001216	9.66	49.53
	GRU	1.00	0.79	0.99	0.56	0.002937	0.008306	19.42	340.61
	GRU + Att	1.00	0.96	1.00	0.96	0.002211	0.000837	10.37	32.36
	LSTM + GRU	1.00	0.93	1.00	0.91	0.002045	0.001635	11.56	66.34
	LSTM + GRU + Att	1.00	0.92	1.00	0.92	0.002791	0.001540	11.76	63.74

3.2 DSE

Also for the DSE, the results in [Tables 5 and 6](#) that the GRU + Att model performs the best in terms of RMSE, MSE, MAE, explained variance, and R-squared score, with the lowest values for these metrics. This was followed by the LSTM model, indicating that these models are able to make more accurate predictions on the DSE stock index compared to the other algorithms. The GRU + Att and the LSTM models have RMSE of 45.57 and 48.42 respectively, which are significantly lower than the other models, and both have R-squared scores of 0.99, which is the highest among all the models.

It is also worth noting that the GRU + Att and LSTM models have the lowest MGD and MPD values. This suggests that these models have the smallest deviation from the actual stock prices and are able to predict stock prices with high accuracy.

On the other hand, the SVR and KNN models perform poorly in comparison, with high RMSE, MSE, MAE, MGD, and MPD values. The SVR model has an RMSE of 170.89, which is the highest among all the models, and an R-squared score of 0.84. The KNN model also has a high RMSE of 186.20 and an R-squared score of 0.79. The RF performed better than the SVR and KNN models but its performance is worse than the RNN-based models.

The GRU, LSTM + ATT, LSTM + GRU, and LSTM + GRU + Att models perform similarly to LSTM, GRU + Att models, with slightly higher RMSE, MSE, MAE and slightly lower R-squared score, explained variance, MGD, and MPD values.

In conclusion, the GRU + Att and LSTM models have the best performance in terms of accuracy in predicting the DSE stock index, followed by the GRU model, LSTM + GRU + Att model, LSTM + GRU model, and RF model, while the SVR and KNN models show the worst performance. The use of attention mechanisms in GRU models has been shown to have slightly improved performance.

Based on these results, it can be recommended to use GRU + Att or LSTM models for stock price prediction on the DSE index.

3.3 KSE

It can be seen that the LSTM + Att and GRU + Att models performed the best in terms of RMSE and MAE, with the lowest values for the test set. These models also had the highest explained variance and R-squared scores, indicating that they were able to accurately capture the underlying trends in the data. The LSTM + mAtt and GRU + Att models also had the lowest values for MGD and MPD, indicating that they had the smallest deviation from the true values.

On the other hand, the SVR model performed the worst in terms of all the evaluation metrics. The RF and KNN models performed slightly better than the SVR model, but still had relatively high values for RMSE, MSE, and MAE. The GRU, LSTM, GRU, and LSTM + GRU + Att models performed similarly to the LSTM + Att and GRU + Att models in terms of RMSE and MAE but had slightly lower explained variance and R-squared scores. Hence, the LSTM + Att and GRU + Att models performed the best for stock index prediction on the KSE dataset.

These models were able to capture the underlying trends in the data and had the smallest deviation from the true values. The SVR model performed the worst, while the RF and KNN models had relatively high errors. The LSTM, GRU, LSTM + GRU, and LSTM + GRU + Att models performed similarly to the LSTM and LSTM + Att models, but with slightly lower explained variance and R-squared scores.

3.4 Using Slices of the BSE, DSE, and KSE Data

Tables 7 and 8 and Fig. 4 are the results obtained when we only used data with closing prices above 2014. From Table 7, the GRU + Att model obtained the best results on the RMSE, MSE and MAE for the BSE index. The LSTM + GRU + Att model also performed close to the GRU + Att model on the metrics for the same BSE index. Considering the DSE index, the GRU + Att model again got the lowest RMSE of 52.86, which is closely followed by the LSTM + GRU model with 58.71 and then the GRU model with 60.40 and the LSTM and LSTM + GRU + Att with 80.02 and 85.45 respectively on the test data. Again for the DSE, the lowest values for the MSE and MAE metrics were obtained by the GRU + Att model. Similar results were obtained on the KSE index where the GRU and GRU + Att models performed best on the RMSE. The MSE metrics. For the MAE, the GRU + Att had the lowest value while the LSTM + Att, the GRU, and the LSTM + GRU + Att were around the same values. The SVR and the KNN performed worst in this case.

Table 7: Results for RMSE, MSE, and MAE using data greater than 2014

Stock index	Algorithm	RMSE		MSE		MAE	
		Train	Test	Train	Test	Train	Test
BSE	SVR	1,777.24	10,328.40	3,158,594.59	106,675,909.96	1,470.15	8,650.30
	RF	218.51	10,756.53	47,745.95	115,703,043.42	169.11	8,078.43
	KNN	730.96	11,533.72	534,305.18	133,026,742.31	566.87	9,094.97
	LSTM	673.40	2,743.37	453,470.50	7,526,065.28	520.19	2,054.81
	LSTM + Att	757.65	3,386.56	574,027.84	11,468,810.91	624.40	2,845.00
	GRU	583.03	3,761.40	339,925.74	14,148,111.28	462.05	2,845.93
	GRU + Att	553.74	2,029.00	306,628.73	4,116,828.94	430.87	1,550.39
	LSTM + GRU	840.30	4,599.78	706,103.47	21,158,021.07	713.50	3,686.04

(Continued)

Table 7 (continued)

Stock index	Algorithm	RMSE		MSE		MAE	
		Train	Test	Train	Test	Train	Test
DSE	LSTM + GRU + Att	615.47	2,387.45	378,803.32	5,699,926.37	488.92	1,846.63
	SVR	57.32	186.10	3,286.08	34,634.73	45.37	146.30
	RF	12.31	224.41	151.53	50,360.21	9.64	163.61
	KNN	47.59	259.66	2,264.43	67,421.73	35.90	209.83
	LSTM	29.19	80.02	851.97	6,403.01	23.28	63.91
	LSTM + Att	52.24	85.56	2,729.22	7,321.35	43.94	67.69
	GRU	29.27	60.40	856.61	3,648.65	22.76	47.78
	GRU + Att	32.12	52.86	1,031.99	2,794.11	24.72	40.44
	LSTM + GRU	32.37	58.71	1,047.73	3,447.03	26.90	45.68
	LSTM + GRU + Att	45.56	85.45	2,075.99	7,301.46	36.66	71.03
KSE	SVR	1,148.81	1,444.73	1,319,763.83	2,087,248.87	913.67	1,014.95
	RF	363.53	1,245.19	132,152.68	1,550,499.38	277.69	980.52
	KNN	1,269.49	2,521.16	1,611,610.80	6,356,269.09	1,005.98	1,917.71
	LSTM	896.77	1,248.06	804,189.20	1,557,662.68	680.26	957.22
	LSTM + Att	1,066.78	1,300.66	1,138,021.14	1,691,712.72	842.91	866.49
	GRU	875.43	1,059.47	766,383.48	1,122,467.16	679.77	836.84
	GRU + Att	945.71	1,091.13	894,375.32	1,190,562.75	734.34	755.78
	LSTM + GRU	866.93	1,222.48	751,559.61	1,494,460.13	658.72	936.92
	LSTM + GRU + Att	1,069.87	1,263.63	1,144,618.99	1,596,764.77	858.18	854.60

Table 8: Results for explained variance, R^2 score, MGD, and MPD using data greater than 2014

Stock index	Algorithm	Explained variance		R^2 score		MGD		MPD	
		Train	Test	Train	Test	Train	Test	Train	Test
BSE	SVR	0.86	0.46	0.85	-0.30	0.003125	0.054293	98.39	2,392.27
	RF	1.00	0.29	1.00	-0.41	0.000052	0.058821	1.56	2,597.00
	KNN	0.97	0.18	0.97	-0.62	0.000536	0.070654	16.71	3,049.31
	LSTM	0.98	0.93	0.98	0.91	0.000447	0.003062	14.05	149.04
	LSTM + Att	0.98	0.93	0.97	0.86	0.000616	0.004903	18.57	233.32
	GRU	0.99	0.90	0.98	0.83	0.000382	0.005553	11.26	277.35
	GRU + Att	0.99	0.97	0.99	0.95	0.000338	0.001901	10.06	86.36
	LSTM + GRU	0.98	0.89	0.97	0.74	0.000716	0.008596	22.24	422.82
	LSTM + GRU + Att	0.98	0.95	0.98	0.93	0.000431	0.002479	12.63	116.48
DSE	SVR	0.90	0.85	0.90	0.83	0.000858	0.007749	1.67	16.07
	RF	1.00	0.76	1.00	0.76	0.000042	0.013415	0.08	25.41
	KNN	0.93	0.67	0.93	0.67	0.000613	0.018517	1.17	34.69
	LSTM	0.97	0.97	0.97	0.97	0.000232	0.002188	0.44	3.64
	LSTM + Att	0.95	0.97	0.91	0.96	0.000749	0.002673	1.42	4.35
	GRU	0.98	0.98	0.97	0.98	0.000231	0.001056	0.44	1.91
	GRU + Att	0.97	0.99	0.97	0.99	0.000282	0.000786	0.54	1.45
	LSTM + GRU	0.97	0.98	0.97	0.98	0.000290	0.001164	0.55	1.95
	LSTM + GRU + Att	0.97	0.97	0.93	0.96	0.000562	0.001989	1.08	3.70

(Continued)

Table 8 (continued)

Stock index	Algorithm	Explained variance		R^2 score		MGD		MPD	
		Train	Test	Train	Test	Train	Test	Train	Test
KSE	SVR	0.95	0.91	0.95	0.91	0.000809	0.001461	32.41	54.81
	RF	1.00	0.93	1.00	0.93	0.000085	0.000992	3.31	38.79
	KNN	0.94	0.74	0.94	0.71	0.000941	0.004202	38.64	162.50
	LSTM	0.97	0.93	0.97	0.93	0.000480	0.001088	19.48	40.76
	LSTM + Att	0.96	0.93	0.96	0.92	0.000713	0.001219	28.25	45.02
	GRU	0.97	0.95	0.97	0.95	0.000480	0.000758	19.03	28.90
	GRU + Att	0.97	0.95	0.97	0.95	0.000543	0.000861	21.83	31.66
	LSTM + GRU	0.97	0.94	0.97	0.93	0.000465	0.001051	18.55	39.26
	LSTM + GRU + Att	0.97	0.94	0.96	0.93	0.000701	0.001094	28.08	41.38

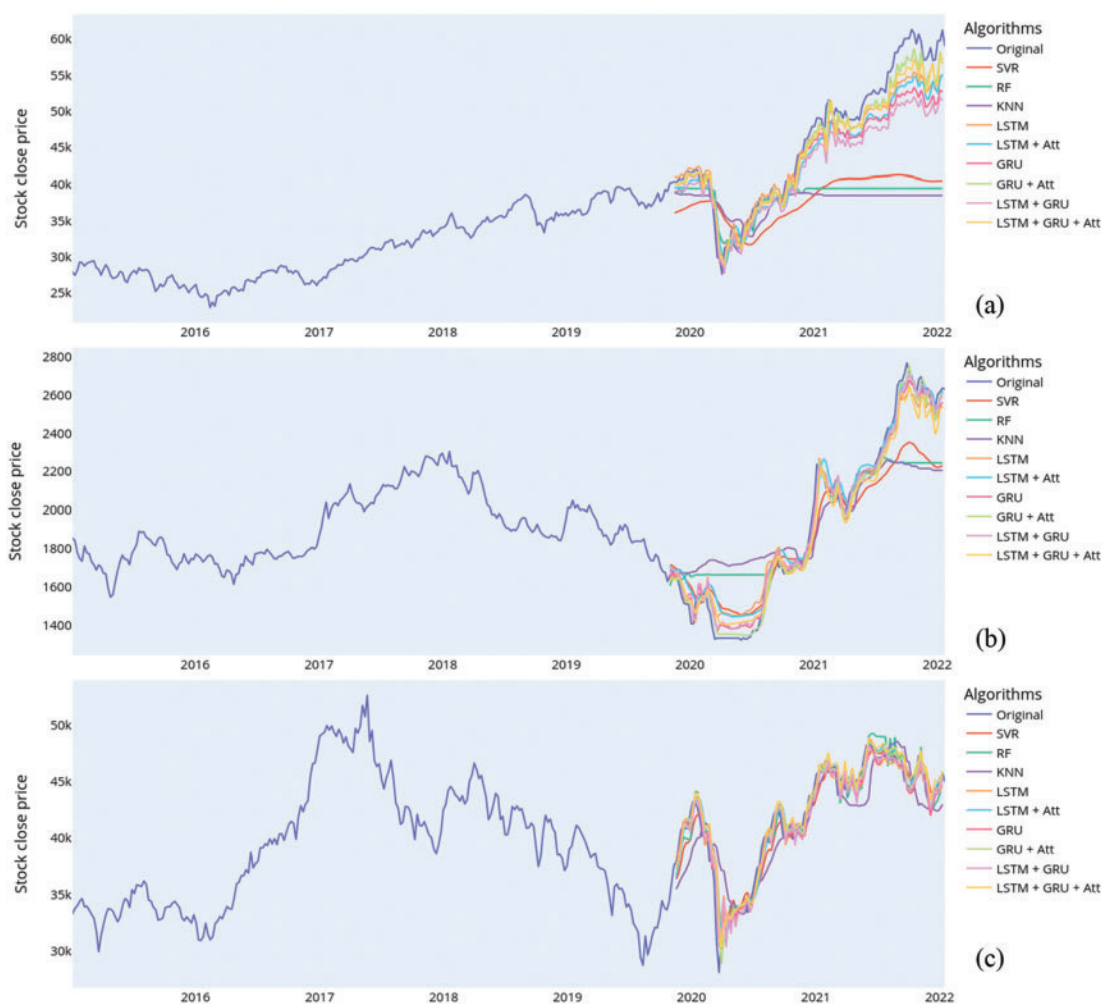
**Figure 4:** Using a slice of the data above the year 2014 for the experiment, the sub-figures compare the original close prices and the predicted close prices for (a) BSE, (b) Dhaka, and (c) Karachi

Table 8 presents the results for the three indices on the explained variance, the R2 score,

the MGD and the MPD. Considering the BSE index, the GRU + Att model had an explained variance of 0.97 followed by the LSTM + GRU + Att with 0.95. The other RNN models also performed much better on the RMSE than the SVR, RF, and KNN for the BSE test data. For the DSE index, the highest explained variance and R2 of 0.99 on the test data were obtained by the GRU + Att model. The lowest MGD and MPD values were again obtained by the GRU + Att model making it the best model for DSE. The other RNN models and their attention-based variations also outperformed the SVR, RF, and KNN. Lastly, considering the KSE, the GRU and GRU + Att models had the highest variance of 0.95 and R2 score of 0.95 On the test data. This is followed by the LSTM + GRU and LSTM + GRU + Att with 0.94 And 0.93 respectively. Adding attention to the LSTM however, did not improve the LSTM on the KSE data. Likewise, the attention does not improve the MGD and MPD for the GRU and the LSTM + GRU models.

Though using the entire dataset allows the models to better capture long-term trends in the stock market, using a more recent slice of the data enables the models to learn to become better suited to predicting current market trends.

4 Conclusion

Predicting the stock market is an interesting research area for both economists and computer scientists. Classical machine learning techniques have been used in much of the literature for predictions, but recent advances in RNN have proven to be very effective with sequence-based data in several research domains. In this study, we evaluated the performance of RNN-based models, including Long-Short Term Memory (LSTM), Gated Recurrent Unit (GRU), and a hybrid model of LSTM and GRU, in time series forecasting;

Predicting the closing prices of three stock indices, namely the KSE 100 index, the DSE 30, and the BSE Sensex. We also explored the use of attention mechanisms in these models to improve their performance. Lastly, traditional machine learning models namely SVR, RF, and KNN were used as baseline models. Our results showed that the hybrid model of LSTM and GRU outperformed the single models of LSTM and GRU. The attention mechanism was also found to be effective in improving the performance of these models. SVR, RF, and KNN models obtained the worst performances on our data.

Overall, our study highlights the effectiveness of RNN-based models and the attention mechanism in time series forecasting. These models provide an effective solution for tasks that require the analysis of input sequences of variable length and temporal dynamic characteristics.

This work provides a basis for further studies on the use of RNNs and attention mechanisms in time series forecasting. Other deep learning techniques such as transfer learning, data augmentation, etc. for time series predictions are also worth studying in the future. While this paper only focused on using historical stock prices and trends. Other factors including economic indicators such as gross domestic product (GDP), inflation, and interest rates do affect future trends and hence further studies are required.

Acknowledgement: This research work was carried out as a joint research collaboration at the University of Electronic Science and Technology of China, Chengdu China, Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Huzhou, 313001, China, and Adbul wali khan Mardan KPK, Pakistan and Brain Institute Peshawar, Peshawar, 25130,

Pakistan. The authors are thankful for this support. The Cobbinah Bernard Mawuli supporting in the proofreading of the revised manuscript.

Funding Statement: This research work is supported by NRPU Project No. 20-16091 awarded by Higher Education Commission, Pakistan. The title of the project is “University Education and Occupational Skills Mismatch (A Case Study of SMEs in Khyber Pakhtunkhwa)”, by the National Natural Science Foundation of China (Grant No. 61370073), the National High Technology Research and Development Program of China, the project of Science and Technology Department of Sichuan Province (Grant No. 2021YFG0322).

Author Contributions: Conceptualization, A.U.H., and A.H.; B.L.Y.A, J.P.L, and A.S methodology, A.U.H., and B.L.Y.A., A.S, software, A.U.H.; validation, A.U.H., and A.H.; A.H.U, formal analysis, A.U.H., R.K, and B.L.Y.A.; investigation, A.U.H., J. P. L, and A.S., resources, A.U.H., and B.L.Y.A., and A.S A; writing–original draft preparation, A.U.H.; writing–review and editing, B.L.Y.A., A.U.H; visualization, A.U.H.; supervision, A.H.; J.P.L, and A.S project administration, A.U.H.; funding acquisition, R.K, and A.H. All authors have read and agreed to the published version of the manuscript.

Availability of Data and Materials: Data for this study is extracted from investing.com, URL: (<https://www.investing.com/indices> indices & major India additional Indices) and can be made available upon request. The corresponding author Amin ul Haq ([http://khan.amin50@yahoo.com](mailto:khan.amin50@yahoo.com)) would be contacted for data availability.

Ethics Approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] R. Halil and M. Demirci, “Predicting the turkish stock market bist 30 index using deep learning,” *International Journal of Engineering Research and Development*, vol. 11, no. 1, pp. 253–265, 2019.
- [2] L. Li, Y. Wu, Y. Ou, Q. Li, Y. Zhou *et al.*, “Research on machine learning algorithms and feature extraction for time series,” in *2017 IEEE 28th Annual Int. Symp. on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Montreal, M, Canada, pp. 1–5, 2017.
- [3] Z. Hu, W. Liu, J. Bian, X. Liu and T. Liu, “Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction,” in *Proc. of the Eleventh ACM Int. Conf. on Web Search and Data Mining*, New York, NY, USA, pp. 261–269, 2018.
- [4] J. Li, “Research on market stock index prediction based on network security and deep learning,” *Security and Communication Networks*, vol. 2021, no. 1, pp. 1–8, 2021.
- [5] X. Zhong and D. Enke, “Forecasting daily stock market return using dimensionality reduction,” *Expert Systems with Applications*, vol. 67, no. 1, pp. 126–139, 2017.
- [6] E. Guresen, G. Kayakutlu and T. U. Daim, “Using artificial neural network models in stock market index prediction,” *Expert Systems with Applications*, vol. 38, no. 8, pp. 10389–10397, 2011.
- [7] Y. Bing, J. K. Hao and S. C. Zhang, “Stock market prediction using artificial neural networks,” in *Advanced Engineering Forum*, vol. 6, Switzerland: Trans Tech Publications, pp. 1055–1060, 2012.
- [8] S. Rajab and V. Sharma, “An interpretable neuro-fuzzy approach to stock price forecasting,” *Soft Computing*, vol. 23, no. 3, pp. 921–936, 2019.

- [9] M. Sedighi, H. Jahangirnia, M. Gharakhani and S. F. Fard, "A novel hybrid model for stock price forecasting based on metaheuristics and support vector machine," *Data*, vol. 4, no. 2, pp. 75, 2019.
- [10] P. Hushani, "Using autoregressive modelling and machine learning for stock market prediction and trading," in *Third International Congress on Information and Communication Technology*. London, L: UK, pp. 767–774, 2019.
- [11] Investing.com, *Stock market quotes & financial news*. Investing.com, 2023. [Online]. Available: <https://www.investing.com>
- [12] A. U. Haq, J. P. Li, B. L. Y. Agbley, C. B. Mawuli, Z. Ali *et al.*, "A survey of deep learning techniques based Parkinson's disease recognition methods employing clinical data," *Expert Systems with Applications*, vol. 208, no. 3, pp. 118045, 2022.
- [13] A. U. Haq, J. P. Li, Z. Ali, I. Khan, A. Khan *et al.*, "Stacking approach for accurate invasive ductal carcinoma classification," *Computers and Electrical Engineering*, vol. 100, no. 3, pp. 107937, 2022.
- [14] U. D. Maiwada, "Introductory technology as it impacts modern society in the world," *Journal of Technology Innovations and Energy*, vol. 1, no. 1, pp. 36–42, 2022.
- [15] A. U. Haq, J. P. Li, B. L. Y. Agbley, A. Khan, I. Khan *et al.*, "IIMFCBM: Intelligent integrated model for feature extraction and classification of brain tumors using MRI clinical imaging data in IoT-healthcare," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 10, pp. 5004–5012, 2022.