Received 2 July 2024, accepted 17 July 2024, date of publication 29 July 2024, date of current version 11 September 2024.
Digital Object Identifier 10.1109/ACCESS.2024.3434524

RESEARCH ARTICLE

Enhancing Stock Market Prediction: A Robust LSTM-DNN Model Analysis on 26 Real-Life Datasets

KHORSHED ALAM, (Member, IEEE), MAHBUBUL HAQ BHUIYAN, INJAMAM UL HAQUE, MD FAHAD MONIR, (Member, IEEE), AND TAREM AHMED, (Senior Member, IEEE)
Department of Computer Science and Engineering, Independent University, Bangladesh (IUB), Dhaka 1212, Bangladesh
Corresponding author: Tarem Ahmed (tarem@iub.edu.bd)

ABSTRACT Predicting the closing price of the stock market with accuracy is highly uncertain and volatile. Deep learning (DL) can analyze vast amounts of historical stock data to identify patterns and correlations, aiding in predictive modeling. By learning from past market behavior, deep learning algorithms can potentially forecast future price movements with some degree of accuracy. This research presents a hybrid Long Short-Term Memory (LSTM) and Deep Neural Network (DNN) model designed to tackle the complexities of stock market prediction, including market volatility and intricate patterns. Initially the proposed model is being trained on Bajaj's stock dataset. Getting noteworthy performance in single dataset does not prove robustness of the model. The research validates the model's robustness and scalability through rigorous comparative analysis on 26 company's stock datasets, achieving an average R-squared (R2) score of 0.98606, a Mean Absolute Error (MAE) of 0.0210, and a Mean Squared Error (MSE) of 0.00111. To assess the contribution of our proposed model, we retrained previously used deep learning models alongside our new approach, utilizing a shared dataset for validation. Additionally, we provide ablation study of LSTM-DNN model which provides insights into the individual contributions of each component towards detecting closing price of stocks, offering valuable information for optimizing future stock market prediction models. The model's exceptional performance sets a new standard in stock market prediction, offering promising implications for investors, traders, and financial analysts. Making our work available open-source on https://github.com/codewithkhurshed/SMP-IUB can enhance its accessibility and promote future research opportunities in stock market price prediction.

INDEX TERMS Neural networks, regression model, stock market analysis, stock market prediction, LSTM, DNN.

I. INTRODUCTION

Stock market prediction is the attempt to predict the future value of a company's stock or other financial instrument traded on an exchange [1]. An correct projection of a stock's future price could result in a significant profit. According to efficient-market theory, stock prices accurately reflect all currently known information, and any price changes that are not based on recently disclosed information are thus intrinsically unexpected. Others disagree, and those that share this viewpoint have a range of strategies and technologies that apparently allow them to predict future pricing.

Predicting stock market trends demands intricate analysis as it depends on multifaceted variables [2]. Its complexity attracts attention due to its pivotal role in shaping financial decisions, yet its elusive nature underscores the challenge of achieving consistent accuracy. Creating a prediction model using deep learning for stock markets is challenging due to the inherent volatility, non-linear patterns, and limited historical data's inability to capture all market dynamics accurately.

The associate editor coordinating the review of this manuscript and approving it for publication was Sajid Ali.

Additionally, market sentiment and external events further complicate modeling.

In the realm of time series prediction analysis, numerous conventional strategies rooted in statistical modeling have been introduced. Among these strategies are linear regression, auto-regression, and moving average, commonly encapsulated in the Auto Regressive Moving Average (ARMA) framework [6], [7]. ARMA emerges as a powerful tool for handling time series data that unfolds over successive instances. This model amalgamates distinct features of the data through its auto-regressive component, which addresses one aspect, and the moving average component, which tends to another feature. Notable mentions include Auto Regressive Conditional Heteroskedasticity (ARCH), Long-Short Term Memory (LSTM) [3], [4], Bidirectional Long-Short Term Memory (BiLSTM), Convolutional Neural Network-Bidirectional Long-Short Term Memory (CNN-BiSLSTM) based hybrid model [5] and Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) models [8]. These models cater to the nuanced fluctuations in time series data. Consequently, a range of models has been proposed that demonstrate particular efficacy in forecasting and dissecting volatility patterns [9]. This array of models play a pivotal role in learning time series data dynamics.

We propose a LSTM-DNN model based custom architecture for predicting close to reality stock closing prices and achieve a noteworthy accuracy on predicting stock prices on test data. The decision to use LSTM-DNN models for stock market prediction is driven by their ability to capture complex patterns, achieve superior predictive accuracy, demonstrate robustness and scalability, and potentially be applied in real-time scenarios. DNNs are great at spotting complex, tricky patterns in data, while LSTM networks are excellent at figuring out what comes next in a sequence, which is perfect for time-based data like stock market data. When we combine these two together, we get the best of both models, allowing us to find all the complex variable to predict future stock price efficiently. The contribution of this study includes the following:

- The paper introduces a novel approach to stock market prediction by combining LSTM networks with DNNs. This hybrid model capitalizes on the strengths of both architectures to enhance predictive accuracy.
- We propose a DNN-LSTM based custom architecture for predicting stock prices which accurately on 26 company's datasets with an average of R-squared (R2) score of 0.98606, a MAE of 0.0210, and a MSE of 0.00111 shown in Table 2. Predicting stock closing price which is close to reality in 26 real life datasets is unseen with such notable performance in any previous studies. We validated this information from recent IEEE Access review paper [18] which reviewed the performance of previous work related to stock market prediction domain until September 2023.
- By training and evaluating the LSTM-DNN model on a broader set of company stocks, the research contributes to the generalizability of its findings. This contrasts with previous studies that focus on a limited number of companies, thereby enhancing the applicability of the proposed approach. To ensure contribution we compare our model's performance with widely used DL approaches having a common dataset highlighted in Table 3.
- Unlike previous studies focused solely on daily changes, this research extends the prediction timeframe to capture longer-term trends in stock prices by developing the proposed model on nifty 50 dataset (2000-2021). This addresses a limitation highlighted in previous studies and enhances the usefulness of the model for investors and traders.
- The research systematically addresses several limitations highlighted in section II. By conducting a thorough comparative analysis, the study fills a critical gap identified by previous researchers who lacked such comprehensive evaluations. This contributes to a more nuanced understanding of the effectiveness of different prediction models and techniques.

The paper is organized into several key sections that systematically present the research findings and methodology. In Section II, we discuss previous work and their limitations. Section III covers the methodology of this research. Furthermore, Section IV describes the performance of the proposed model. In Section V, we discuss the connection between potential profitability and our proposed model. In the next section, we describe how our proposed model deals with market volatility and intricate patterns. In Section VII, we provide a discussion on why the LSTM-DNN model is suitable for addressing application issues in stock market prediction. In Section VIII, we address limitations, and in Section IX, we discuss future work. Finally, we conclude our research in Section X.

II. PREVIOUS WORK

Predicting stock market trends and movements is a challenging task that has garnered significant attention due to its potential implications in financial decision-making. The study [10] proposes LSTM based prediction model for stock market prediction using nifty fifty datasets and achieves 83.88% accuracy. However, the study does not mention comparisons with other forecasting models or methods. This makes it challenging to determine whether the LSTM model is genuinely superior to existing techniques. The paper [11] presents a novel outlier mining algorithm is proposed to detect anomalies on the basis of volume sequence of high frequency tick-by tick data of stock market. Author compared the algorithm's performance with a traditional cluster analysis (like k-means) is a good step, but it does not guarantee that the novel algorithm is superior in all cases or for all stocks.

Authors in [12] propose an SVM based model to predict stock price based on the efficient market hypothesis. They fetch the stock comments information from social media and then preprocessing the data to emotion vectors. However, the social media data can be noisy, containing irrelevant or misleading information. It is crucial to ensure the data quality and address noise before drawing conclusions. Another paper [13] uses the Elman network to predict the opening price of the stock market. The authors use a self-adapting form of the PSO method to improve network weights and thresholds. The optimized data, referred to as the starting weight and threshold value, is then sent into the Elman network for training, resulting in the formation of a prediction model for stock market opening prices based on the self-adapting variation PSO-Elman network. Although the model's fault tolerance is discussed, the study should focus on how well the model performs when confronted with noisy or unpredictable data.

The study [14] aimed to predict the daily price changes of three major stocks traded on the Borsa Istanbul 100 index. Technical indicators calculated using stock prices and dollar-gold prices were used as features for prediction. Class labels representing price changes were determined based on the closing prices of the stocks. Two different Convolutional Neural Network (CNN) models were trained and evaluated using accuracy and Macro-average F1-score metrics. However, the study focused on daily price changes, which might not fully capture longer-term trends or sudden market movements. Another study at [15] suggests a method for predicting the closing price of stocks using autoencoder long short-term memory (AE-LSTM) networks. However, the paper mentions predicting "daily" prices. If the model is only capable of short-term predictions, its usefulness could be limited for investors and traders looking to make longer-term decisions. The study conducted in [20] examines Stock Market Analysis and Prediction utilizing an LSTM-based Approach. It reveals a considerable disparity between the predicted and actual stock prices.

The study [23] proposes CNN-BiSLSTM based stock market price prediction model. However, it only predicts the closing price of stock for the next trading day, limiting its reference value for investment. Investors prefer to predict the price and trend of the stock over a longer period, suggesting a limitation in the study's timeframe for prediction. Authors from [24] proposed machine learning models (KNN, RF, LR, GB) to predict the next day's closing price of stocks from three different companies, with evaluation based on R2, RMSE, and MAE. The study [24] only focused on three companies, which may limit the generalizability of the findings to a broader range of companies in the stock market. The study [25] utilized Artificial Neural Network and Random Forest techniques for stock closing price prediction, using financial data to create new variables as inputs, and evaluating the models with RMSE and MAPE. However, lack of discussion on specific limitations of machine learning models. The methodology proposed by [26] involves using RNNs, specifically LSTM models, along with stock data and technical indicators for stock closing price prediction. PCA is used for dimension reduction, and optimization strategies like Adam and Glorot uniform initialization are applied. The faces challenges in accurately predicting stock market prices due to high volatility. Authors from [27] and [28] uses machine learning approaches to predict stock market. However, these models were trained on limited data samples which can not promise the robustness in terms of prone to sudden changes.

A. DEALING WITH LIMITATIONS

Our work addresses several limitations highlighted in previous studies on stock market prediction. Firstly, we conduct a comparative analysis, contrasting our hybrid architecture based on LSTM-DNN model with existing techniques, addressing the concern raised by [10], which lacked such comparisons. This comparison helps establish the superiority of our proposed model. Secondly, while [11] focuses on outlier mining algorithms, our research emphasizes the importance of data quality and noise handling, particularly in the context of utilizing 26 real life company's stock data for stock market prediction. This ensures the robustness and reliability of our predictive model, a point of concern raised by [12]. Moreover, our research extends the prediction timeframe beyond just daily changes, aiming to capture longer-term trends in stock prices, thus addressing a limitation highlighted by [13]. Additionally, our work offers predictions not only for daily price changes but also for longer-term trends, addressing a limitation of studies that focus solely on short-term predictions, as noted by [14]. By aiming to predict price movements over a longer period, our research enhances the usefulness of our model for investors and traders, contrary to the limited scope of prediction highlighted in [15]. Furthermore, our hybrid LSTM-DNN model is trained and evaluated on a broader set of company stocks, contributing to the generalizability of its findings compared to studies that focus on a limited number of companies, such as [20] and [23]. By conducting a comparative analysis on 26 company stocks, our research aims to enhance the generalizability of its findings compared to studies that focus on a limited number of companies, as mentioned in [24]. Overall, our work provides a comprehensive approach to stock market prediction, addressing various limitations observed in prior studies and contributing to a more robust understanding of predictive modeling in finance.

III. METHODOLOGY

Long Short-Term Memory (LSTM) [16] and Deep Neural Network (DNN) [17] based hybrid custom architecture is used to develop for achieving this framework. This hybrid approach It analyzes information in complex ways using advanced math modeling. Our framework is divided into the following three parts.

The first one is Dataset Formation where we gather stock market data from various datasets. Second one is

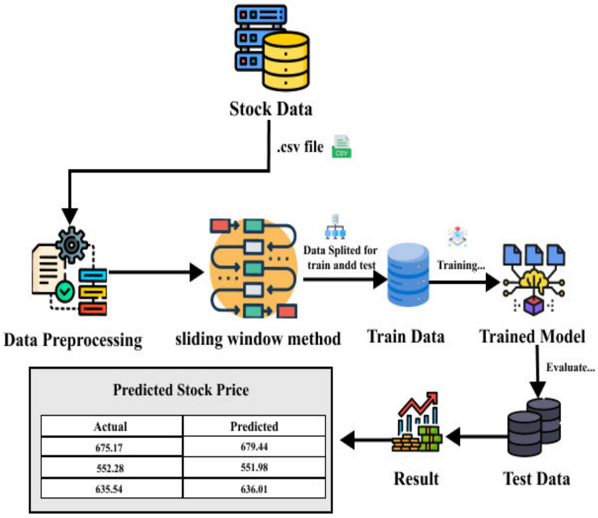


FIGURE 1. Comprehensive stock market prediction methodology.

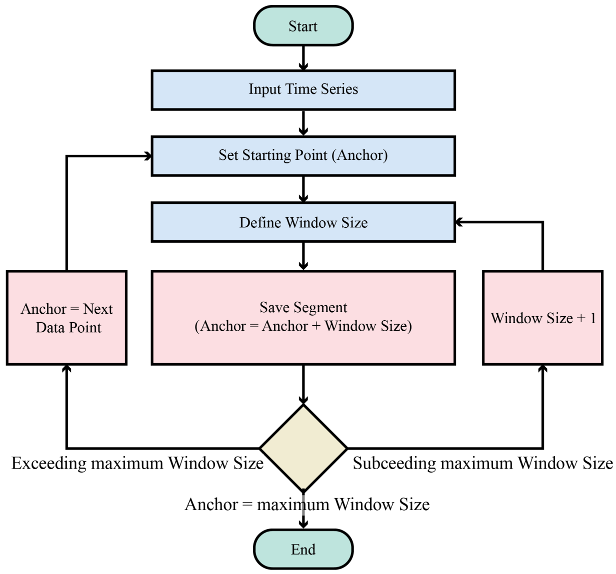


FIGURE 2. Illustration of the sliding window approach for LSTM.

Data Preprocessing. In this part, to make the best out of datasets, we preprocess our data with rolling window approach algorithm. Last one is Model Development where we develop a LSTM-DNN based deep learning model for framework shown in Fig. 1.

A. DATASET FORMATION

In this study, we have collected Google's Stock market data from Kaggle, and we collected 25 company's Data from NIFTY-50 [13] Stock Market Data (2000 - 2021) Kaggle repository. All of the stock data have 5303 data points in each dataset. The data spans from 1st January 2000 to 30th April 2021.

B. DATA PREPROCESSING

The sliding window method in time series data preprocessing involves iteratively segmenting the data into fixed-sized windows shown in Fig 2. This method aids in capturing changing patterns over time by creating overlapping or non-overlapping segments, enabling analysis of temporal dynamics and enhancing predictive modelling accuracy.

The process involves creating input-output pairs from historical data using a rolling window approach. Two lists store input and target sequences, respectively. The "window size" variable is set to define the length of the rolling window. An outer loop iterates over data indices, forming pairs within each window. The baseline value is stored for normalization. An inner loop constructs the input sequence by calculating percentage changes for each data point. "temp2" calculates the percentage change for the next data point after the window. The input and output sequences are appended to lists as NumPy arrays, forming pairs for model training.

C. MODEL DEVELOPMENT

We propose a LSTM-DNN based hybrid model for predicting future closing price of stock market and that ensures a R-squared (R2) score of 0.981352, a Mean Absolute Error (MAE) of 0.0092, and a Validation MAE of 0.0114. Let us begin by discussing the Model Architecture shown in Table 1.

TABLE 1. Model architecture.

Layer (type)	Output Shape	Param
Lstm 38 (LSTM)	(None, 1, 16)	117248
Lstm 39 (LSTM)	(None, 32)	49408
Dense 95 (Dense)	(None, 64)	13000
Dense 96 (Dense)	(None, 64)	100500
Activation 57 (Activation)	(None, 64)	0
Dense 97 (Dense)	(None, 64)	275550
Activation 58 (Activation)	(None, 64)	0
Dense 98 (Dense)	(None, 64)	82650
Activation 59 (Activation)	(None, 64)	0
Flatten 19 (Flatten)	(None, 64)	0
Dense 99 (Dense)	(None, 1)	151
Total Params		**28417 (111.00 KB)**
Trainable Params		**28417 (111.00 KB)**
Non-trainable Params		**0**

1) LONG SHORT-TERM MEMORY (LSTM)

A Long Short-Term Memory network is a type of recurrent neural network that can hold over long-term input data sequences. LSTM is good for those problems where we rely on long input data sequences.

In figure 3, c_{t-1} is the previous cell state where c_t is the next cell state. h_{t-1} is the previous hidden state and h_t is the next hidden state. x_t are our current inputs. In our case it is accelerometer data and timestamps.

Forget gate is a type of state where cell decides which information must be keep or forget by taking the weights and biases with respect to x_t in sigmoid activation function. So that it can return values from 0 to 1. The closer to zero means forget and the closer to 1 means keep.

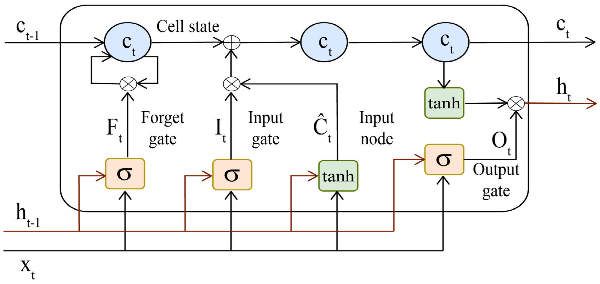


FIGURE 3. LSTM cell diagram.

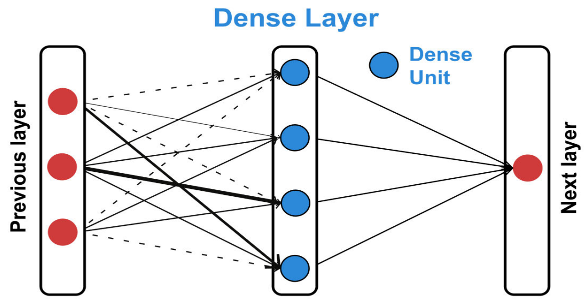


FIGURE 4. Dense layer.

Forget gate:

$$f_t = \sigma\left(w_{fh} \cdot h_{t-1} + w_{fx} \cdot x_t + b_f\right) \tag{1}$$

Input Gate:

$$i_t = \sigma\left(w_{ih} \cdot h_{t-1} + w_{ix} \cdot x_t + b_i\right) \tag{2}$$

Input Node:

$$g_t = \tanh\left(w_{gh} \cdot h_{t-1} + w_{gx} \cdot x_t + b_g\right) \tag{3}$$

$$c_{ti} = i_t \cdot g_t \tag{4}$$

$$c_t = c_{ti} + c_{tf} \tag{5}$$

Output gate:

$$o_t = \sigma\left(w_{oh} \cdot h_{t-1} + w_{ox} \cdot x_t + b_o\right) \tag{6}$$

$$h_t = \tanh(c_t) \cdot o_t \tag{7}$$

In the case of LSTM, the process is characterized by selective memory enhancement. It starts with the current cell state, denoted as c_t. To determine this new cell state, LSTM considers various factors, including the input states x_t and h_{t-1}, which represent the incoming data and context from the previous time step. Crucially, LSTM leverages a forget gate that outputs c_{ti} and a corresponding forget gate signal f_t. These components enable LSTM to decide which information from the previous cell state should be retained and which should be discarded. By adding c_{ti} & f_t the next cell state c_t is obtained. Additionally, LSTM calculates an input gate i_t and an input node g_t using sigmoid and tanh activation functions, respectively, to determine what new information should be added to the cell state. The output state o_t is derived by considering the inputs x_t and h_{t-1}, followed by the next hidden state h_t, which is generated by applying a tanh activation function to the current cell state c_t and combining it with the output gate o_t. This intricate process allows LSTM to selectively update calendar information by fine-tuning the cell state while maintaining the context.

2) DEEP NEURAL NETWORK (DNN)

The DNN part of our model consists of Dense, Activation layers. Dense layers are commonly used to turn input train data into a high-dimensional feature space. It helps in the effective learning of complex features of underlying stock market price patterns from data. The dense layer also applies a linear transformation to the input data by multiplying it by a weight and adding a bias, as shown in equation (8), where w is the weight, x is the input data, and b is the bias [30].

$$y = x_n \cdot w_n + b_n \tag{8}$$

For the output layer, we utilize the SoftMax activation function layer demonstrated in equation (9), which is a popular logistic function that accepts values 0 to 1. The value closest to 1 approaches the final outcome. The z represents the value of neurons derived from the output layer.

$$\mathrm{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{9}$$

The Flatten layer is mostly used to prepare data for the Dense levels that follow. The Flatten layer handles this conversion by "flattening" the output, yielding a one-dimensional representation. We use a linear activation to keep outputs unscaled, optimizes learning with the Adam method, and measures performance with Mean Squared Error (MSE) and Mean Absolute Error (MAE). The model is trained for 100 epochs on batches of 50 examples, with progress tracked and reported. We use Adaptive Moment Estimation (Adam) [31] optimizer as it is a popular optimization algorithm widely used in training deep neural networks. It combines the concepts of momentum and adaptive learning rates to achieve faster convergence during training. The batch size is set to 64. Batch size refers to the number of training examples used in a single iteration or update of the model's parameters during the optimization process. The epoch is set to 100 and ut refers to a single pass through the entire training dataset during the optimization process.

IV. PERFORMANCE ANALYSIS

Based on our analysis, our model performs better in predicting future closing price of stock market in 26 stock market datasets and that ensures an average of R-squared (R2) score of 0.98606, a MAE of 0.0210, and a MSE of 0.00111. To validate this performance, we run our trained model against test data to see if it is sufficiently capable to predict proper labels from unseen data. Table 2 serves as compelling evidence of the robustness of the LSTM-DNN

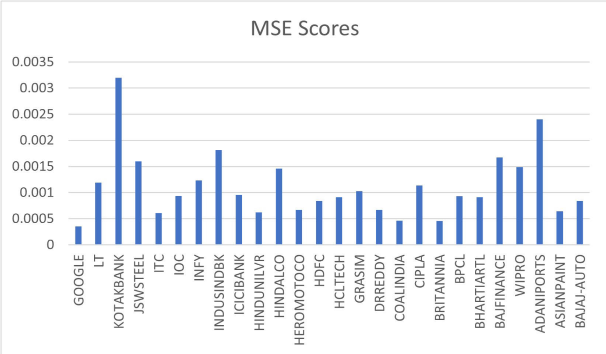


FIGURE 5. MSC score accorss 26 datasets.

model in stock market prediction. The performance metrics displayed in the table, including Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R2) score, and Max Error, demonstrate the model's ability to accurately forecast stock prices across a diverse range of datasets from different companies. The consistently low MSE and MAE values indicate that the model effectively minimizes errors in its predictions, showcasing its precision and reliability in capturing the underlying patterns in the data. Moreover, the high R2 scores across multiple datasets signify that a significant proportion of the variance in stock prices is explained by the model, highlighting its strong predictive capabilities and ability to generalize well to unseen data.

Furthermore, the relatively low Max Error values suggest that the model performs well even in scenarios where the prediction deviates from the actual value, showcasing its resilience to outliers and extreme fluctuations in stock prices. By achieving impressive results across various datasets, the LSTM-DNN model demonstrates its adaptability and effectiveness in handling different market conditions and company-specific trends. This consistency in performance underscores the model's robustness and its potential to provide valuable insights for investors and traders seeking accurate and reliable stock market predictions. The overall aim of these graphs in Fig. 6 and 7 are likely to assess the performance of a stock price prediction model. The closeness of the predicted prices to the actual prices indicates the model's accuracy. In each case, the yellow line (predicted price) attempts to follow the green line (actual price), and the degree of overlap would typically be used to evaluate how well the model is performing.

Table 3 and Fig. 8 presents a comparative performance analysis of various neural network models used in past study to predict stock market trends, specifically using the Asian Paint's stock market dataset. The models compared include a CNN-BiLSTM model, a standard LSTM model, a BiLSTM model, and the proposed model by the authors. The metrics used for comparison are Mean Squared Error (MSE), Mean Absolute Error (MAE), R2 Score, and Maximum Error. The proposed model provides a significant improvement in both MSE and MAE, suggesting it can predict stock prices with lower errors and greater accuracy. The high R2 score demonstrates its superior predictive power and efficiency in handling the dataset. By improving these metrics, the proposed model likely offers a more reliable and robust tool for investors and analysts in predicting stock market trends. This can lead to more informed decision-making and potentially higher investment returns. In Fig. 5, the MSE score of all 26 datasets are shown. Tables 4 - 8 in the provided data showcase the actual versus predicted stock prices for various companies using neural network models. Table 4 details predictions for Google's stock, showing that the model's estimates closely match the actual trading prices, indicating effective predictive accuracy for this dataset. In Table 5, the actual and predicted prices for Asian Paints are also closely aligned, albeit the predictions are slightly higher, demonstrating good model performance. Table 6, which pertains to HDFC Bank, follows a similar trend with predictions modestly overestimating the actual prices. Table 7 for Bharat Petroleum Corporation Limited (BPCL) and Table 8 for Hero MotoCorp exhibit minimal deviations between predicted and actual prices, with the model generally tending to slightly overestimate. Overall, these tables collectively demonstrate the capability of neural network models to generate relatively accurate stock price forecasts, albeit with room for fine-tuning to further minimize the differences between predicted and actual values.

A. PERFORMANCE METRICS

We use Mean Square Error, R-squared score, Mean Absolute Error and Maximum error as performance metrics [29]. The rationale behind using Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared (R2) Score, and Maximum Error metrics to evaluate the model's performance in stock market prediction lies in their complementary nature in assessing different aspects of the model's predictive capabilities. MAE provides a straightforward measure of the average prediction error, offering insights into the model's accuracy in predicting stock prices. MSE, on the other hand, emphasizes the squared errors, giving more weight to larger deviations between predicted and actual values and providing a measure of overall variance. The R2 score indicates how well the model explains the variability in stock prices, offering a measure of the model's goodness of fit. Additionally, the Maximum Error metric highlights the worst-case scenario in terms of prediction accuracy, identifying potential outliers or extreme errors. By considering these metrics together, researchers can gain a comprehensive understanding of the model's performance, including accuracy, variance, goodness of fit, and outlier detection, which collectively contribute to assessing the model's effectiveness in predicting stock market trends accurately and reliably. Let us discuss the performance matrix one by one. Mean Square Error (MSE) is a common metric used to evaluate the accuracy of a predictive model. It is particularly popular in regression analysis, where the goal is to predict a continuous outcome. MSE measures the

TABLE 2. Performance analysis of our model with other datasets.

Dataset	MSE	MAE	R2 Score	Max Error
GOOGLE	0.00035	0.0129	0.974231	0.154305
LT	0.00119	0.0217	0.990054	0.431874
KOTAKBANK	0.0032	0.0281	0.988097	0.671204
JSWSTEEL	0.0016	0.0271	0.991917	0.242372
ITC	0.00061	0.018	0.984111	0.127921
IOC	0.00094	0.0215	0.98566	0.33207
INFY	0.00123	0.02	0.982257	0.610621
INDUSINDBK	0.00182	0.0271	0.988422	0.332332
ICICIBANK	0.00096	0.022	0.988494	0.187998
HINDUNILVR	0.00062	0.0183	0.986331	0.181255
HINDALCO	0.00146	0.0279	0.987865	0.305045
HEROMOTOCO	0.00067	0.0183	0.990799	0.161497
HDFC	0.00084	0.0172	0.982152	0.494104
HCLTECH	0.00091	0.0211	0.989961	0.130313
GRASIM	0.00103	0.0228	0.989638	0.23897
DRREDDY	0.00067	0.0182	0.986934	0.189712
COALINDIA	0.00046	0.016	0.980217	0.079939
CIPLA	0.00114	0.0246	0.979094	0.245481
BRITANNIA	0.000454	0.0155	0.992323	0.107638
BPCL	0.00093	0.0219	0.985405	0.198234
BHARTIARTL	0.000909	0.0208	0.990183	0.252053
BAJFINANCE	0.00167	0.0257	0.991273	0.460553
WIPRO	0.00149	0.0203	0.983221	0.558332
ADANIPORTS	0.0024	0.0237	0.972225	0.967157
ASIANPAINT	0.00064	0.0154	0.98752	0.413545
BAJAJ-AUTO	0.00084	0.0199	0.989429	0.210491
Average	**0.00111**	**0.0210**	**0.98606**	**0.3186**

TABLE 3. Performance analysis of our model with other neural network models in asian paint's stock market dataset.

Model	MSE	MAE	R2 Score	Max Error
CNN-BiSLSTM [5], [23]	0.00428	0.0428	0.9095	0.518524
LSTM [3], [4], [10], [20], [21]	0.00074	0.0163	0.9784	0.442627
BiLSTM [22]	0.00082	0.0163	0.9838	0.466463
Proposed Model	0.00064	0.0154	0.9875	0.413545

TABLE 4. Actual vs predicted in GOOGLE's stock dataset.

Actual Price (USD)	Predicted Price (USD)
675.17	679.44
552.28	551.98
635.54	636.01
618.36	618.65
781.96	783.85
695.94	692.42
726.66	730.74

TABLE 5. Actual vs predicted in ASIAN PAINT's stock dataset.

Actual Price (Rupees)	Predicted Price (Rupees)
985.83	979.97
983.52	983.84
976.14	977.65
964.07	958.79
970.55	974.98
967.45	976.27
967.93	976.54

average squared difference between the predicted values and the actual values. We use this equation (10) to calculate

TABLE 6. Actual vs predicted in HDFC Bank's stock dataset.

Actual Price (Rupees)	Predicted Price (Rupees)
1393.08	1409.82
1391.75	1397.73
1398.90	1396.99
1378.45	1382.08
1381.96	1390.61
1415.06	1419.25
1386.96	1395.12

TABLE 7. Actual vs predicted in BPCL's stock dataset.

Actual Price (Rupees)	Predicted Price (Rupees)
706.68	700.86
713.87	704.17
690.81	702.87
661.17	677.09
662.92	668.85
659.58	650.25
713.66	703.59

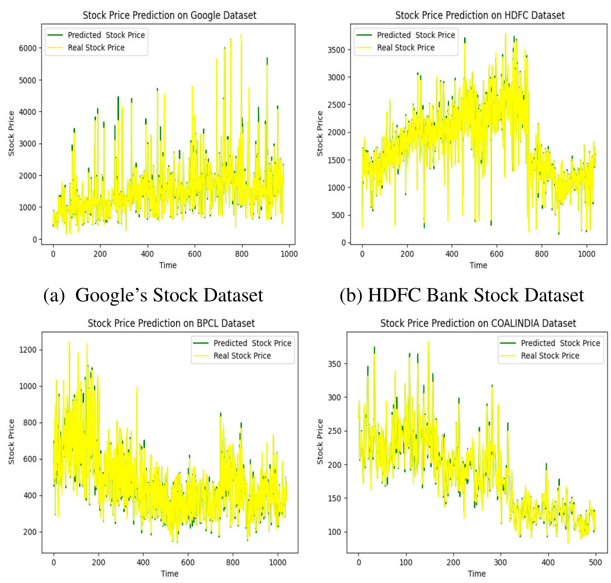
MSE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(actual - predicted)^2 \tag{10}$$

The R-squared (R2) score is a statistical measure that represents the proportion of the variance in the dependent variable (target) that is explained by the independent variables (features) in a regression model. It is a measure of how

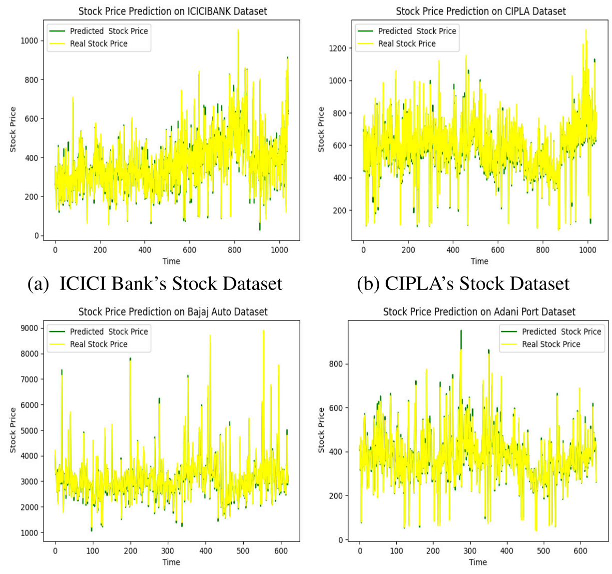
TABLE 8. Actual vs predicted in HERO MOTORS CO's stock dataset.

Actual Price (Rupees)	Predicted Price (Rupees)
3291.20	3280.65
3317.44	3322.68
3169.35	3199.05
3130.11	3148.02
3146.12	3156.70
3187.67	3174.36
3190.29	3200.89



(a) Google's Stock Dataset (b) HDFC Bank Stock Dataset
(c) BPCL's Stock Dataset (d) Coal India's Stock Dataset

FIGURE 6. Actual vs predicted stock price (lr=0.001, batch=100, opt=adam).



(a) ICICI Bank's Stock Dataset (b) CIPLA's Stock Dataset
(c) Bajaj Auto's Stock Dataset (d) Adani Port's Stock Dataset

FIGURE 7. Actual vs predicted stock price (lr=0.001, batch=100, opt=adam).

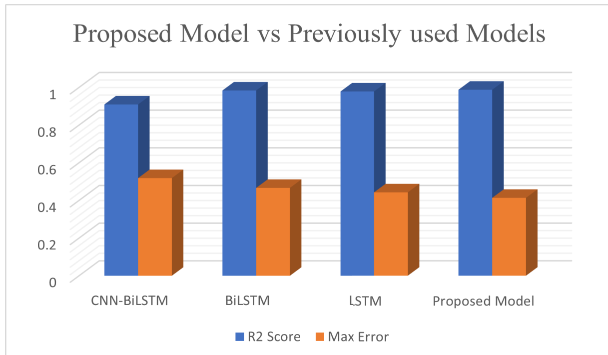


FIGURE 8. R2 score and Max error score (proposed model vs widely used models).

well the independent variables explain the variability of the dependent variable. We calculate R2 score by using equation (11). The R2 score ranges from 0 to 1, where:

- 0 indicates that the model does not explain any of the variability in the target variable.
- 1 indicates that the model perfectly explains the variability in the target variable.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(actual - predicted)^2}{\sum_{i=1}^{n}(actual - \text{mean of actual})^2} \tag{11}$$

MAE, or Mean Absolute Error, is a metric commonly used to evaluate the accuracy of a predictive model, particularly in regression analysis. It measures the average absolute difference between the predicted values and the actual values. The formula for MAE is as follows equation (12):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|actual - predicted| \tag{12}$$

V. UNDERSTANDING THE PROFITABILITY POTENTIAL OF PROPOSED MODEL

The potential profitability of the proposed LSTM-DNN model in stock market prediction can be supported by the performance metrics presented in Table 2 of this paper, which includes Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R2) Score, and Maximum Error. These metrics serve as key indicators of the model's potential profitability by assessing its ability to make precise predictions and minimize errors in forecasting stock prices.

Moreover, a deeper insight into the model's profitability can be gained by examining the actual versus predicted stock prices in Tables 4, 5, 6, 7, and 8. Table 4 illustrates the close alignment between predicted and actual prices for Google's stock dataset, indicating the model's accuracy in forecasting. Similarly, Table 5 shows consistent predictions for Asian Paints' stock, although slightly higher, suggesting good model performance and potential profitability. In Table 6, focusing on HDFC Bank's stock dataset, the model's predictions modestly overestimate actual prices, yet maintain a reasonable level of accuracy, highlighting its profitability potential. Table 7 for Bharat Petroleum Corporation

Limited (BPCL) and Table 8 for Hero MotoCorp exhibit minimal deviations between predicted and actual prices, with the model generally slightly overestimating, further emphasizing its reliability and profitability in stock market forecasting.

The impressive performance metrics in Table 2, such as low MSE and MAE values alongside high R2 scores, indicate the model's capability to generate accurate predictions with minimal deviation from actual stock prices, essential for profitable trading decisions. The model's resilience to outliers and extreme fluctuations, as evidenced by low Maximum Error values, further enhances its profitability potential by ensuring reliable forecasts in dynamic market conditions. By combining the robust performance metrics from Table 2 with the actual versus predicted stock prices in Tables 4-8, stakeholders can gain a comprehensive understanding of the LSTM-DNN model's profitability and effectiveness in guiding investment decisions. The model's accuracy, consistency in predictions, and ability to minimize errors underscore its potential to provide valuable insights for investors and traders seeking profitable opportunities in the stock market.

VI. DEALING WITH MARKET VOLATILITY AND INTRICATE PATTERNS

The LSTM-DNN proposed model effectively addresses market volatility and intricate patterns in stock market prediction through its advanced architecture and training methodology. Specifically, the LSTM component of the model plays a crucial role in capturing and retaining long-term dependencies in the data, allowing the model to remember important information over extended periods. This capability is particularly valuable in the context of market volatility, where sudden and significant price fluctuations occur. By utilizing LSTM networks, the model can effectively learn and adapt to these volatile market conditions, enabling it to make more informed predictions even in the presence of rapid changes.

Additionally, the DNN component of the model excels at identifying complex patterns within the data as described in Section III-C, including subtle trends and correlations that may not be immediately apparent. This aspect is essential for handling intricate patterns in stock market data, where multiple variables and factors can influence price movements. The DNN component's ability to analyze large datasets and extract meaningful features allows the model to uncover hidden relationships and make more accurate predictions based on these intricate patterns.

Furthermore, the integration of LSTM and DNN architectures in a hybrid model combines the strengths of both approaches, enhancing the model's overall predictive capabilities. The LSTM component provides the model with the ability to understand temporal dynamics and long-term dependencies, while the DNN component enhances its pattern recognition and feature extraction capabilities. This combination enables the model to effectively capture the distinct features of market volatility and intricate patterns, leading to more accurate and reliable predictions in stock market forecasting from real life company stock data. This combination allows the model to adapt to volatile market conditions, identify complex patterns in the data, and make informed predictions that account for the intricacies of stock market dynamics.

VII. DISCUSSION

The LSTM-DNN model stands out as a highly effective solution for tackling the intricate challenges involved in predicting stock market movements. This hybrid architecture combines the strengths of LSTM networks, renowned for their capability to capture temporal dependencies and patterns over time, with DNNs, which excel in extracting complex features and patterns from extensive datasets. Stock market data is inherently nonlinear and influenced by numerous factors, including economic indicators, market sentiment, and geopolitical events. LSTM networks are particularly well-suited for this task due to their ability to learn and retain long-term dependencies in sequential data.

In the realm of stock market prediction, where accurate forecasting is crucial for informed decision-making, the LSTM-DNN model's ability to capture these sequential dependencies plays a pivotal role. By discerning subtle trends and seasonal variations in stock prices, the model enhances predictive accuracy and provides valuable insights into market dynamics. This is underscored by its robust performance across 26 real-life company datasets, achieving an average R-squared (R2) score of 0.98606, Mean Absolute Error (MAE) of 0.0210, and Mean Squared Error (MSE) of 0.00111. Such results highlight the model's reliability and effectiveness in handling diverse market conditions and dataset complexities.

Moreover, the LSTM-DNN model addresses key limitations observed in previous research. Unlike traditional methods that often struggle with noise in data or fail to capture long-term trends effectively discussed in Section II-A, this hybrid approach extends prediction timeframes and emphasizes robust data handling techniques. By utilizing the complementary strengths of LSTM networks for sequential pattern recognition and DNNs for feature extraction from large-scale datasets, the model demonstrates superior generalizability and performance consistency across different stocks and market scenarios.

VIII. LIMITATIONS

Despite the promising results of our custom hybrid architecture based on LSTM and DNN model, it is important to acknowledge its limitations. Although the model demonstrates considerable robustness across a wide range of datasets, its performance is still susceptible to variations in the quality and granularity of the input data. For instance, the presence of anomalies in market data, such as missing values, inaccuracies in financial reports, or inconsistencies in historical records can significantly hamper the model's predictive accuracy.

In financial markets, data integrity is most important. Inconsistency or errors in data can arise from various sources, including reporting delays, typographical errors in financial statements, and gaps in data collection. When such issues are present, the model's ability to generate reliable forecasts may be compromised. This is because machine learning models, including our hybrid LSTM-DNN, heavily rely on the assumption that the input data is both accurate and complete. Any deviation from this assumption can lead to erroneous predictions.

Furthermore, the granularity of the data also plays a crucial role. High-frequency trading data, for example, provides a very different landscape compared to end-of-day market summaries. If the model is trained on one type of data but deployed on another, discrepancies in performance may arise. The temporal resolution of input data–whether it is minute-by-minute, hourly, or daily–can affect the model's ability to capture relevant patterns and trends, thereby impacting its overall efficacy.

In conclusion, while our hybrid LSTM-DNN model shows significant potential and robustness in diverse applications, careful consideration must be given to the quality and granularity of the input data. Ensuring high data quality and appropriate temporal resolution is essential for maintaining the model's predictive power and reliability in real-world financial forecasting.

IX. FUTURE WORK

Implementing real-time prediction capabilities and adaptive learning mechanisms for the hybrid LSTM-DNN model involves several critical steps. Firstly, integrating real-time data streams from financial data providers through APIs and employing an event-driven architecture will ensure efficient handling of incoming market data. Developing a robust data preprocessing pipeline will allow for real-time cleaning and normalization. Deploying the model on scalable platforms like TensorFlow Serving or AWS SageMaker will enable low-latency predictions, supported by optimization techniques such as model quantization and hardware acceleration. Incorporating incremental learning and sliding window approaches will facilitate continuous model updates with new data, ensuring adaptability to changing market trends. Continuous monitoring of performance metrics and automated retraining mechanisms will help maintain high predictive accuracy. Additionally, leveraging cloud services and distributed computing frameworks will ensure scalable infrastructure, while real-time dashboards and alert systems will provide valuable insights and timely notifications. Ensuring robust data security and compliance with financial regulations will be essential for ethical and legal use of the predictive models. This comprehensive approach will enable the hybrid model to respond effectively to dynamic market conditions, offering valuable insights for investors and traders.

X. CONCLUSION

This research on enhancing stock market prediction through a hybrid LSTM-DNN based custom architecture represents a significant advancement in the field of financial analytics. By introducing an innovative hybrid model that combines the strengths of LSTM and DNN architectures, the study has demonstrated remarkable adaptability and consistency across diverse datasets and company-specific trends. Through rigorous evaluation using metrics such as MAE, MSE, R2 Score, and Maximum Error, the model has showcased its accuracy, variance explanation, goodness of fit, and outlier resilience, highlighting its robust performance in predicting stock prices. The research not only addresses limitations observed in previous studies but also extends the prediction timeframe to capture longer-term trends, offering valuable insights for investors and traders. By comparing the proposed model with existing neural network approaches and showcasing significant improvements in predictive accuracy, the study underscores the model's potential for guiding profitable investment decisions in dynamic market conditions. Overall, the research contributes to a deeper understanding of predictive modeling in finance, emphasizing the model's adaptability, consistency, and profitability potential in stock market forecasting.

DATASET OBTAINED

1) https://www.kaggle.com/datasets/rohanrao/nifty50-stock-market-data
2) https://www.kaggle.com/datasets/varpit94/google-stock-data

REFERENCES

[1] *Stock Market Prediction*. Accessed: May 10, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Stock_market_prediction

[2] Y. Li, L. Chen, C. Sun, G. Liu, C. Chen, and Y. Zhang, "Accurate stock price forecasting based on deep learning and hierarchical frequency decomposition," *IEEE Access*, vol. 12, pp. 49878–49894, 2024, doi: 10.1109/ACCESS.2024.3384430.

[3] A. Ruke, S. Gaikwad, G. Yadav, A. Buchade, S. Nimbarkar, and A. Sonawane, "Predictive analysis of stock market trends: A machine learning approach," in *Proc. 4th Int. Conf. Data Eng. Commun. Syst. (ICDECS)*, Bangalore, India, Mar. 2024, pp. 1–6, doi: 10.1109/icdecs59733.2023.10503557.

[4] M. Yang and J. Wang, "Adaptability of financial time series prediction based on BiLSTM," *Proc. Comput. Sci.*, vol. 199, pp. 18–25, Jan. 2022, doi: 10.1016/j.procs.2022.01.003.

[5] X. Zheng, "Stock price prediction based on CNN-BiLSTM utilizing sentiment analysis and a two-layer attention mechanism," *Adv. Econ., Manage. Political Sci.*, vol. 47, no. 1, pp. 40–49, Dec. 2023, doi: 10.54254/2754-1169/47/20230369.

[6] Y. Chen, J. Liu, Y. Gao, W. He, H. Li, G. Zhang, and H. Wei, "A new stock market analysis method based on evidential reasoning and hierarchical belief rule base to support investment decision making," *Frontiers Psychol.*, vol. 14, Feb. 2023, Art. no. 1123578, doi: 10.3389/fpsyg.2023.1123578.

[7] Y. Tong, J. Liu, L. Yu, L. Zhang, L. Sun, W. Li, X. Ning, J. Xu, H. Qin, and Q. Cai, "Technology investigation on time series classification and prediction," *PeerJ Comput. Sci.*, vol. 8, p. 982, May 2022, doi: 10.7717/peerj-cs.982.

[8] X. Zhang, "Research on CSI 300 index volatility based on GARCH model," in *Proc. 12th Int. Conf. Intell. Comput. Technol. Autom. (ICICTA)*, Xiangtan, China, Oct. 2019, pp. 77–80, doi: 10.1109/icicta49267.2019.00023.
[9] E. Liu, "Comparison of stock price prediction ability based on GARCH and BP_ANN," in *Proc. 2nd Int. Conf. Comput. Data Sci. (CDS)*, Stanford, CA, USA, Jan. 2021, pp. 90–93, doi: 10.1109/cds52072.2021.00021.
[10] P. S. Sisodia, A. Gupta, Y. Kumar, and G. K. Ameta, "Stock market analysis and prediction for Nifty50 using LSTM deep learning approach," in *Proc. 2nd Int. Conf. Innov. Practices Technol. Manag. (ICIPTM)*, vol. 2, Gautam Buddha Nagar, India, Feb. 2022, pp. 156–161, doi: 10.1109/ICIPTM54933.2022.9754148.
[11] L. Zhao and L. Wang, "Price trend prediction of stock market using outlier data mining algorithm," in *Proc. IEEE 5th Int. Conf. Big Data Cloud Comput.*, Dalian, China, Aug. 2015, pp. 93–98, doi: 10.1109/BDCLOUD.2015.19.
[12] Y. Wang and Y. Wang, "Using social media mining technology to assist in price prediction of stock market," in *Proc. IEEE Int. Conf. Big Data Anal. (ICBDA)*, Hangzhou, China, Mar. 2016, pp. 1–4, doi: 10.1109/ICBDA.2016.7509794.
[13] Z. Zhang, Y. Shen, G. Zhang, Y. Song, and Y. Zhu, "Short-term prediction for opening price of stock market based on self-adapting variant PSO-Elman neural network," in *Proc. 8th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Beijing, China, Nov. 2017, pp. 225–228, doi: 10.1109/ICSESS.2017.8342901.
[14] H. Gunduz, Z. Cataltepe, and Y. Yaslan, "Stock market direction prediction using deep neural networks," in *Proc. 25th Signal Process. Commun. Appl. Conf. (SIU)*, Antalya, Turkey, May 2017, pp. 1–4, doi: 10.1109/SIU.2017.7960512.
[15] M. Faraz, H. Khaloozadeh, and M. Abbasi, "Stock market prediction-by-prediction based on autoencoder long short-term memory networks," in *Proc. 28th Iranian Conf. Electr. Eng. (ICEE)*, Tabriz, Iran, Aug. 2020, pp. 1–5, doi: 10.1109/ICEE50131.2020.9261055.
[16] Y. Liu, Z. Su, H. Li, and Y. Zhang, "An LSTM based classification method for time series trend forecasting," in *Proc. 14th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Xi'an, China, Jun. 2019, pp. 402–406, doi: 10.1109/ICIEA.2019.8833725.
[17] S. K. Sarvepalli, "Deep learning in neural networks: The science behind an artificial brain," 2015, doi: 10.13140/RG.2.2.22512.71682.
[18] R. Vopani. *Nifty-50 Stock Market Data (2000–2021)*. Accessed: Aug. 29, 2023. [Online]. Available: https://www.kaggle.com/datasets/rohanrao/nifty50-stock-market-data
[19] P. Singh, M. Jha, M. Sharaf, M. A. El-Meligy, and T. R. Gadekallu, "Harnessing a hybrid CNN-LSTM model for portfolio performance: A case study on stock selection and optimization," *IEEE Access*, vol. 11, pp. 104000–104015, 2023, doi: 10.1109/access.2023.3317953.
[20] Q. Li, J. Tan, J. Wang, and H. Chen, "A multimodal event-driven LSTM model for stock prediction using online news," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 10, pp. 3323–3337, Oct. 2021, doi: 10.1109/TKDE.2020.2968894.
[21] E. Koo and G. Kim, "A hybrid prediction model integrating GARCH models with a distribution manipulation strategy based on LSTM networks for stock market volatility," *IEEE Access*, vol. 10, pp. 34743–34754, 2022, doi: 10.1109/ACCESS.2022.3163723.
[22] S. Wang, "A stock price prediction method based on BiLSTM and improved transformer," *IEEE Access*, vol. 11, pp. 104211–104223, 2023, doi: 10.1109/ACCESS.2023.3296308.
[23] H. Wang, J. Wang, L. Cao, Y. Li, Q. Sun, and J. Wang, "A stock closing price prediction model based on CNN-BiSLSTM," *Complexity*, vol. 2021, pp. 1–12, Sep. 2021, doi: 10.1155/2021/5360828.
[24] D. S. A. Elminaam, A. E. Tanany, M. A. Salam, and M. A. E. Fattah, "CPSMP_ML: Closing price prediction of stock market using machine learning models," in *Proc. 2nd Int. Mobile, Intell., Ubiquitous Comput. Conf. (MIUCC)*, May 2022, pp. 251–255, doi: 10.1109/miucc55081.2022.9781756.
[25] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock closing price prediction using machine learning techniques," *Proc. Comput. Sci.*, vol. 167, pp. 599–606, Jan. 2020, doi: 10.1016/j.procs.2020.03.326.
[26] T. Gao and Y. Chai, "Improving stock closing price prediction using recurrent neural network and technical indicators," *Neural Comput.*, vol. 30, no. 10, pp. 2833–2854, Oct. 2018, doi: 10.1162/neco_a_01124.
[27] P. Amal, J. Rodrigues, R. R. Naik, K. M. C. Kumar, and P. Shetgaonkar, "Stock market prediction using machine learning—A survey," in *Proc. 3rd Int. Conf. Innov. Technol. (INOCON)*, Bangalore, India, Mar. 2024, pp. 1–7, doi: 10.1109/inocon60754.2024.10511901.
[28] S. Dinushan and M. C. Wijegunasekara, "Machine learning-driven sri Lankan stock market prediction: Harnessing economic indicators and sentiment analysis," in *Proc. 4th Int. Conf. Adv. Res. Comput. (ICARC)*, Belihuloya, Sri Lanka, Feb. 2024, pp. 61–66, doi: 10.1109/icarc61713.2024.10499774.
[29] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning*. Berlin, Germany: Springer, 2010, doi: 10.1007/978-0-387-30164-8.
[30] V. E. Balas, S. S. Roy, D. Sharma, and P. Samui, *Handbook of Deep Learning Applications*. Berlin, Germany: Springer, 2019, doi: 10.1007/978-3-030-11479-4.
[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

KHORSHED ALAM (Member, IEEE) received the B.Sc. degree in CSE from East Delta University. He is currently pursuing the M.Sc. degree in CSE with United International University. He is currently a Software Engineering Enthusiast. He is also a Research Assistant with Independent University, Bangladesh. His passion lies in exploring the possibilities of artificial intelligence, signal processing, human–computer interaction, medtech, and software engineering. He is a driven and dedicated individual who is eager to make a positive impact in the world of technology and contribute his skills and knowledge to the dynamic field of software development and research.

MAHBUBUL HAQ BHUIYAN received the Bachelor of Science degree in computer science and engineering from Independent University, Bangladesh. He is currently pursuing the Master of Science degree in computer science and engineering with United International University.

He is currently an Emerging Scholar in the field of AI and ML, with a keen focus on signal processing and software engineering. He is also an Adjunct Lecturer with the Department of CSE, Independent University, Bangladesh. He is known for his dedication to fostering research skills among undergraduate students. He actively encourages young minds to delve into the world of research, highlighting its importance in the evolving tech landscape. He is deeply involved in mentoring and supervising numerous student-led research projects, facilitating their exploration of innovative ideas in AI, ML, and ed-tech. His approach in the classroom is characterized by a blend of practical insights, a question-based approach alongside theoretical knowledge, aimed at preparing students for the challenges of modern technological advancements.

INJAMAM UL HAQUE was born in Dhaka, Bangladesh, in 2002. He is currently pursuing the B.Sc. degree in computer science with Independent University, Bangladesh. He passionately explores cutting-edge technologies, specializing in image processing, computer vision, AI for healthcare, quantum machine learning, and deep learning. In image processing, he manipulates visual data to advance our understanding of images, while his work in computer vision aims to bridge the gap between computers and human perception. Committed to societal betterment, he leverages AI for healthcare to revolutionize medical diagnostics and treatment. In quantum machine learning, he explores quantum computing's potential to enhance machine learning algorithms, showcasing forward-thinking. In deep learning, he contributes to unraveling neural networks' intricacies, staying dedicated to technological innovation. His diverse research interests include significant impact as he navigates the dynamic world of technology in pursuit of knowledge and advancement.

MD FAHAD MONIR (Member, IEEE) received the B.Eng. degree (Hons.) in communication engineering from International Islamic University Malaysia (IIUM), Malaysia, in 2014, the M.Sc. degree in information technology from the KTH Royal Institute of Technology, Sweden, in 2016, and the M.Eng. degree in communication engineering from the University of Trento, Italy, in 2017. From 2017 to 2018, he was a Research Assistant in exploiting wireless links diversity in software-defined 802.11-based enterprise WLAN with Fondazione Bruno Kessler (FBK), Italy. From 2018 to 2019, he was a full-time Lecturer with American International University of Bangladesh. Since 2019, he has been a Senior Lecturer with the Department of Computer Science and Engineering, Independent University, Bangladesh. His current research interest includes software-defined networking.

TAREM AHMED (Senior Member, IEEE) received the Graduate degree (summa cum laude) in physics and in economics from the Middlebury College, USA, master's degree in electrical engineering from the Ivy League University of Pennsylvania, USA, and the Doctorate degree in computer science from International Islamic University Malaysia, Malaysia. During various stages in his career, he has worked in industrial product development in the silicon valley area of CA, USA, and in academic research with McGill University, Canada, and ETH Zurich, Switzerland, before returning to his native city of Dhaka, Bangladesh. He has presented in research workshops at various international universities, such as the University of Naples Federico II, Italy, and École Normale Supérieure (ENS), France, and was an Erasmus+ Visiting Scholar with Middle East Technical University, Turkey.