IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 19 August 2023, accepted 17 September 2023, date of publication 22 September 2023, date of current version 27 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3317953

RESEARCH ARTICLE

Harnessing a Hybrid CNN-LSTM Model for Portfolio Performance: A Case Study on Stock Selection and Optimization

PRIYA SINGH[1], MANOJ JHA[1], MOHAMED SHARAF[2], MOHAMMED A. EL-MELIGY[2], AND THIPPA REDDY GADEKALLU[3,4]

[1]Department of Mathematics, Bioinformatics and Computer Applications, Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh 462003, India
[2]Department of Industrial Engineering, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia
[3]Zhongda Group, Haiyan, Jiaxing, Zhejiang 314312, China
[4]College of Information Science and Engineering, Jiaxing University, Jiaxing 314001, China

Corresponding author: Priya Singh (193104004@stu.manit.ac.in)

This work was supported by King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSPD2023R704.

ABSTRACT Portfolio theory underpins portfolio management, a much-researched yet uncharted field. This research suggests a collective framework combined with the essence of deep learning for stock selection through prediction and optimal portfolio formation through the mean-variance (MV) model. The CNN-LSTM model, proposed in Stage I blends the benefits of the convolutional neural network (CNN) and the long-short-term memory network (LSTM). The model combines feature extraction and sequential learning about temporal data fluctuations. The experiment considers thirteen input features, combining fundamental market data and technical indicators to capture the nuances of the wildly fluctuating stock market data. The input data sample of 21 stocks was collected from the National Stock Exchange (NSE) of India from January 2005 to December 2021, spanning two significant market crashes. Thus, the sample makes it possible to catch subtle market shifts for model execution. The shortlisted stocks with high potential returns are advanced to Stage II for optimal stock allocation using the MV model. The proposed hybrid CNN-LSTM outperformed the single models, i.e., CNN and LSTM, per the six-performance metrics and advocated by the 10-fold cross-validation technique. Furthermore, the statistical significance of the model is established using non-parametric tests followed by post hoc analysis. In addition, this method is validated by comparing the proposed model to four baseline strategies and relevant pieces of research, which it considerably outperforms in terms of cumulative return per year, Sharpe ratio, and average return to risk with and without transaction cost. These findings highlight the effectiveness of the hybrid CNN-LSTM approach in stock selection and portfolio optimization.

INDEX TERMS Deep learning, CNN-LSTM, mean-variance, asset selection, portfolio optimization.

I. INTRODUCTION

Portfolio selection involves creating a profitable portfolio. Due to return unpredictability, choosing assets is hard. A portfolio selection issue seeks optimal stock proportions to create a portfolio that meets investors' preferences, assuming they want to balance return and risk. The portfolios in the efficient set have the highest level of anticipated return for each degree of risk. As a result, the investor's risk aversion, or attitude toward risk, will determine the best portfolio to choose [1], [2], [3], [4].

The conventional mean-variance portfolio theory, which dates back to Markowitz's key study from 1952 [5], helped the financial viewpoint recognize the value of diversity. A prior selection of assets is vital for portfolio management since the anticipated return on an investment is a key aspect of

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

the portfolio optimization process [6]. Applying complicated portfolio optimization algorithms is not beneficial without high-quality asset input [1], [2], [7], [8]. However, few studies focus on the asset selection process before creating a portfolio. Stock forecasting is among the most challenging time series issues due to the stock market's multi-noise, nonlinearity, high frequency, and chaos. However, because of its significance, stock forecasting continues to get more and more attention from academics and investors. In this context, a significant amount of research has attempted to employ Artificial Intelligence and Machine Learning for the forecast of the returns and volatilities of the stock market [7], [9], [10], [11], [12], [13]. This is due to the advanced computer processing capacity and abundant data available. Early statistical and machine learning techniques are not the best for learning and preserving financial time-series data over a prolonged period [1], [12], [14]. The field arose as a result of deep learning's capacity to acquire intricate nonlinear mapping and self-adaptation mechanisms that aid in the identification of both latent patterns and underlying data dynamics [1]. Recent research has focused on several machine learning & deep learning models [15], including the multilayer perceptron (MLP) [8], convolutional neural network (CNN) [16], recurrent neural network (RNN), and long-short term memory (LSTM) neural network [2], [17], [18]. Also, various aspects were studied pertaining to advancements in portfolio optimization, viz., Reinforcement learning for dynamically adjusting portfolio weights in response to changing market conditions, which involves an agent learning optimal actions through trial and error. These methods aim to maximize portfolio returns while managing risk [19], [20], graph theory [21], credibility theory account for uncertainty and incomplete information when historical data is limited or unreliable [22] etc.

Fischer and Krauss [7] utilized LSTM neural networks to forecast the direction of the S&P 500's component stocks from 1992 to 2015. Portfolio models based on memory-free classification (i.e., Random Forest (RF), Deep Neural Network (DNN), and Logistic Regression (LR) were shown to be inferior to those based on LSTM neural networks. In Sezer and Ozbayoglu [23], a novel algorithmic trading model called CNN-TA was introduced; it makes use of a two-dimensional convolutional neural network with image processing features. It uses 15 separate technical indicators, each with its own set of parameters, to render two-dimensional representations of financial time series. The results show that the trained model outperforms other conventional trading strategies regarding stock and ETF performance. Chen and He [24] suggested a deep learning strategy based on CNN for predicting the stock market price movement in China. The input of the structure was the internet-sourced open price, high price, low price, close price, and volume of stock. As the results have demonstrated, a deep learning system based on CNN can reasonably anticipate the movement of Chinese stock prices. Paiva et al. [1] designed a revolutionary decision-making model for day trading on the stock market by fusing the support vector machine (SVM) and MV models for portfolio selection. Two additional models, SVM+ Equally weighted (1/N) and Random+ MV, were compared to the suggested model. The suggested model outperformed the competition in an experimental evaluation using data from the Ibovespa stock market. Komori [25] developed a trading system by training a CNN on images of 2D technical candlestick charts. In this article, the CNN model structure is used from Inception v3. This research uses CNN to examine the performance of the S&P 500 index between January 1, 1985, and June 30, 2020. Yu et al. [26] advanced six portfolio optimization techniques, i.e. mean-variance, mean absolute deviation, downside risk, linearized value-at-risk, conditional value-at-risk, and omega models, using a combination of autoregressive integrated moving average (ARIMA) estimates. Before developing these portfolio optimization methods, they employed the ARIMA model to forecast future stock returns. The extended MV and omega models with ARIMA prediction were shown to be the most effective in the experiments. Ta et al. [27] used LSTM neural networks and the equal-weighted approach, Monte Carlo simulation, and MV model to construct portfolios. The prediction accuracy of LSTM was shown to be greater in the experiments than that of linear regression and support vector machines. Wang et al. [2] employed an LSTM neural network for stock selection and an MV for portfolio optimization. Here, the focus was constructing an MV portfolio model using just k out of the full stock set. They employed MV for portfolio optimization after comparing LSTM neural network with SVM, RF, and ARIMA models in the stock selection. Results from their experiments validated the superiority of their suggested model to its competitors. Bhandari et al. [28] predicted the next day's closing price of the S&P 500 index using an LSTM architecture. Root Mean Square Error, Mean Absolute Percentage Error, and the Correlation Coefficient are used to evaluate the performance of both single-layer and multilayer LSTM models. The experimental findings reveal that, compared to multilayer LSTM models, the single-layer LSTM model gives a better fit and higher prediction accuracy. Habbab et al. [29] showed the value of integrating Real Estate Investment Trusts (REITs) in mixed-asset portfolios and conducted comprehensive research spanning 456 portfolios. To achieve superior performance over a global minimum variance portfolio, they developed a genetic algorithm-based strategy to optimize the Sharpe ratio of the portfolios. Sisodia et al. [30] designed and devised an LSTM algorithm, a Deep Learning (DL) model. From December 10, 2011, to December 10, 2021, ten years of historical data for the NIFTY 50 index from India's National Stock Exchange (NSE) were selected. The proposed model was quite reliable, with an accuracy of 83.88 percent. Aksan et al. [31] showed two of the most famous hybrid deep learning (HDL) models based on a hybridization of CNN and LSTM that can predict the power flow in the examined network cluster. A case study of the High-Voltage Subnet in North-East Germany was used to train two separate models, CNN-LSTM and LSTM-CNN, with vastly different datasets in terms of size and included parameters. Xu et al. [32] designed the SK-GCN model, which uses a graph convolutional neural network to sort stocks into several groups. This model uses two convolutional layers and activation functions to categorize stocks using external nodes and short text categorization. They crawled all stocks listed on the GEM of the Oriental Fortune website to create the dataset and obtained an accuracy of 83.04% and a macro-F1 value of 0.8303 under small sample training. Cui et al. [33] suggested a multi-scale convolutional neural feature extraction network (MS-CNN) for stock data, which can better find features of stock trends and help make better choices. The network architecture was motivated by the way humans trade stocks, wherein they take into account all available information, including the open, the close, and the number of trades, before reaching a conclusion.

Similarly, predictions of the securities market have been studied using a variety of deep learning benchmark algorithms [8], [12], [34], [35]. The benefits of various models are clear. For instance, CNN is strong at processing and extracting data with spatial dimension, while RNN and LSTM are successful for time series data, and ARIMA is good at processing linear data. However, in practice, predicting problems is usually complicated and has a variety of features. Consequently, the development of hybrid models was unavoidable. For example, Islam et al. [16] presented a deep learning strategy for autonomously diagnosing COVID-19 from X-ray pictures by combining a CNN with an LSTM. Kim and Kim [36] suggested a technique to forecast stock prices using information from stock time series and stock chart pictures. This model is called the feature fusion LSTM-CNN model. In Lu et al. [37], the authors suggested a CNN-LSTM-based stock price forecasting system. Eight features were examined for data input: open, high, low, close, volume, turnover, ups and downs, and change. Ahmed et al. [38] uses a fusion of poly-linear regression, LSTM, and data augmentation to forecast time series. Thus, Poly-linear Regression with Augmented Long Short-Term Memory Neural Network (PLR-ALSTM-NN). The model can effectively anticipate financial markets in the future.

In the prior field of research, numerous possibilities for development require focused attention. Many studies in this area solely rely on historical datasets, such as Open-High-Low-Close (OHLC) data, which fail to capture the dynamic nature of the market, influenced by various factors. Evaluating the performance of prediction models is of utmost importance to prove their reliability in capturing the intrinsic nature of the market data along with the future price movement assessment. Moreover, researchers often overlook the importance of statistical significance as an add-on to the reliability of the prediction models. In deep learning model evaluation, k-fold cross-validation [39], [40] is commonly used. Every portion of the data is used to quantify performance and evaluate the prediction model in each fold of cross-validation [41]. Some research attempted to exploit the data type to explore better performance possibilities, which leads to information loss. To address these gaps, this study proposes a hybrid deep learning approach to anticipate stock prices using numerical data, specifically fundamental and technical indicators of stock closing prices. Numerical data provides a clearer link to stock prices and reduces the risk of information loss associated with changing data types. The study emphasizes the importance of statistical significance and the inclusion of multiple performance metrics in evaluating prediction models. This indicates that the research will go beyond simple accuracy measures and consider various statistical techniques and metrics to assess the effectiveness of the proposed hybrid deep learning approach. The contributions of this research can be summarized as follows:

- To extract rich and crucial information from data, a deep-learning hybrid model is suggested that combines the feature extraction capabilities of CNN with the sequential learning capability of LSTM while capturing key long-term dependencies. Unlike conventional prediction models, which typically only use one deep learning module to collect data.
- To enhance the efficacy of the prediction-based models by incorporating a combination of fundamental and technical input features, depicting better insight into the noisy stock price data. Top k stocks with high predicted returns are shortlisted from the sample of 21 stocks to form a portfolio.
- The performance of the proposed system is measured in detail experimentally using six performance measures, including Mean absolute error (MAE), Root mean squared error (RMSE), Mean absolute percentage error (MAPE), R2, Maximum error (Max Error), and Meadian absolute error (MedAE), as well as the addition of thirteen feature inputs.
- Statistical significance of the proposed hybrid model over comparison models adds scientific rigor to the research. Additionally, 10-fold cross-validation advocates the generalizability and credibility of the proposed model.
- To prevent unrealistic solutions, we impose floor and ceiling constraints on the capital that may be invested in a portfolio asset.

This paper is structured as follows for the remainder: In Section II, we will go through several relevant models and proposed methods. In Section III, we analyze the results of our experiments and describe how we got there in greater depth. In Section IV, we will conclude with some discussion.

II. MODELS AND METHODS

A. CONVOLUTIONAL NEURAL NETWORK MODEL

CNN proposed by Lecun et.al [42], is a feed-forward neural network. In addition to the input, hidden, and output layers that comprise the standard neural network design,

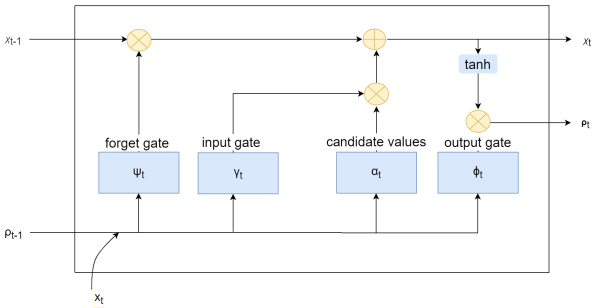


FIGURE 1. A typical LSTM memory cell.

a CNN consists primarily of the convolution layer and the pooling layer. Multiple convolution kernels are stored in each convolution layer, and their calculation formula is displayed below in Eq. 11. The convolution layer extracts data features; however, the feature dimensions are quite high, so a pooling layer is added after the convolution layer to minimize the feature dimension provided by Eq. 13. The fully connected layer is implemented as a classifier to make a judgment using the features learned in the convolutional and pooling layers. CNNs excel at image classification, object recognition, and analysis of medical images [43]. CNNs take local features from high-layer inputs and transfer them to lower layers to learn more complex characteristics. In the context of financial timeseries data, CNNs have the ability to recognise intricate and nonlinear patterns in data. This is especially crucial when stock prices are affected by a wide range of variables and display complex patterns of behaviour.

B. LONG SHORT TERM MEMORY MODEL

An improved RNN, or sequential network, called LSTM [44] enables information to endure. Using memory cells, LSTM can handle the vanishing gradient problem concerning RNN. This structure comprises an input layer, a hidden layer, a cell state, and an output layer. The essential element of the LSTM design is the cell state, which passes through the chain with only linear interaction, preserving the information flow. The LSTM gate mechanism deletes or alters cell state information. Sigmoid, hyperbolic tangent, and point-wise multiplication layers transfer information selectively.

There are three gates that constitute the LSTM:

- Input gate: It adds data to the state of the cell.
- Forget Gate: It deletes data that is no longer needed by the model.
- Output Gate: It decides what data is displayed.

The architecture of an LSTM network, which is intended to mimic sequential input, is seen in Figure. 1. Gate activities at time t are depicted in Equations 15 to 20.

C. PORTFOLIO MATHEMATICAL MODEL: MEAN-VARIANCE MODEL

Modern portfolio theory (MPT) has its origins in the MV model introduced by Markowitz [5] to address the challenge of optimum portfolio selection. Investment returns and risks are quantified through the use of expected return and variance in this method. Using a framework for multi-objective optimization, the MV model may be described. Markowitz's proposed approach does not provide a cutoff for when an investment should be made. The investor can then be supplied with a collection of MV combinations that satisfy the specified risk–return tradeoffs. This collection of ideal options is referred to as the "efficient frontier of investments." The model is officially described by the following equations 1 & 2. Additionally, a lower and upper limit on asset allocation is imposed in the classical MV model [45].

$$
\begin{aligned}
&Minimize \ \sum_{i=1}^{N}\sum_{j=1}^{N} s_i s_j \sigma_{ij}\\
&Maximize \ \sum_{i=1}^{N} s_i \mu_i\\
&\text{subject to,}\\
&\qquad \sum_{i=1}^{N} s_i = 1\\
&\qquad l_i \leq s_i \forall i = 1, 2, 3, \ldots N,\\
&\qquad s_i \leq u_i \forall i = 1, 2, 3, \ldots N
\end{aligned} \tag{1}
$$

A mono-objective formulation (Eq. 2) can be used to reach the same set of ideal assets [46]. Following Paiva et al. [1] and Wang et al. [2], a variable that indicates the investor's risk aversion (λ) is added to the model to characterize their risky investing behavior:

$$
\begin{aligned}
&Minimize \ \lambda\Big[\sum_{i=1}^{N}\sum_{j=1}^{N} s_i s_j \sigma_{ij}\Big] - (1-\lambda)\Big[\sum_{i=1}^{N} s_i \mu_i\Big]\\
&\text{subject to,}\\
&\qquad \sum_{i=1}^{N} s_i = 1\\
&\qquad l_i \leq s_i \leq u_i \forall i = 1, 2, 3, \ldots N
\end{aligned} \tag{2}
$$

where λ is the coefficient of risk aversion, s_i and s_j stand for the portfolio's initial investment of assets i and j, respectively, with l_i and u_i as lower and upper bound for the initial investment, and σ_{ij} indicates the covariance between the two, while μ_i represents the expected return on the asset i. The risk aversion coefficient falls anywhere between zero and one. When $\lambda = 0$, the investor is extremely risk averse and focuses solely on maximizing return. When $\lambda = 1$, on the other hand, investors focus solely on reducing risk, regardless of potential reward. The optimal value strikes a middle ground between these extremes, maximizing projected return while minimizing risk. Therefore, investors may construct a suitable portfolio by selecting one of these options that best suits their risk tolerance [47]. Since there are no risk-free

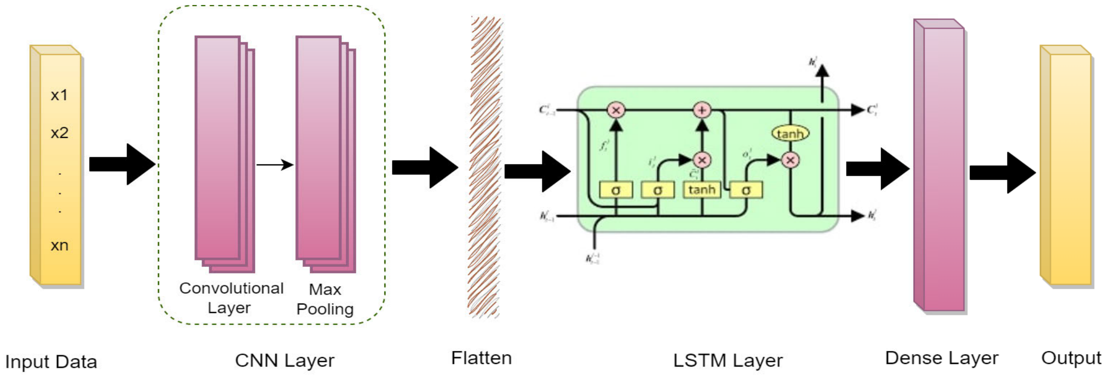


FIGURE 2. **Hybrid CNN-LSTM model.**

assets in the aforementioned formulation, only risky assets are used to build the portfolio.

D. PROPOSED MODEL: CNN-LSTM + MV

A stock forecasting hybrid model named CNN-LSTM is constructed for prediction-based asset selection based on the properties of CNN and LSTM, as seen in Figure 2. Out of a sample of 21 stocks, top k stocks with high predicted returns are shortlisted through CNN-LSTM model to further formulate an optimal portfolio through MV model. Since CNNs can handle numerous channels of input data, the model may learn complicated connections between different forms of data and price changes (such as OHLC prices, technical indicators, market sentiment, etc.). It can efficiently process all of these channels, gleaning useful characteristics from each one to feed into the LSTM's learning process. Combining the sequential nature of LSTM with the pattern-finding abilities of CNNs is a powerful combination. LSTMs are able to process these characteristics hierarchically to capture longer-term dependencies, focusing on more important and meaningful information, whereas CNNs are able to extract low-level features in the initial layers (such as recognising local pricing trends). This can lead to increased model robustness.

To examine the model performance and suitability over the single models, a basic topology pertaining to the parameters and no. of layers is incorporated. For better investment choices, this research proposes a system called CNN-LSTM+MV that combines deep learning's strengths in time-series forecasting with the MV model's proficiency in portfolio optimization. Our model consists of a two-part process. First, the future returns of the sample stocks are forecast using the CNN-LSTM approach. The top stocks will go on to the next phase once all the anticipated outcomes are sorted in descending order. For each stock that has been nominated, the capital allocation proportion will be determined using Markowitz's MV model in the second step. The architecture of the proposed model is given in Figure 3. The comprehensive description and computations of the model are provided below.

TABLE 1. **The stock exchange codes.**

S.No.	Stocks	S.No.	Stocks	S.No.	Stocks
S1	500875	S8	500180	S15	532488
S2	500770	S9	500331	S16	532155
S3	532540	S10	500510	S17	500043
S4	509480	S11	532454	S18	500800
S5	532174	S12	531642	S19	532522
S6	500112	S13	532689	S20	532281
S7	500247	S14	532538	S21	532514

1) DATA

This study utilizes daily stock data from NSE India of 21 stocks between January 2005 and December 2021, spanning sixteen years. The exchange codes of the sample stocks for the research work are presented in Table 1. The bulk of relevant research has been completed within the past 15 years, or fewer [1], [24], [48]. Our 16-year sample period should be regarded long enough to yield statistically meaningful results for the two major crashes we examined, i.e., the financial crisis of 2007-2008 and the Covid-19 crisis of 2020, which shed light on the volatility and unpredictability of the financial market.

2) INPUT VARIABLE SELECTION

For problems involving timeseries prediction, the choice of input variables is crucial. A forecast model that takes into account only one attribute may not yield reliable predictions. The effectiveness of forecasting algorithms may be greatly improved by financial aspects, such as fundamental and technical indicators that examine the pattern and value of stocks [49]. According to prior research, technical indicators are useful tools to characterize and represent the actual market condition. For example, Chen and Hao [24] suggests that the stock market may be understood in light of the relationships between the exponential moving average (EMA), the relative strength index (RSI), and the momentum index (MoM). Wang et al. [2] proposed true range (TR), average true range (ATR), MoM, and RSI as useful stock price forecasting indicators. The efficacy of stock price forecasting is enhanced by the technical indicators, which indicate market behavior through the rise, drop, and trend of stock prices.

Due to the time series nature of the dataset, it is crucial to take into account the dynamic between the features and the outcome variable over time. In this way, after considering the perspectives of domain papers, we select features using a common filtering technique, Pearson's correlation coefficient [28], [29]. More specifically, characteristics with a positive correlation with the target variable and a correlation value more significant than a threshold value are chosen using Pearson's correlation coefficient. The formula for the Pearson correlation coefficient, r, is as follows:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y}))^2}} \tag{3}$$

where n is the sample size, x_i and y_i are the sample points indexed by i, and $\bar{x}$, $\bar{y}$ represent the sample mean.

In the presented research, five fundamental features comprising of historical data viz., Open price(O), Close price(C), High price(H), Low price (L), Adj. Close (AC) are used along with eight technical indicators. The formulae of the indicators used in the research presented are discussed below briefly:

- Simple Moving Average (period or n=12): SMA is employed to assist in smoothing out price data, hence lowering the noise.

$$SMA = \frac{1}{n}\Sigma_{i=1}^{n} C_i \tag{4}$$

- Exponential Moving Average (n=14): EMA is a kind of moving average exhibiting certain characteristics of the SMA. It places greater emphasis on more recent data. Stock in an uptrend has a rising moving average, whereas a downtrend has a falling one.

$$EMA = (C_i - EMA_{i-1}) \times multiplier + EMA_{i-1}, \tag{5}$$

where multiplier $= \frac{2}{n+1}$

- Relative Strenght Index (n=14): The RSI is shown as an oscillator with a 0–100 scale. An asset is often regarded as overbought when it is above 70 and oversold when it is below 30.

$$RSI = 100 - \frac{100}{1 + \frac{AverageGain}{AverageLoss}} \tag{6}$$

- Rate of Change(n=12): Momentum, a pure momentum oscillator, quantifies the percent price change from one period to the next. Higher ROC values indicate overbuying and lower ROC indicates overselling.

$$ROC = \frac{C_i - C_{i-n}}{C_{i-n}} \times 100 \tag{7}$$

- True Range: TR is a stock's maximum price range. It is the largest of the following-

$|H_i - L_i|$
$|C_{i-1} - H_i|$
$|C_{i-1} - L_i|$

- Average True Range (n=14): ATR measures market volatility by dissecting an asset price's whole range during a time, including gaps.

$$ATR_i = \frac{[(ATR_{i-1} \times (n-1)) + TR_i]}{n} \tag{8}$$

- Momentum Index (n=10): It is a momentum indicator that assesses a security's price or volume acceleration.

$$MoM = (C_i - C_{i-1}) \tag{9}$$

- Commodity Channel Index (n=20): Trend indicator CCI compares a security's price movement to its average price change. Prices over average suggest strength. Prices below their average imply weakness.

$$CCI = \frac{[Tp - SMA_{Tp}]}{0.015 \times MeanDeviation_{Tp}} \tag{10}$$

Lambert adjusted the constant to 0.015 such that around 70 to 80% of CCI readings would fall within the range of −100 to +100.

where, Tp (Typical price) $= \frac{(High+Low+Close)}{3}$; i corresponds to the current price and $i-1$ corresponds to the previous price and so on.

3) ASSET SELECTION: CNN-LSTM

In a hybrid model with an LSTM backend, a CNN model is utilized to interpret input subsequences that are then sent as a sequence to an LSTM model for interpretation. The name for this hybrid model is CNN-LSTM. At this stage, the converted data is a two-dimensional array [samples, features]. CNN-LSTM model expects tensor input data shaped as [samples, subsequences, timesteps, and features]. Therefore, the actions required to make the input data compliant with the model are evaluated. Before doing anything else, we made sure to preserve the chronological sequence of the data by dividing it into a 75%-25% split. Furthermore, we split the training data into 80%-20%, with the former for training & the latter 20% (representing 15% of the total data) utilized for validation during hyperparameter tuning. Therefore, the training, testing, and validation ratio is 60:25:15 After all the training data has been accounted for, the final models are fitted using the best possible hyperparameter values. The test results are then summarised with the performance scores. This deep learning model configuration consists of 06 distinct layers: a one-dimensional (1D)-convolutional layer, a max pooling layer, a dropout layer, a flattened layer, an LSTM layer, and a dense layer, as shown in Figure 2. The traits and variables employed in the suggested hybrid deep learning model are presented in Table 3. A set of kernels [43] are included in the convolutional layer to create a tensor of feature mappings. The convolutional layer's functioning is described as follows:

$$f_i = \zeta\left(\Sigma_{j=1}^{n} Conv(f_j, k_{ij}) + b_i\right) \tag{11}$$

where f_j represents the j^{th} feature map; f_i represents i^{th} feature map; k_{ij} kernel; b_i bias; and n is the no. of feature map.

The rectified linear unit (ReLU) layer is utilized to increase nonlinearity in feature maps [50]. ReLU determines activation by keeping a threshold input of zero. The following is how it is stated mathematically:

$$\zeta = max(0, f) \tag{12}$$

After the convolution layer, a max pooling layer is used to reduce the filter maps to the most important features. To minimize the number of parameters, the pooling layer does a down-sampling of a certain input dimension. The most popular technique, max pooling, yields the greatest value in an input area. The procedure is performed by-

$$maxf_i^t = max(maxf_i^{\tilde{t}} : t \leq \tilde{t} \leq t + p(d \times s)) \tag{13}$$

where, f_i^t & $f_i^{\tilde{t}}$ represent t^{th} neuron of i^{th} featre map before and after maxpooling layer. $p(d \times s)$ represents pooling window of dimension of size d and stride s. The presented work pooling window is of size 1×1.

The flattening layer prior to the LSTM layer is included, which functions as a bridge since it facilitates the transition from the convolutional layers, which capture local features, to the LSTM layers, which capture temporal dependencies. For the model to learn and model the time-series temporal linkages and dependencies, the flattened feature vector is fed into the LSTM layers as an input. To avoid the suggested deep model from excessively fitting the data, a dropout layer is included in the architecture. The effect of dropout on the units is defined by the equation below.

$$f_i = \zeta\left(\Sigma_{n>1}\Sigma_{j=1}^{n}(Conv(f_j, k_{ij}) + b_i)\delta_i\right) \tag{14}$$

where δ_i is a probabilistic bernoulli variable ($0 \leq \delta_i \leq 1$). In this research work, δ_i is taken to be 0.2.

After that, an LSTM model is used to interpret the input sequence as read by the CNN model and generate a prediction. Time-series LSTM is an architecture, as shown in Figure 1 built to model sequential input. We considered a single-layer LSTM model since, compared to multilayer LSTM models, they offer a better fit and more reliable predictions [28]. Specifically, the functions of the four gates, the output, change, input, and forget gates, are illustrated in real-time.

$$lstm_{cell} = \{\gamma_t, \psi_t, \phi_t\}$$

The respective procedures involved in the gates are given below in equations 15 to 20:

$$\gamma_t = \sigma\{w_\gamma(\rho_{t-1}, f_i^t) + b_\gamma\} \tag{15}$$

$$\psi_t = \sigma\{w_\psi(\rho_{t-1}, f_i^t) + b_\psi\} \tag{16}$$

$$\chi_t = tanh\{w_\chi(\rho_{t-1}, f_i^t) + b_\chi\} \tag{17}$$

$$\alpha_t = \psi_t\alpha_{t-1} + \gamma_t\chi_t \tag{18}$$

$$\phi_t = \sigma\{w_\phi(\rho_{t-1}, f_i^t) + b_\phi\} \tag{19}$$

$$\rho_t = \phi_t tanh(\alpha_t) \tag{20}$$

where γ_t represents the input gate, ψ_t is the forget gate, χ_t is the current cell state, α_t is candidate value, ϕ_t is output and, ρ_t is hidden state of the LSTM cell for timestep t. The hybrid CNN-LSTM model for multivariate stock price prediction ties everything together.

4) PROCESS OF OPTIMAL PORTFOLIO FORMATION

The second step is to determine which portfolio has the highest risk-adjusted return potential, or capital allocation percentage, for each asset. Therefore, it makes sense for investors to choose portfolios with either a lower risk level and stable anticipated returns or a greater risk level and higher expected return [5], [51]. And to continue with this stage, Markowitz's MV model (II-C) will be applied. It must be stressed that the proposed model does not take into consideration investors' risk preferences or risk-free assets; hence the portfolios contain only high-risk investments. In order to evaluate the efficacy of the recommended risk-return portfolio models, we utilize the realized portfolio return to compute the Sharpe ratio of the portfolios. This is the formula for the Sharpe ratio (SR):

$$SR = \frac{R_p - R_f}{Sd} \tag{21}$$

where, R_p=Return of the portfolio, R_f=Risk-free Rate and, Sd=Standard deviation of Portfolio's Excess return. The portfolio with the lowest variance will ultimately receive the resources that are available. As a result, when assets and investment allocations are verified at the opening of the next trading day, allocations can be made. During the course of the investment day, we will examine the top ten assets. Figure 3 displays the step-by-step process of the recommended method.

5) BASELINE STRATEGIES FOR PORTFOLIO FORMATION

These were constructed using the research model suggested in the preceding section and used to assess the effectiveness of this approach and its modifications.

1) CNN + MV: The architecture of this type of model is identical to the suggested structure of the CNN-LSTM+MV model. The primary aim is to determine if the optimal portfolio may be formed differently depending on the forecast outcomes of asset return. In particular, in the first step, the return on assets in $t + 1$ will be forecasted using the CNN approach, and in the second stage, assets with greater future returns will be selected. It is important to note that the number of assets chosen must exactly match the one indicated in the proposed model. Markowitz's MV approach for optimizing portfolios continues in its second stage.
2) LSTM + MV: The LSTM + MV model differs from the aforementioned baseline CNN + MV model and the proposed model in terms of the asset preselection phase. When it comes to asset preselection, the LSTM model specifies the same amount of assets as the CNN-LSTM + MV model. The Markowitz optimization of

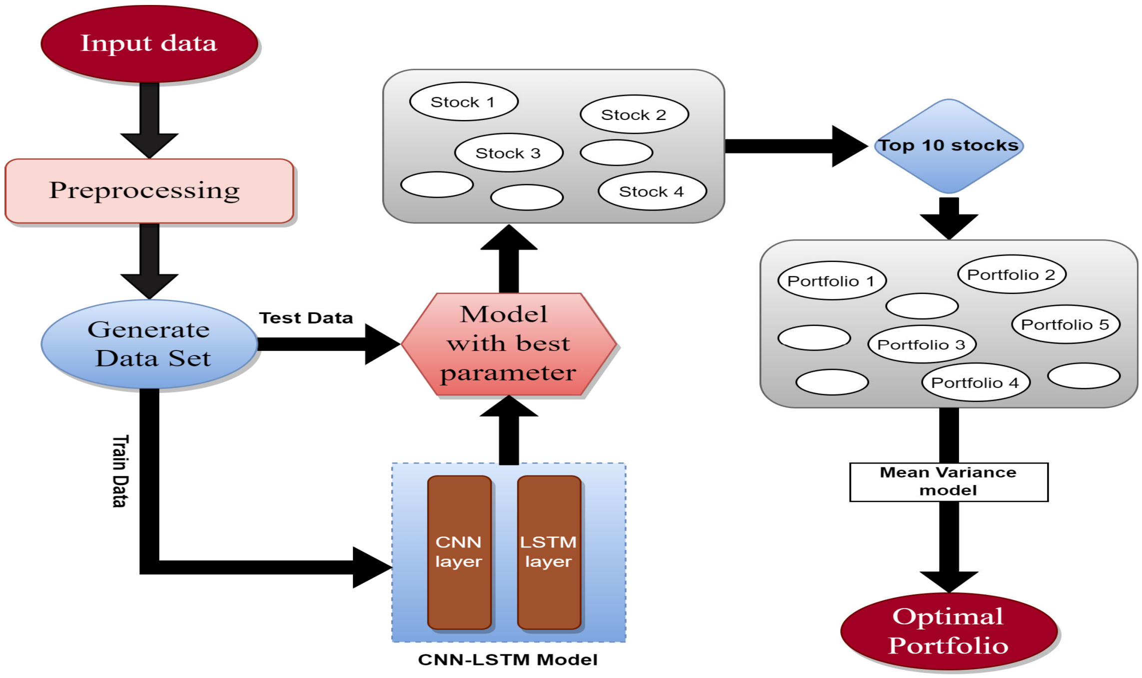


FIGURE 3. Architecture diagram.

the investment portfolio remains an integral part of the process.

3) Random + MV: In contrast to the CNN-LSTM + MV model, the Random + MV model tends to achieve the predicted return during the asset selection phase. The process of asset preselection is being approached naively. However, the random selection of assets must equal the number specified by the CNN-LSTM + MV model. The Markowitz optimization of the investment portfolio remains an integral part of the process. This type of baseline strategy's goal is to assess if deep learning-based asset preselection is necessary.
4) CNN-LSTM + 1/N: The architecture of this type of model is identical to the proposed model CNN-LSTM+MV. The primary aim is to examine the effectiveness of the MV model for optimal asset allocation. In particular, in the first step, the return on assets will be forecasted using the CNN-LSTM approach to shortlist the sample for stage II, in which the optimal portfolio selection is performed with a naive 1/N or equally weighted technique.

III. EXPERIMENTS AND RESULTS

A. DATA: INPUT AND PREPROCESSING

This paper randomly picks 21 stocks from NSE India, fetched from *yahoo finance*. For analysis, data from January 2005 to December 2021 are gathered and listed in Table 1 as potential assets. With such a big data set, individual traders may pick equities for their portfolio construction. Table 2 lists the input features for the presented research work. The close price is the target feature predicted (t+1 period) by examining the characteristics with the highest correlation coefficients. We employ features after a threshold of 0.3. The outcomes of feature selection using Pearson's correlation coefficient are displayed in Figure 4. We finally chose thirteen important indicators as input factors. These included eight technical indicators and five fundamental indicators.

Most DL algorithms may struggle if the variation of one characteristic is significantly larger than the variation of the others. To deal with this issue, we have used a min-max normalization method for the feature scaling. The equation for the min-max normalization method is as follows:

$$z = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{22}$$

where x and z are the original and the scaled input, respectively; similarly, x_{min} and x_{max} corresponds to the minimum and the maximum values of the input, respectively. According to the methods outlined, normalized data of the specified characteristics (listed in Table 2) has been generated. In addition, the data is reshaped and partitioned into training, testing, and validation data sets (Section II-D3). The objective is to accurately forecast the closing price of a selection of equities traded on NSE India, a market known for its complicated, chaotic, and volatile behavior. The whole process of choosing a model can be categorized into two groups: preselection and optimal portfolio construction.

TABLE 2. List of suggested feature input for the model.

Data Input	Frequency	Abbreviation	Source	Type of Indicator
Fundamental Indicator				
Open	Daily	O	Yahoo	Historical
Close	Daily	C	Yahoo	Historical
High	Daily	H	Yahoo	Historical
Low	Daily	L	Yahoo	Historical
Adj. Close	Daily	AC	Yahoo	Historical
Technical Indicator				
Simple Moving Average	Daily	SMA	_	Trend
Exponential Moving Average	Daily	EMA	_	Trend
Relative Strength Index	Daily	RSI	_	Momentum
Rate of Change	Daily	RoC	_	Momentum
True Range	Daily	TR	_	Volatility
Average True Range	Daily	ATR	_	Volatility
Momentum Index	Daily	MoM	_	Momentum
Commodity Channel Index	Daily	CCI	_	Trend

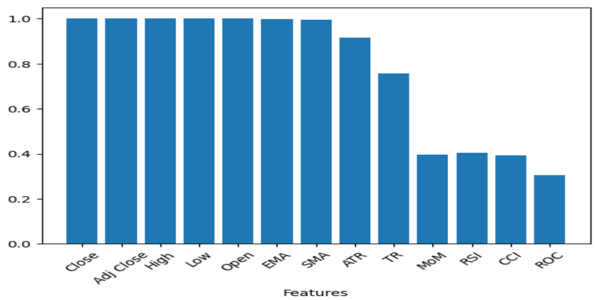


FIGURE 4. Feature selection results.

TABLE 3. Best parameters of models.

Parameters	Value		
	CNN	LSTM	CNN-LSTM
Filters	32	-	32
Kernel size	01	-	01
Padding	Same	-	Same
Pool size	01	-	01
Dropout	0.1	0.2	0.2
Neurons in LSTM layer	-	100	100
Timesteps	05	05	05
Activation function	Relu	Relu	Relu
Batch size	128	128	128
Learning rate	0.001	0.001	0.001
Optimizer	Adam	Adam	Adam
Loss function	MSE	MSE	MSE
Epochs	30	30	30

B. STAGE I- PRESELECTION: PREDICTION-BASED APPROACH

To demonstrate CNN-LSTM's effectiveness, we compared it to both the CNN and LSTM single models, running each against the same training, validation, and test datasets in the same environment. All tests are performed on a Windows 11 machine equipped with an Intel i5-4700H processor operating at 2.6 GHz, 12 GB of RAM, a 500 GB hard drive, and the latest service pack. Forecasts for the next day's close price are made using the variables listed in Table 2.

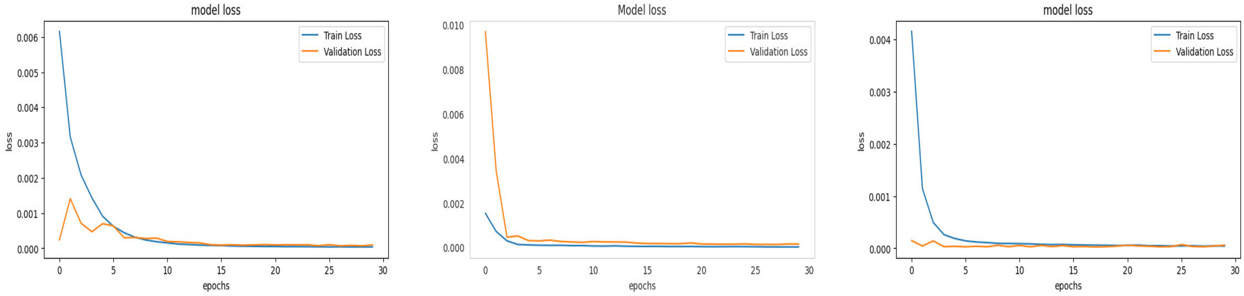
The following hyperparameters: (1) the number of epochs: 10 to 50, (2) activation function: tanh, Relu; (3) the number of neurons ranges from 2 to 200 per hidden layer; (4) learning rate: 0.1, 0.01, 0.001, 0.0001; (5) batch size, ranging from 50 to 128; (6) optimizer: RMSProp, SGD, Adam (7) loss function: Mean Absolute Error (MAE), Mean Squared Error (MSE) are examined. Following the model execution included in the Equations 11 to 20, the CNN-LSTM network's specified topology is validated. The properties and parameters employed by the proposed hybrid deep learning model and the single models, CNN and LSTM are provided in Table 3. The selection of the hyperparameters is based on empiricism [8]. The selected hyperparameter values were based on the insights gained from our trial and error experiments, where we sought to strike a balance between model complexity and its best performance on the validation data. The effectiveness of the chosen hyperparameters and suggested model is demonstrated for the '532488' stock. Figure 5 displays the loss function for training and validation data with each plot from the above-mentioned deep models. Furthermore, Figure 6 presents a comparison between the actual (test data) and predicted stock prices. The parameters and the layers common to the proposed CNN-LSTM model and CNN and LSTM models are kept the same for unbiased comparison. Mean squared error is used as the loss function because it best characterizes the discrepancy between the output value generated by the output layer and the actual value of the data.

These models' capacity to make accurate predictions is measured using six distinct performance metrics: MAE, RMSE, MAPE, R2, Max Error, and MedAE. Different metrics capture different error attributes. By using multiple metrics, a more holistic evaluation of the model's performance can be obtained. Multiple metrics test the model's robustness across different assessment criteria. Following is a description of the analytical version of these metrics:

Model Evaluation: Performance Metrics

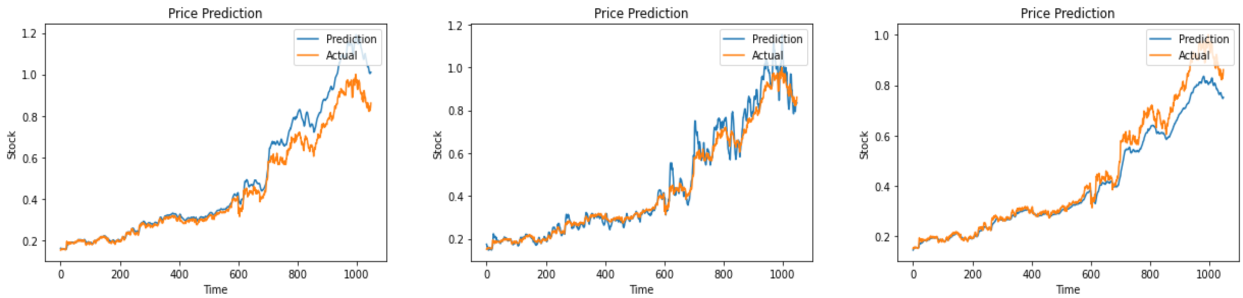
- Mean Absolute Error

$$MAE = \frac{1}{N}\Sigma_{i=1}^{N}|Y_i - \hat{Y}_i| \tag{23}$$



(a) CNN (b) CNN-LSTM (c) LSTM

FIGURE 5. Loss function for train and validation data.



(a) CNN (b) CNN-LSTM (c) LSTM

FIGURE 6. Stock price prediction on test data.

•Root Mean Squared Error

$$RMSE = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(Y_i - \hat{Y}_i)^2} \tag{24}$$

•Mean Absolute Percentage Error

$$MAPE = \frac{1}{N}\Sigma_{i=1}^{N}\left|\frac{Y_i - \hat{Y}_i}{Y_i}\right| \tag{25}$$

• R2 Score

$$R2 = 1 - \frac{\Sigma_{i=1}^{N}(Y_i - \hat{Y}_i)^2}{\Sigma_{i=1}^{N}(Y_i - \bar{Y}_i)^2} \tag{26}$$

• Maximum Error

The maximum residual error is calculated by the max_error() function. Calculates the maximum gap between the forecast and the actual result.

• Median Absolute Error

$$MedAE = Median(\Sigma_{i=1}^{N}|Y_i - \hat{Y}_i|) \tag{27}$$

where Y_i = Actual value, $\bar{Y}_i$ = Mean value and $\hat{Y}_i$ = Predicted value

The optimal model would have the lowest MAE, RMSE, MAPE, Max Error, and MedAE and the highest R2 score attainable. To account for the stochastic tendency, we run each model several times separately. The average score from these independent tests is used as one of the considerations for establishing the supremacy of the proposed model over other benchamarks. Tables 4, 5, and 6 provide comprehensive details of the results obtained using CNN, the suggested CNN-LSTM model, and the LSTM model, respectively. Moreover, Figure 7 displays the visual comparison among the aforementioned models.

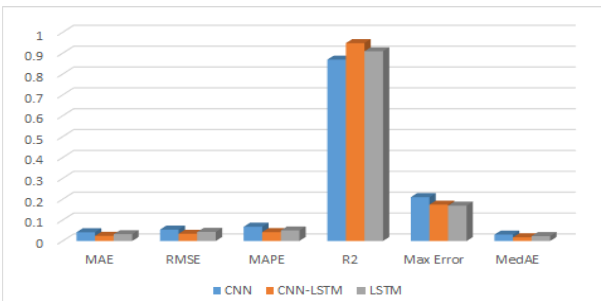


FIGURE 7. Average performance metrics.

In terms of prediction accuracy near real values, the CNN-LSTM model surpasses the individual models, CNN and LSTM. Compared to the single models, the suggested hybrid model has the lowest error rate and the greatest R2 score based on the performance criteria as shown in Tables 4, 5, and 6.

The suggested model's credibility and generalizability are established by the application of k-fold cross-validation (CV) [39], [40], which is a straightforward and productive

TABLE 4. Prediction performance of CNN model.

Stocks	MAE	RMSE	MAPE	R2	Max Error	MedAE
500875	0.0219	0.0265	0.0338	0.9549	0.1015	0.0196
500770	0.0267	0.0392	0.0604	0.9501	0.4416	0.0220
532540	0.0225	0.0309	0.0361	0.9687	0.1163	0.0160
509480	0.0767	0.1080	0.1079	0.7831	0.2911	0.0560
532174	0.0267	0.0372	0.0470	0.957	0.1281	0.0168
500112	0.0186	0.0263	0.0382	0.9752	0.1528	0.0137
500247	0.0448	0.0513	0.0655	0.8590	0.1891	0.0410
500331	0.0523	0.0659	0.0846	0.8596	0.2220	0.0418
500510	0.0264	0.0315	0.0406	0.9392	0.1251	0.0236
532454	0.0222	0.0291	0.0379	0.9737	0.1005	0.0173
500180	0.0361	0.0441	0.0519	0.8877	0.2115	0.0305
531642	0.0636	0.0715	0.0998	0.6670	0.1852	0.0546
532689	0.0656	0.0786	0.1014	0.6010	0.3581	0.0587
532538	0.0304	0.0391	0.0540	0.9424	0.1692	0.0240
532488	0.0519	0.0809	0.0841	0.8895	0.2415	0.0159
532155	0.0211	0.0267	0.0404	0.9735	0.1300	0.0181
500043	0.0329	0.0435	0.0529	0.9322	0.2198	0.0285
500800	0.0896	0.1257	0.1693	0.7310	0.3488	0.0438
532522	0.0158	0.0219	0.0201	0.8958	0.1412	0.0122
532281	0.0479	0.0732	0.0801	0.8379	0.2712	0.0188
532514	0.0857	0.0978	0.1280	0.6765	0.2671	0.0720
MEAN	0.04187619	0.054709524	0.068285714	0.869285714	0.210080952	0.030709524

TABLE 5. Prediction performance of CNN-LSTM model.

Stocks	MAE	RMSE	MAPE	R2	Max Error	MedAE
500875	0.0097	0.0138	0.0151	0.9877	0.0682	0.0071
500770	0.0168	0.0300	0.0394	0.970	0.4182	0.0101
532540	0.0178	0.0252	0.0332	0.979	0.1358	0.0127
509480	0.0795	0.0943	0.1381	0.8346	0.2247	0.0642
532174	0.0160	0.0256	0.0293	0.980	0.1743	0.0092
500112	0.0164	0.0235	0.0288	0.9802	0.1383	0.0110
500247	0.0243	0.0356	0.0363	0.9319	0.1808	0.0160
500180	0.0260	0.0334	0.0393	0.935	0.1364	0.0222
500331	0.0266	0.0373	0.0490	0.9549	0.2562	0.0213
500510	0.0113	0.0145	0.0181	0.9870	0.0726	0.0096
532454	0.0144	0.0210	0.0239	0.9862	0.0932	0.0091
531642	0.0258	0.0373	0.0376	0.9091	0.1381	0.0164
532689	0.0275	0.0429	0.0452	0.8811	0.2583	0.0189
532538	0.0266	0.0379	0.0405	0.9456	0.1467	0.0165
532488	0.0264	0.0406	0.0530	0.9720	0.1765	0.0142
532155	0.0138	0.0183	0.0270	0.9875	0.1060	0.0109
500043	0.0273	0.0373	0.0446	0.9500	0.1696	0.0200
500800	0.0297	0.0449	0.0594	0.9656	0.2016	0.0168
532522	0.0140	0.0187	0.0174	0.9243	0.1295	0.0105
532281	0.0245	0.0333	0.0510	0.9664	0.1340	0.0181
532514	0.0424	0.0580	0.0668	0.8863	0.2975	0.0330
MEAN	0.024609524	0.034447619	0.04252381	0.948304762	0.174119048	0.017514286

approach for performance evaluation and model comparison [41]. Since 75% of the dataset is utilized by all the models for training and validation, CV is applied to that portion; the remaining 25% is preserved as unseen data for testing the models. The data are arbitrarily partitioned into k distinct subsets of approximately equal size. In the i_{th} fold of cross-validation, the i_{th} subset is utilized to estimate the performance of a model trained on the remaining $k - 1$ subsets. The performance of the sample-trained models may be estimated as the mean of the performance seen overall k folds. The average predictive performance of the three models is determined using six performance indicators across $k = 10$ splits of the data. We compare a set of three models, CNN-LSTM, CNN, and LSTM, using cross-validated prediction performance. Figure 8 shows ten-fold cross-validation results for the aforementioned three models, which demonstrates the reliability and efficacy of the proposed CNN-LSTM model.

Additionally, the models of relevant researches of Fischer and Krauss [7] and Ta et al. [27] are reconstructed to establish the statistical significance and comparison with proposed research work. The average inference or the test time for the proposed model, along with comparison models, is listed in Table 7.

After predicting each asset individually, for the period $t + 1$, we will arrange the stocks in descending order of their predicted return. To go on to the next round, only the top k stocks in terms of return are considered. In accordance with Wang et al. [2], who discovered that a portfolio with

TABLE 6. Prediction performance of LSTM model.

Stocks	MAE	RMSE	MAPE	R2	Max Error	MedAE
500875	0.0144	0.0183	0.0240	0.978	0.1067	0.0125
500770	0.0369	0.0476	0.0712	0.926	0.3680	0.0344
532540	0.0363	0.0462	0.0556	0.930	0.1362	0.0258
509480	0.0734	0.1027	0.1050	0.8037	0.2708	0.0440
532174	0.0217	0.0326	0.0364	0.9677	0.1540	0.0126
500112	0.0161	0.0247	0.0286	0.9782	0.1147	0.0091
500247	0.0341	0.0424	0.0475	0.9036	0.1390	0.0283
500180	0.0355	0.0421	0.0485	0.897	0.1410	0.0289
500331	0.0291	0.0414	0.0438	0.944	0.1423	0.0191
500510	0.0109	0.0156	0.0169	0.9849	0.0842	0.0076
532454	0.0215	0.0294	0.0358	0.9730	0.1175	0.0147
531642	0.0310	0.0392	0.0462	0.899	0.1514	0.0250
532689	0.0353	0.0424	0.0510	0.8837	0.1778	0.0316
532538	0.0324	0.0505	0.0469	0.9040	0.2100	0.0155
532488	0.0355	0.0551	0.0592	0.9487	0.1962	0.0130
532155	0.0144	0.0193	0.0287	0.9861	0.1234	0.0113
500043	0.0344	0.0421	0.0536	0.9364	0.1524	0.0312
500800	0.0464	0.0683	0.0833	0.9203	0.2109	0.0221
532522	0.0419	0.0464	0.0511	0.534	0.1649	0.0406
532281	0.0267	0.0408	0.0455	0.9495	0.2086	0.0142
532514	0.0566	0.0674	0.0814	0.8462	0.1852	0.0464
MEAN	0.032505	0.043547619	0.050485714	0.909238095	0.169295238	0.023233333

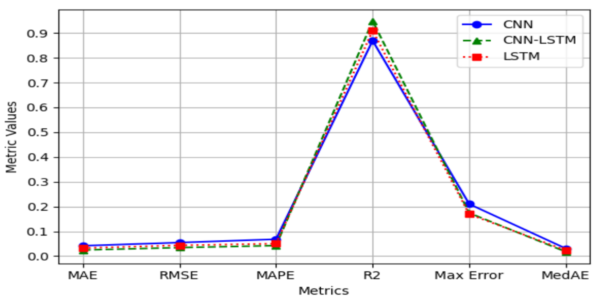


FIGURE 8. Average cross validation scores.

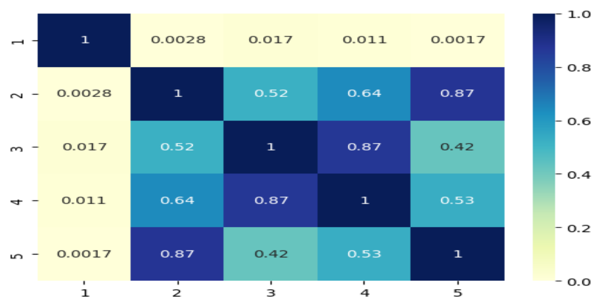


FIGURE 9. P-values.

TABLE 7. Average inference time.

Models	Time (In Sec.)
CNN	0.57
CNN-LSTM	0.61
LSTM	1.60
Fischer and Krauss (2018) [7]	0.97
Ta (2020) [27]	0.76

ten assets outperforms those with other numbers, we selected $k = 10$ stocks. Individual investors find it challenging to handle an excessive number of stocks. Consequently, several studies analyzed a portfolio containing equal or fewer than ten stocks [1], [52], [53]. The top ten shortlisted stocks are chosen for the second round of the portfolio optimization process. The second phase's objective is to acquire capital allocation for the chosen stocks.

C. STATISTICAL SIGNIFICANCE

For a more comprehensive evaluation of the models' performance and to add scientific rigor to our research, we employed statistical analysis of the performance of the proposed model. The RMSE was used as a performance metric which did not follow a normal distribution as indicated by the Shapiro test [54]. To evaluate the null hypothesis that the forecasts of method i have no significant difference as compared to the forecasts of method j, with $i, j \in$ {LSTM, CNN, CNN-LSTM, Fischer and Krauss [7], Ta et al. [27]} a non-parametric Kruskal-Wallis test [55] (also known as non-parametric ANOVA test) with a significance level of 0.05 is performed. The test yielded a p-value of $0.039 < 0.05$ (alpha), rejecting the null hypothesis, indicating a significant difference between the performance of the models. Further, post hoc analysis was conducted using the Conover test [56] for pairwise comparisons between the five models. The heatmap for the p-values is indicated below in Figure 9:

From the heatmap representation (see Figure 9), we can see that the p-value for the comparison between CNN-LSTM (model 1) and CNN (model 2) is 0.0086, and the p-value for the comparison between CNN-LSTM (model 1) and LSTM (model 3) is 0.0412, both are less than 0.05, indicating a significant difference between these two pair of models respectively. However, the p-value for the comparison between CNN (model 2) and LSTM (model

TABLE 8. Results of portfolio optimization (Before TC).

Objectives	Return	Risk	Sharpe Ratio
CNN +MV	22.10%	22.10%	0.91
CNN-LSTM +MV	24.00%	22.60%	0.97
LSTM +MV	22.30%	21.60%	0.94
Random+MV	17.90%	19.90%	0.80
CNN-LSTM + 1/N	24.1%	23.62%	0.936
Fischer and Krauss [7]	23.0%	21.9%	0.96
Ta et al. [27]	21.8%	21.6%	0.92

TABLE 9. Results of portfolio optimization (After TC).

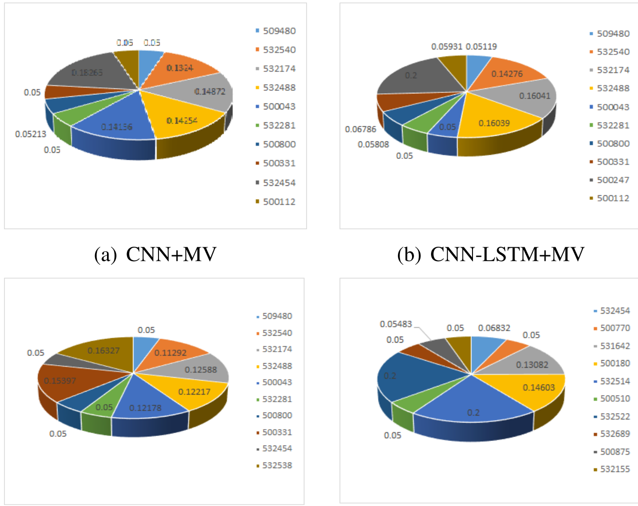
Objectives	Return	Risk	Sharpe Ratio
CNN +MV	22.10%	22.50%	0.89
CNN-LSTM +MV	23.90%	23.00%	0.95
LSTM+MV	22.70%	21.90%	0.95
Random+MV	16.90%	20.30%	0.73
CNN-LSTM + 1/N	23.1%	23.62%	0.932
Fischer and Krauss [7]	23.0%	22.2%	0.95
Ta et al. [27]	21.6%	21.9%	0.89

3) is 0.5306, which is greater than 0.05, suggesting that there is no significant difference between these two models in terms of their performance on the dataset. Additionally, p-values for the comparison between the suggested model (model 1) and the relevant researches of Fischer and Krauss [7] (model 4) and Ta et al. [27] (model 5) are 0.011 and 0.0017 respectively. Thus, the CNN-LSTM model proves its statistical significance over single models, i.e., CNN and LSTM, and recent relevant researches of Fischer and Krauss [7], and Ta et al. [27] respectively.

D. STAGE II: OPTIMAL PORTFOLIO CONSTRUCTION

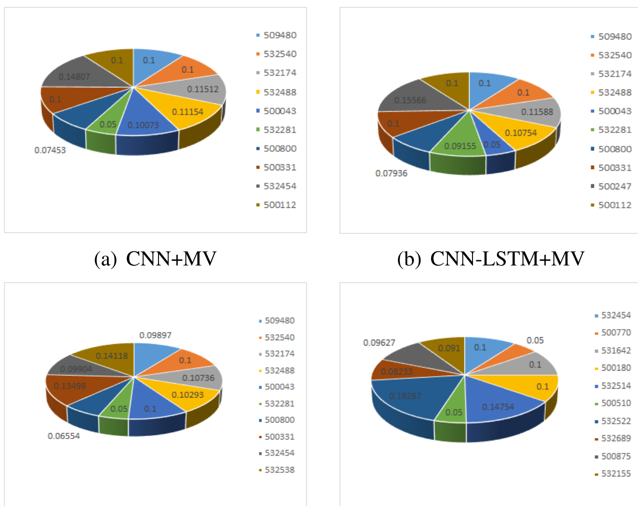
Stage II, "Optimal Portfolio Construction," seeks to get the capital allocation of the chosen portfolio that satisfies the risk-return tradeoff. It is implemented using the MV model described by Equation 1. The floor constraint is set to $li = 0.05$, or 05%, and the ceiling constraint to $ui = 0.2$, or 20% of the capital amount, to prevent the imbalanced allocation. Considering that a high turnover would lead to a higher transaction fee in actual stock trading investments, it is important to investigate these models' true performance after taking into account the impact of transaction costs (TC) [57]. This part aims to analyze the profitability of various models in the stock market by discussing their performance after the transaction fee generated by turnover has been deducted. For the sake of brevity, this study utilizes the transaction charge due to turnover of 0.01 % per unit to represent the total transaction fee of a real trading investment. The average annual financial results of portfolio optimization with and without transaction costs for the CNN-LSTM+MV are laid down in Table 8 and Table 9, respectively, against the baselines and relevant published research of Fischer and Krauss [7] and Ta et al. [27].

The CNN-LSTM+MV yields the greatest Sharpe ratio (Eq. 21) both before and after accounting for transaction costs, assuming a risk-free rate of 0.02. When the cost of



(a) CNN+MV (b) CNN-LSTM+MV
(c) LSTM+MV (d) Random+MV

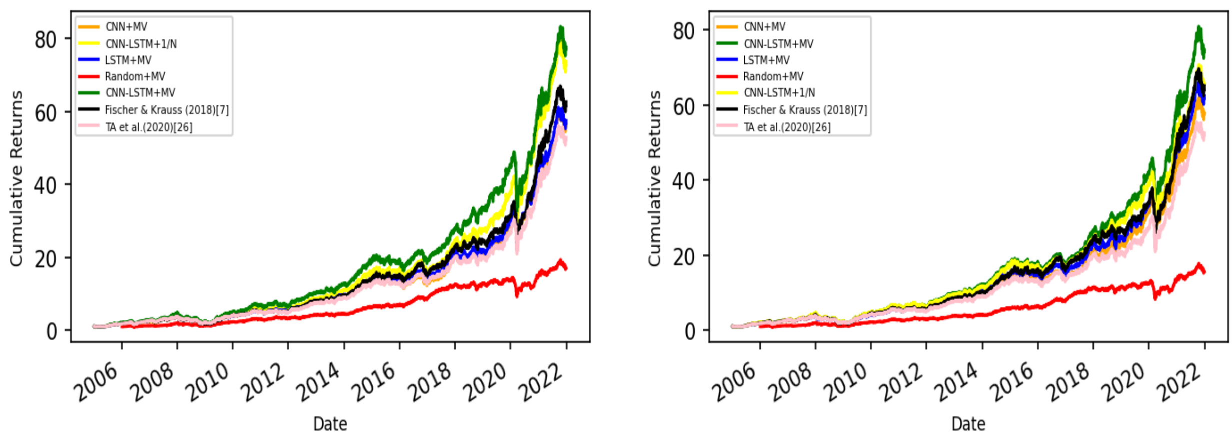
FIGURE 10. Asset allocation (before transaction cost).



(a) CNN+MV (b) CNN-LSTM+MV
(c) LSTM+MV (d) Random+MV

FIGURE 11. Asset allocation (after transaction cost).

the transaction is included in the LSTM+MV algorithm, a tie remains. Figure 10 and Figure 11 show the pre and post-transaction cost asset allocation for the selected companies as calculated by the proposed hybrid deep learning model and the baselines, respectively. The MV model is used to determine the optimal allocation. The daily cumulative return for each model, excluding transaction costs, is shown in Figure 12(a). The average annualized cumulative returns for the CNN-LSTM+MV, LSTM+MV, and CNN-LSTM+ 1/N models are 25.62%, 24.26%, and 24.91% respectively, while those for the CNN+MV and Random+MV models are 24.06% and 19.9%, respectively. In contrast, Figure 12(b) shows daily cumulative return after accounting for transaction costs.



(a) Before Transaction Cost

(b) After Transaction Cost

FIGURE 12. Cumulative return.

An average annualized cumulative return of 25.39% is achieved by the suggested hybrid CNN-LSTM+MV model, followed by the CNN-LSTM+ 1/N at 24.01%, the LSTM+MV model at 24.45%, the CNN+MV model at 23.91%, and the Random+MV model at 19.5%. We also compared the proposed model with two relevant pieces of research [7], [27]. The average annualized cumulative return for [7] and [27] before transaction cost are 24.88 % and 23.8 %, in addition to 24.78 % and 23.6 % after the inclusion of transaction costs, respectively. Multiple indicators corroborate the superior performance of the proposed hybrid deep learning model.

IV. CONCLUSION

A. DISCUSSION

This research proposes a methodology for making financial investment decisions entitled CNN-LSTM+MV. Our technique fills a gap in the literature by combining feature extraction with sequential learning to preserve better the continuity and memory of financial time series data. We show that CNN-LSTM networks outperform both CNN and LSTM models when it comes to forecasting financial time series. Top ten stocks with high anticipated returns are shortlisted from a sample of 21 stocks to form an optimal portfolio through MV model. Six performance indicators are used to demonstrate the superiority of our proposed model compared to existing prediction models. Statistical significance is determined with the non-parametric Kruskal-Wallis test, followed by pairwise comparisons with the Conover test, verifying the CNN-LSTM model's superiority over single models, CNN and LSTM along with reconstructed recent piece of researches, Fischer and Krauss [7] and Ta et al. [27]. A 10-fold cross-validation is established to compare the models and advocate the generalizability and credibility of the proposed model. In addition, we demonstrate the practical application of our model to portfolio construction. We achieve favorable returns, risks, and risk-return measures by leveraging the predicted values and employing the MV diversification method. Comparative analysis with relevant research and baseline models demonstrates the outperformance of our proposed model in terms of annualized return, Sharpe ratio, risk, and cumulative returns prior to and following the incorporation of transaction costs.

B. LIMITATIONS AND FUTURE WORK

To convey a full picture of the consequences, and potential future applications of this work, its limitations must be acknowledged. The CNN-LSTM+MV model's efficacy and precision depend on the integrity of the information it is fed. Furthermore, the model's results are based on experiments on 21 datasets of the Indian stock market. Thus, the generalizability of the model over markets with different geographical, political, and economic conditions must be addressed with careful study and experimentation. Adding more variables to the model or using other data sources to improve its prediction power are areas that may be explored in further study. A wide range of portfolio selection criteria may be investigated for various investing scenarios. By acknowledging and addressing these limitations, we can foster a more robust and informed approach to financial investment decision-making.

In conclusion, the results of this study show that the CNN-LSTM+MV approach is better and more effective for use in the context of financial investment decision-making. By combining advanced deep learning techniques with comprehensive performance evaluation, our model provides valuable insights for investors seeking improved prediction accuracy and portfolio performance.

ACKNOWLEDGMENT

This work was supported by King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSPD2023R704.

REFERENCES

[1] F. D. Paiva, R. T. N. Cardoso, G. P. Hanaoka, and W. M. Duarte, "Decision-making for financial trading: A fusion approach of machine learning and portfolio selection," *Expert Syst. Appl.*, vol. 115, pp. 635–655, Jan. 2019.

[2] W. Wang, W. Li, N. Zhang, and K. Liu, "Portfolio formation with preselection using deep learning from long-term financial data," *Expert Syst. Appl.*, vol. 143, Apr. 2020, Art. no. 113042.

[3] P. Gupta, M. K. Mehlawat, and A. Z. Khan, "Multi-period portfolio optimization using coherent fuzzy numbers in a credibilistic environment," *Expert Syst. Appl.*, vol. 167, Apr. 2021, Art. no. 114135.

[4] S. Ashfaq, U. Ayub, G. Mujtaba, N. Raza, and S. Gulzar, "Gainers and losers with higher order portfolio risk optimization," *Phys. A, Stat. Mech. Appl.*, vol. 563, Feb. 2021, Art. no. 125416.

[5] H. Markowitz, "Portfolio selection," *J. Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[6] J. B. Guerard, H. Markowitz, and G. Xu, "Earnings forecasting in a global stock selection model and efficient portfolio construction and management," *Int. J. Forecasting*, vol. 31, no. 2, pp. 550–560, Apr. 2015.

[7] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, Oct. 2018.

[8] Y. Ma, R. Han, and W. Wang, "Prediction-based portfolio optimization models using deep neural networks," *IEEE Access*, vol. 8, pp. 115393–115405, 2020.

[9] F. D. Freitas, A. F. De Souza, and A. R. De Almeida, "Prediction-based portfolio optimization model using neural networks," *Neurocomputing*, vol. 72, no. 10, pp. 2155–2170, Jun. 2009.

[10] T.-J. Chang, S.-C. Yang, and K.-J. Chang, "Portfolio optimization problems in different risk measures using genetic algorithm," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10529–10537, Sep. 2009.

[11] S. T. Tayali, "A novel backtesting methodology for clustering in mean–variance portfolio optimization," *Knowl.-Based Syst.*, vol. 209, Dec. 2020, Art. no. 106454.

[12] Y. Ma, R. Han, and W. Wang, "Portfolio optimization with return prediction using deep learning and machine learning," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113973.

[13] L. Min, J. Dong, J. Liu, and X. Gong, "Robust mean-risk portfolio optimization using machine learning-based trade-off parameter," *Appl. Soft Comput.*, vol. 113, Dec. 2021, Art. no. 107948.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[15] Y. S. Asawa, "Modern machine learning solutions for portfolio selection," *IEEE Eng. Manag. Rev.*, vol. 50, no. 1, pp. 94–112, 1st Quart., 2022.

[16] M. Z. Islam, M. M. Islam, and A. Asraf, "A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images," *Informat. Med. Unlocked*, vol. 20, 2020, Art. no. 100412.

[17] Y. Baek and H. Y. Kim, "ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module," *Expert Syst. Appl.*, vol. 113, pp. 457–480, Dec. 2018.

[18] K. Vijayaprabakaran K. Sathiyamurthy, "Towards activation function search for long short-term model network: A differential evolution based approach," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2637–2650, Jun. 2022.

[19] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," Jul. 2017, *arXiv:1706.10059*.

[20] W. Zhang, T. Yin, Y. Zhao, B. Han, and H. Liu, "Reinforcement learning for stock prediction and high-frequency trading with T+1 rules," *IEEE Access*, vol. 11, pp. 14115–14127, 2023.

[21] X. Yan, H. Yang, Z. Yu, S. Zhang, and X. Zheng, "Portfolio optimization: A return-on-equity network analysis," *IEEE Trans. Computat. Social Syst.*, early access, Apr. 7, 2023, doi: 10.1109/TCSS.2023.3261881.

[22] J. K. Pahade and M. Jha, "A hybrid fuzzy-SCOOT algorithm to optimize possibilistic mean semi-absolute deviation model for optimal portfolio selection," *Int. J. Fuzzy Syst.*, vol. 24, no. 4, pp. 1958–1973, Feb. 2022.

[23] O. B. Sezer and A. M. Ozbayoglu, "Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach," *Appl. Soft Comput.*, vol. 70, pp. 525–538, Sep. 2018.

[24] S. Chen and H. He, "Stock prediction using convolutional neural network," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 435, Nov. 2018, Art. no. 012026.

[25] Y. Komori, "Convolutional neural network for stock price prediction using transfer learning," Social Sci. Res. Netw., Rochester, NY, USA, Tech. Rep. ssrn.3756702, Dec. 2020.

[26] J.-R. Yu, W.-J. Paul Chiou, W.-Y. Lee, and S.-J. Lin, "Portfolio models with return forecasting and transaction costs," *Int. Rev. Econ. Finance*, vol. 66, pp. 118–130, Mar. 2020.

[27] V.-D. Ta, C.-M. Liu, and D. A. Tadesse, "Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading," *Appl. Sci.*, vol. 10, no. 2, p. 437, Jan. 2020.

[28] H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal, K. R. Dahal, and R. K. C. Khatri, "Predicting stock market index using LSTM," *Mach. Learn. With Appl.*, vol. 9, Sep. 2022, Art. no. 100320.

[29] F. Z. Habbab, M. Kampouridis, and A. A. Voudouris, "Optimizing mixed-asset portfolios involving REITs," in *Proc. IEEE Symp. Comput. Intell. Financial Eng. Econ. (CIFEr)*, May 2022, pp. 1–8.

[30] 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. Manage. (ICIPTM)*, vol. 2, Feb. 2022, pp. 156–161.

[31] F. Aksan, Y. Li, V. Suresh, and P. Janik, "CNN-LSTM vs. LSTM-CNN to predict power flow direction: A case study of the high-voltage Subnet of Northeast Germany," *Sensors*, vol. 23, no. 2, p. 901, Jan. 2023.

[32] H. Xu, Y. Zhang, and Y. Xu, "Promoting financial market development–financial stock classification using graph convolutional neural networks," *IEEE Access*, vol. 11, pp. 49289–49299, 2023.

[33] K. Cui, R. Hao, Y. Huang, J. Li, and Y. Song, "A novel convolutional neural networks for stock trading based on DDQN algorithm," *IEEE Access*, vol. 11, pp. 32308–32318, 2023.

[34] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jul. 2005.

[35] P. Yu and X. Yan, "Stock price prediction based on deep neural networks," *Neural Comput. Appl.*, vol. 32, no. 6, pp. 1609–1628, Mar. 2020.

[36] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," *PLoS ONE*, vol. 14, no. 2, Feb. 2019, Art. no. e0212320.

[37] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-based model to forecast stock prices," *Complexity*, vol. 2020, pp. 1–10, Nov. 2020.

[38] S. Ahmed, R. K. Chakrabortty, D. L. Essam, and W. Ding, "Poly-linear regression with augmented long short term memory neural network: Predicting time series data," *Inf. Sci.*, vol. 606, pp. 573–600, Aug. 2022.

[39] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Inf. Sci.*, vol. 191, pp. 192–213, May 2012.

[40] L. A. Yates, Z. Aandahl, S. A. Richards, and B. W. Brook, "Cross validation for model selection: A review with examples from ecology," *Ecolog. Monographs*, vol. 93, no. 1, Feb. 2023, Art. no. e1557.

[41] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statist. Surv.*, vol. 4, pp. 40–79, Jan. 2010.

[42] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[43] A. M. Hasan, H. A. Jalab, F. Meziane, H. Kahtan, and A. S. Al-Ahmad, "Combining deep and handcrafted image features for MRI brain scan classification," *IEEE Access*, vol. 7, pp. 79959–79967, 2019.

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[45] P. Gupta, M. K. Mehlawat, and A. Saxena, "Asset portfolio optimization using fuzzy mathematical programming," *Inf. Sci.*, vol. 178, no. 6, pp. 1734–1755, Mar. 2008.

[46] N. J. Jobst, M. D. Horniman, C. A. Lucas, and G. Mitra, "Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints," *Quant. Finance*, vol. 1, no. 5, pp. 489–501, May 2001.

[47] R. O. Michaud and R. Michaud, "Estimation error and portfolio optimization: A resampling solution," New Frontier Advisors, LLC, Boston, MA, USA, Tech. Rep. 1138, 2007.

[48] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 259–268, Jan. 2015.

[49] E. Beyaz, F. Tekiner, X.-j. Zeng, and J. Keane, "Comparing technical and fundamental indicators in stock price forecasting," in *Proc. IEEE 20th Int. Conf. High Perform. Comput. Commun.; IEEE 16th Int. Conf. Smart City; IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 1607–1613.

[50] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," Nov. 2018, *arXiv:1811.03378*.
[51] M. R. Lyle and T. L. Yohn, "Fundamental analysis and mean-variance optimal portfolios," *Accounting Rev.*, vol. 96, no. 6, pp. 303–327, Nov. 2021.
[52] M. Ehrgott, K. Klamroth, and C. Schwehm, "An MCDM approach to portfolio optimization," *Eur. J. Oper. Res.*, vol. 155, no. 3, pp. 752–770, Jun. 2004.
[53] P. Gupta, M. K. Mehlawat, and A. Saxena, "Hybrid optimization models of portfolio selection involving financial and ethical considerations," *Knowl.-Based Syst.*, vol. 37, pp. 318–337, Jan. 2013.
[54] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, nos. 3–4, pp. 591–611, Dec. 1965.
[55] R. A. K. Sherwani, H. Shakeel, W. B. Awan, M. Faheem, and M. Aslam, "Analysis of COVID-19 data using neutrosophic Kruskal wallis h test," *BMC Med. Res. Methodol.*, vol. 21, no. 1, p. 215, Oct. 2021.
[56] W. J. Conover and R. L. Iman, "Rank transformations as a bridge between parametric and nonparametric statistics," *Amer. Statistician*, vol. 35, no. 3, pp. 124–129, Aug. 1981.
[57] Y. Fang, K. K. Lai, and S.-Y. Wang, "Portfolio rebalancing model with transaction costs based on fuzzy decision theory," *Eur. J. Oper. Res.*, vol. 175, no. 2, pp. 879–893, Dec. 2006.

PRIYA SINGH received the B.Sc. degree (Hons.) in mathematics from the University of Delhi, India, and the M.Sc. degree in mathematics from the Government Holkar Science College, Indore, Madhya Pradesh, India. She is currently a Doctoral Researcher with the Department of Mathematics, Bioinformatics and Computer Applications, Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, exploring a hybrid approach for asset selection and portfolio optimization through intelligent techniques.

MANOJ JHA received the B.Sc. and M.Sc. degrees in mathematics from Barkatullah University, Bhopal, Madhya Pradesh, India, the M.Phil. degree from DAVV, Indore, Madhya Pradesh, and the Ph.D. degree in financial mathematics from the Maulana Azad National Institute of Technology, Bhopal. He authored three books with various research publications. His research interests include financial mathematics and portfolio optimization.

MOHAMED SHARAF received the Ph.D. degree in industrial engineering from Chiba University, Japan. He is currently the Head of the Development and Quality Unit, College of Engineering, King Saud University. He has published more than 30 articles in the areas of spare parts control, quality management, maintenance, six-sigma methodology, and academic accreditation.

MOHAMMED A. EL-MELIGY received the B.Sc. degree in information technology from Menoufia University, Egypt, in 2005. He has been a Software Engineer with King Saud University, Riyadh, Saudi Arabia, since 2009. His research interests include Petri nets, supervisory control of discrete event systems, database software, and network administration.

THIPPA REDDY GADEKALLU received the bachelor's degree in computer science and engineering from Nagarjuna University, India, in 2003, the master's degree in computer science and engineering from Anna University, Chennai, Tamil Nadu, India, in 2011, and the Ph.D. degree from the Vellore Institute of Technology, Vellore, Tamil Nadu, in 2017. He was an Associate Professor with the School of Information Technology and Engineering, Vellore Institute of Technology. He is currently a Chief Engineer with Zhongda Group, Haiyan, Jiaxing, China. He has more than 14 years of experience in teaching. He has more than 200 international/national publications in reputed journals and conferences. His current research interests include machine learning, the Internet of Things, deep neural networks, blockchain, and computer vision. He is an Editor of several publishers, such a Springer, Hindawi, Plosone, Scientific Reports (Nature), and Wiley. He also acted as the guest editor of several reputed publishers, such as IEEE, Springer, Hindawi, and MDPI. He is recently recognized as one among the top 2% scientists in the world as per the survey conducted by Elsevier, in 2021.