Deep Learning for Stock Price Prediction and Portfolio Optimization

Ashy Sebastian, Dr. Veerta Tantia
Department of Commerce, Christ University, Bengaluru, India

***Abstract*—Using deep learning for stock market predictions and portfolio optimizations is a burgeoning field of research. This study focuses on the stock market dynamics in developing countries, which are often considered less stable than their developed counterparts. The study is structured in two stages. In the first stage, the authors introduce a stacked LSTM model for predicting NIFTY stocks and then rank the stocks based on their predicted returns. In the second stage, the high-return stocks are selected to form 30 different portfolios with six different objectives, each comprising the top 7, 8, 9, and 10 NIFTY stocks. These portfolios are then compared based on risk and returns. Experimental results show that portfolios with five stocks offer the best returns and that adding more than nine stocks to the portfolio leads to excessive diversification and complexity. Therefore, the findings suggest that the proposed two-stage portfolio optimization method has the potential to construct a promising investment strategy, offering a balance between historical and future information on assets.**

Keywords*—Deep learning; long-short term memory; stock price prediction; portfolio optimization; emerging markets; Indian stock market

I. Introduction

Portfolio optimization is an essential component of a trading system. The goal of optimization is to determine the optimal asset allocation within a portfolio to maximize returns for a given level of risk. This concept, widely known as modern portfolio theory (MPT), was pioneered by study [1].

The main advantage of creating an optimal portfolio is that it encourages diversification, which helps stabilize the equity curve and results in a higher return per unit of risk than trading a single asset. Nevertheless, despite the undeniable power of such diversification, the selection and implementation of the right asset allocations in a portfolio can be challenging due to significant fluctuations in financial market dynamics over time. For instance, assets that exhibited strong negative correlations in the past could be positively correlated in the future, and individual assets in the same asset class often show high positive correlations [2]. This adds extra risk to the portfolio, degrades its subsequent performance and undermines investor confidence

Most traditional portfolio research overlooks the selection of high-quality assets and instead focuses on enhancing strategy performance. However, pre-selecting high-quality assets is crucial for optimal portfolio formation [3]. Zhou et al. [4] noted that the success of portfolio management relies on the initial selection of high-quality stocks. Investors attempt to predict the future returns of stocks and determine the optimal weight based on the highest predicted returns to construct a portfolio [5]**.** During the portfolio optimization process, the expected return on an asset is a vital consideration, highlighting the importance of preliminary asset selection in effective portfolio management [6]. Doing this substantially reduces the scope of possible assets to choose from. Typically, high profits are associated with high risk. However, in portfolio optimization, risk can be minimized by selecting optimal securities and assets. Thus, combining forecasting theory with portfolio selection could improve portfolio returns [7].

With advancements in machine learning and deep learning, though predicting asset returns has become feasible, these prediction results are not yet effectively utilized in practice for portfolio creation and optimization. As a result, many portfolios fail to fully capitalize on the available predictive insights, limiting their potential for improved performance and risk management. The challenge lies in effectively utilizing these predicted returns to construct an optimal investment portfolio. Considering this, the research seeks to address the issue by exploring how to integrate advanced forecasting information into the portfolio selection process.

Literature evidence that the strong functioning of stock markets has a significant effect on the overall growth of an economy, especially in a developing one. Nevertheless, forecasting in emerging markets is more difficult than in developed ones [8] due to their greater volatility, which can be affected by external reasons such as oil prices and the performance of developed markets. Emerging markets, such as China and India, comprise a significant portion of the global economy. Trading in these markets requires a different approach than in developed markets, as they possess unique characteristics such as higher volatility and potentially greater average returns. These markets often lack full information efficiency, owing to institutional barriers that impede information flow and the inexperience of market participants in quickly incorporating new information into security prices. In [9], a genetic algorithm was used to select stock portfolios in the emerging Asian markets of Vietnam, Thailand, Philippines, Singapore, Malaysia, and Indonesia. Although the Indian stock market has made significant advancements, there is a dearth of research on prediction-based portfolios. Furthermore, NSE (National Stock Exchange) and BSE (Bombay Stock Exchange), India's two major stock exchanges, have around 1600 and 5000 listed companies, respectively, traded on them daily. Hence, it is challenging for individual investors to decide on the number and type of stocks. Therefore, the development of appropriate asset selection and portfolio optimization models can assist investors in emerging markets to maximize profits and minimize risk. Sen et al. [10] conducted a study on the Indian stock wherein the top

five stocks from nine different sectors of NSE were taken for creating minimum-variance and optimum-risk portfolios. An LSTM (Long-short term memory) model was then designed to predict the future prices of the stocks in each portfolio. Five months after the portfolio construction, the actual return and the return predicted by the LSTM model are computed and compared. Nevertheless, the study was limited to return prediction and did not consider asset pre-selection. Hence, to fill this gap, this study attempts to select high-quality assets from the Indian stock market through deep learning-based forecasting and build a competitive portfolio for improved returns.

The purpose of this paper is to construct an optimized portfolio based on the asset returns forecasted by deep learning. The authors argue that a complete portfolio consists of two stages, where the first stage is the pre-selection of high-quality assets, and the second stage is the determination of optimal weights for the portfolio. In the empirical part, this study selects the constituent stocks of NIFTY50 to test the proposed methodology. In the primary stage, this paper proposes a stacked LSTM for stock price prediction and then calculates returns based on these predicted values. In the second stage, the stocks are ranked based on these returns, and the top N stocks are selected for portfolio formation and optimization. These selections are then compared against a benchmark index to establish superior performance.

This study addresses the following questions.

- Does asset pre-selection before optimal portfolio formation enhance the portfolio performance compared to the traditional approach without pre-selection?
- Does the performance of different-sized portfolios show significant variations?
- What is the optimal number of stocks to be held in a portfolio for maximum return?

The major contributions of this paper are:

This research makes significant theoretical contributions by developing a stacked LSTM model for predictions in the Indian stock market. By demonstrating the model's efficacy in handling the intricacies of the Indian stock market, characterized by high volatility and noise, this study fills a crucial gap in the literature.

Most relevant studies concentrate on the stage of optimal portfolio formation using the mean-variance framework but overlook the pre-selection of stocks, which occurs before the formation of the optimal portfolio. Markowitz advised against putting all "eggs in one basket," emphasizing risk reduction through diversification. However, the mean-variance approach does not tackle the issue of deciding the number of assets to invest in. The application of LSTM for identifying high-potential stocks before constructing portfolios introduces a novel approach to portfolio management.

By accurately predicting the future performance of various stocks, the model helps investors identify those with the highest potential returns and the most favorable risk profiles. This pre selection process ensures that only the most promising stocks are included in the portfolio, improving overall performance. This further helps to determine the optimum number of stocks to be held while creating a portfolio.

Additionally, this study highlights the significant potential for advancing research on deep learning in emerging markets, integrating new academic insights to better understand financial decision-making and time-series forecasting. Consequently, this work enhances the existing literature on the application of deep learning models for emerging markets.

To the best of the authors' knowledge, this is the first study of its kind that combines forecasting theory with portfolio optimization for the Indian stock market index NIFTY50, and it is believed that this paper is making a substantial contribution to the related literature.

The remainder of this article is organized in the following order. Previous literature is discussed in Section II and Methodology is in Section III. The experimental results and discussion are in Section IV. In Section V, the authors mark their concluding thoughts, and Section VI discusses limitations and future scope.

II. Literature Review

This literature review of this study is divided into two sections. The first section discusses the concept of deep learning and the second section deals with portfolio optimization. This is followed by the research gap.

A. Deep Learning

Deep learning, a subset of machine learning, employs artificial neural networks (ANN) to address complex problems. ANNs are composed of an input layer, hidden layers, and an output layer, with each node in a layer connected to every node in the subsequent layer. By adding more hidden layers, the network becomes deeper, leading to the formation of deep neural networks (DNNs). The term "deep" in deep learning refers to this increased complexity, achieved by adding more hidden layers between the input and output layers.

They have been used for financial forecasting, including economic recession predictions [11], portfolio optimization [2],[12],[13], sentiment analysis [14],[15] and much more. Several studies have investigated the use of deep learning models like LSTM, CNN, RNN and their variations for predicting price movements and trends in financial markets [16]. These studies have shown promising outcomes, demonstrating that deep learning techniques can effectively capture and extract important features and patterns from trading data. Dixon et al. [17] employed deep neural networks to predict daily market movement directions and validated their model through backtesting with a basic trading strategy. A CNN was used by [18] to forecast the hourly direction of BIST 100 stocks, utilizing a "specifically ordered feature set." This research extracted various technical indicators, price data, and temporal features while using chi-square feature selection to minimize noise and dimensionality. Another study in [19] focused on predicting the next-minute direction of the SPDR S&P 500 by proposing slim versions of LSTM models. Their results indicated that Slim LSTM when combined with technical indicators, outperformed the standard LSTM model. Moreover, [20] examined the predictability of intraday movements over different time

intervals, analyzing their model's performance during high-volatility periods. The findings revealed the presence of "time-delayed correlations" in S&P 500 stocks in both stable and volatile market conditions and provided evidence supporting the effectiveness of deep learning methods in forecasting trends across numerous interrelated time series. In study [21], several ML and DL-based techniques, such as MNB classifier, SVM, Naïve Bayes, linear regression, and LSTM, were effectively created and executed. These methods utilized the general public's sentiment, viewpoints, news, and past stock prices to predict BSE and Infosys stock prices. Shah et al. [22] developed a model for predicting the NIFTY closing values by combining a CNN, LSTM, and dense layers within a 20-day time frame. Ghosh et al. [23] utilized both Random Forests and LSTM networks to assess their performance in predicting out-of-sample directional movements of S&P 500 constituent stocks for intraday trading. Their analysis spanned from January 1993 to December 2018. More recently, [24] successfully predicted daily stock movements of the BIST 30 index using an ensemble learning algorithm combined with a set of ten different variable groups.

1) LSTM: Despite the existence of numerous deep learning techniques, considerable efforts have been dedicated to demonstrating that LSTM can outperform other methods in the prediction of time-series data. Li et al. [25] conducted a systematic review investigating the application of deep learning models in predicting stock market trends using technical analysis. The review found that the LSTM model was the most commonly used and preferred algorithm for stock market prediction due to its ability to store memory and address the gradient vanishing problem. In their research, [26] utilized ten stock technical indicators along with ten years of historical data from the Tehran stock exchange across multiple machine learning and deep learning models. The study showed that the LSTM model outperformed others in terms of prediction accuracy, displaying the lowest error rate and the highest capacity to fit the data effectively. Abbasimehr et al. [27] proposed a technique of a multi-layer LSTM network with a grid search method. The suggested approach looks for the LSTM network's ideal hyperparameters. The authors used actual demand data from a furniture company to test the efficacy of their suggested strategy and compared it to other cutting-edge time series forecasting methods. The researchers concluded that the proposed model outperformed the alternatives significantly and can be applied to real-world scenarios, like stock price prediction, weather forecasting, and energy demand forecasting. Furthermore, LSTM neural networks have emerged as leading models for a diverse range of machine learning tasks, varying greatly in scale and characteristics. The core concept underlying LSTM architecture is the presence of a memory cell capable of retaining its state over time, complemented by non-linear gating units that control the flow of information into and out of the cell [28]. Consequently, the existing literature inspires us to adopt LSTM for this study because of its ability to analyze relationships among time-series data through its memory function.

B. Portfolio Optimization

Portfolio formation involves two primary concerns: choosing assets with the potential for higher returns and determining the optimal asset composition to achieve the goal of maximizing returns while minimizing risk. A quantitative approach is often employed in making these investment decisions. During the portfolio optimization process, the expected return on an asset is a vital consideration, highlighting the importance of preliminary asset selection in effective portfolio management [6]. Selecting the appropriate asset allocations for a portfolio is challenging because financial market conditions can fluctuate significantly over time.

1) Return prediction and asset pre-selection: Consequently, integrating stock return predictions with portfolio optimization models is essential for effective financial investment [29]. Scholars often use predicted returns instead of historical averages to enhance portfolio optimization models [30], [31].

Huang [32] introduced a model for stock selection combining SVR with genetic algorithms. In this model, SVR was used to forecast the future returns of individual stocks, while the genetic algorithm optimized the model's parameters and input features. The highest-ranked stocks were then equally weighted to construct a portfolio. The findings demonstrated that this proposed model outperformed the benchmarks in investment performance. Hao et al. [31] used an Auto Regressive-Multi Resolution Neural Network (AR-MRNN) and SVM for return prediction, and then prediction-based portfolio selection models were developed using these methods. Comparing the prediction accuracy, the SVM predictor outperforms the AR-MRNN predictor. Additionally, the SVM-based portfolio selection model surpassed the AR-MRNN-based and mean-variance models in performance. The analysis also showed that higher prediction accuracy leads to better returns. Performance comparison of an RNN, GRU, and LSTM for predicting stock prices was conducted by [33]. Their experimental results indicated that the LSTM neural network outperformed the other models. Additionally, they developed portfolios based on predictive thresholds using the LSTM neural network's forecasts. This approach was more data-driven compared to traditional models in portfolio design. Experimental results revealed that these portfolios achieved promising returns.

Wang et al. [3] proposed a mixed method of LSTM and mean-variance model for creating an optimal portfolio. The LSTM was used for return prediction, and then the stocks with the highest predicted returns were selected for portfolio formation. The effectiveness of this methodology is validated by comparing it with five baseline strategies. The proposed model significantly outperforms these strategies in terms of annual cumulative return, Sharpe ratio per three-year period, and average monthly return relative to risk over each three-year period, demonstrating superior potential returns and risk management. Ma et al. [34] integrated return prediction into

portfolio formation by utilizing two machine learning models—Random Forest and SVR—along with three deep learning models: LSTM, Deep Multilayer Perceptron (DMLP), and CNN. Specifically, it first applies these prediction models for stock preselection prior to portfolio formation. The predictive results are then used to enhance the mean-variance and omega portfolio optimization models.

2) Problems of index investing: The introduction of index mutual funds in the 1970s, followed by the rapid growth of exchange-traded funds (ETFs) in the 2000s, made it cheaper for ordinary investors to own well-diversified portfolios. This development had two significant consequences. First, many investors who previously held individual stocks switched to passive indexing to reduce transaction and asset management expenses. Second, the affordability of index funds allowed numerous households who had not previously invested in stocks to enter the equity market [35]. But in spite of this, index investing often yields lower returns compared to actively managed portfolios or strategic asset pre-selection methods. This is primarily because index funds aim to replicate the performance of a market index rather than outperform it. Consequently, they are limited by the underlying index's returns, which may not capture high-growth opportunities or effectively manage risks through selective asset allocation. As a result, investors seeking higher returns and better risk-adjusted performance may find more success with approaches that involve active management and careful pre-selection of high-potential assets. However, actively managed funds have a very high expense ratio, which makes them non-feasible and less attractive. Accordingly, there is growing research on combining forecasting theory with portfolio optimization.

C. Research Gap

The effectiveness of portfolio construction largely hinges on the anticipated performance of stock markets. Traditional portfolio theory often relies heavily on expected returns and neglects future information [4]. Advances in machine and deep learning have introduced substantial opportunities to integrate predictive analytics into portfolio selection. Despite this potential, the concept of prediction-based portfolios has been underexplored in academic research, with notable contributions like the one by [30] standing out. Consequently, integrating deep learning predictions to assist in selecting the best investment strategies represents a valuable and promising avenue for future research [36]**.** Many researchers [31], [32], [37], [38], [39] have applied these models in the stock pre-selection process prior to portfolio formation and achieved promising and satisfying results. However, to the best of the authors' knowledge, there are no studies existing with regard to the Indian stock market. We extend the work by [9] where GA was used to select stock portfolios in six different Asian markets, excluding India. This is the first study of its kind, and consequently, it is believed that this paper is making a substantial contribution to the related literature.

III. METHODOLOGY

The following sections provide details on the methodology of this study. The study is undertaken in two stages.

A. Prediction model for NIFTY 50

In the first stage, this study aims to build an LSTM-based prediction model for the NIFTY 50 stocks.

1) Data description: Financial time-series forecasting is always explained by historical values or lagged observations [40]. Hence, this dataset consists of historical values of NIFTY stocks spanning 12 years taken from the official NSE website[1]. The study opts for the NSE over the BSE because of its larger size and greater market participation. The Nifty 50 is the primary index of the National Stock Exchange, encompasses 50 diversified stocks across 13 sectors of the economy, and represents the country's leading blue-chip companies. The selected sample includes liquid stocks from various sectors and sizes, thus minimizing sample bias and avoiding concentration on a specific group of stocks.

The values included six features- Open, High, Low, Close (OHLC), adjusted close, and volume. Kumar et al. [41] conducted a survey on stock market forecasting using computational techniques and reported that only 8% of the studies used a combination of historical values and technical indicators for prediction purposes. Hence, this study uses a synthesis of historical data and STIs as the predictor or input variables. The output or target variables are the close values of subsequent days. The total dataset consists of daily trading data of 2,956 trading days (April 2012 – March 2024). This data covers two stock market crashes, the crypto crash in 2018 and the 2020 COVID crisis, and the Russia-Ukraine war in 2022 so that the extreme volatilities of the assets can be considered for optimal portfolio construction. This research chooses a sliding window approach of a 30-day time frame [42], [43] for developing the prediction model. This implies that data from the preceding 30 days will be utilized to forecast close values for the 31st day. Pandas library is used to import data. Matplotlib and Seaborn are used for data visualizations.

2) Data pre-processing: The accuracy of predictions significantly depends on the quality of the data. Therefore, it is vital to preprocess the raw data before incorporating it into the model-building process. The collected sample of 2956 trading days was removed from duplicates and NAN values. The cleaned dataset consisted of 2906 trading days. The outliers identified using a boxplot are treated using the winsorization technique [44], [45], [46]. Winsorization helps to eliminate outliers by capping extreme values, thereby making the distribution of the transformed data more symmetrical and closer to a normal distribution [45], [47]. This data is then normalized using a min-max scalar. Data normalization involves transforming real numerical attributes to a scale between 0 and 1, resulting in a training model that is less

[1]https://www.nseindia.com/reports-indices-historical-index-data

affected by the variable scales [26]. This also ensures that all values fall within a range of [0,1], thus leading to faster convergence. Normalization is very useful for improving the accuracy of neural network models [43]. The equation for normalization is as follows.

$$X\_scaled = \frac{\text{X}-\text{XMin}}{XMax-XMin} \quad (1)$$

X is the feature's initial value, *Xmin* is the lowest X value, *Xmax* is the highest X value, and *X_scaled* is the new scaled X value between 0 and 1.

3) Proposed model

a) STI: Statistical Technical Indicators (STIs) are mathematical calculations based on factors like price, volume, or other relevant metrics related to stocks, securities, or contracts. Unlike fundamental analysis, they do not take into account business fundamentals such as earnings, revenue, or profit margins. The primary goal of technical analysis is to predict future price movements, and deep learning algorithms enhance the accuracy of these predictions. By combining these two approaches, the reliability of price forecasts can be significantly strengthened. While technical indicators are essential for identifying stock price patterns, trends, and momentum, it's important to note that many studies limit their use to trend indicators, often overlooking key momentum, volatility, and strength indicators that are equally crucial for comprehensive financial analysis [22]. Hence, this study uses a total of ten momentum and volatility STIs as identified by [46] and is calculated through Ta-Lib. They are listed in Table I.

b) LSTM: Hochreiter et al. [51] introduced LSTM in 1997 to address the issue of vanishing gradients in conventional RNNs, which hindered their ability to capture long-term relationships in sequential data effectively. This problem occurs because gradients tend to become smaller and smaller as they propagate back through time, making it difficult for the network to update the weights in earlier layers. The main advantage of LSTMs over traditional RNNs is their capability to choose whether to remember or forget information from earlier time steps, enabling them to handle long-term dependencies more effectively. This is achieved through a set of specialized memory cells and gating mechanisms that balance information flow through the network. "An LSTM layer is composed of one or more LSTM units, and an LSTM unit consists of cells and gates to perform classification and prediction based on time series data" [49]. The cell contains three gating mechanisms: the input gate *i*, the output gate *o*, and the forget gate *f*. The quantity of new information added to the cell state is dictated by the input gate, the amount of old information that is discarded from the cell state is regulated by the forget gate, and the quantity of information that is transferred from the cell state to the next time step is controlled by the output gate [26]. The cell state is the memory of the LSTM and can be thought of as a conveyor belt that runs through the entire LSTM chain, enabling the transmission of information from one time step to the subsequent one.

This study uses a double-layered LSTM since deeper LSTM architecture is known to yield superior prediction outcomes compared to a single LSTM network [50]. Furthermore, since the input variables are 18 in total, a PCA is conducted for dimensionality reduction.

c) Working of the stacked LSTM model: The input data for the LSTM model is organized into a three-dimensional array, where each dimension captures a different aspect of the data. The time dimension corresponds to the sliding time window, which is used to capture temporal dependencies in the data. The sample dimension represents the size of the dataset used for training and testing the model. Finally, the feature dimension indicates the number of input features provided to the LSTM model, allowing it to process multiple attributes simultaneously for more accurate predictions. This study chose 30 days as a time window, and the input features are the OHLC, volume, and STI values reduced as PC's through PCA. The input layer is linked to the LSTM layer with 32 neurons, which is the hidden layer. This is connected to another LSTM with 16 neurons. Two LSTM layers are stacked sequentially, where the output of the first LSTM layer serves as the input to the second LSTM layer. This stacking enables the model to better capture and process sequential patterns and long-term dependencies in the data. The second LSTM layer is then connected to a dense layer with a single neuron, which serves as the output layer for making predictions, such as forecasting stock prices or classifying trends. This setup allows the model to refine complex temporal relationships and generate more accurate results.

Dropouts and Early stopping are used as regularization techniques. The idea behind dropout [51] is to randomly "drop out" or disable some of the neurons in a layer during each training iteration. This is done by setting the output of some of the neurons to zero with a certain probability (usually around 0.5). The exact neurons that are dropped out are randomly selected during each training iteration. The reason for the dropout is to prevent the neural network from depending too much on any particular set of neurons. By randomly dropping out neurons during training, the network is forced to learn more robust and generalized useful features across a wider range of inputs. At test time, the full network is used without any dropout. However, the output of each neuron is multiplied by the dropout probability to ensure that the expected output of each neuron is the same as during training. A dropout rate of 0.2 is used in this case. EarlyStopping is a Keras callback that allows you to stop training when a monitored quantity (like validation loss or accuracy) has stopped improving. It helps avoid overfitting by terminating the training process once the model performance on the validation set no longer improves. This ensures that the model does not waste time training for too many epochs, which can lead to overfitting. Here, it is configured to monitor the validation loss and stop training if accuracy does not improve for 8 consecutive epochs.

The model is initialized with random weights and biases. Each LSTM layer receives an input consisting of the previous 30 time steps and attempts to predict the next time step in the

sequence, which corresponds to the closing value for the 31st day; this output from the LSTM is passed on to the dense layer, which gives the final output values. ReLU and Linear activation functions are utilized in the hidden and output layers, respectively. Mean Squared Error (MSE) is used as the loss function. Each LSTM network computes its individual loss, and the total loss is calculated by summing the losses from both LSTM networks and the dense layer. The Adam optimization algorithm is used during training to minimize this total loss, and the number of epochs is set to 100, following the approach used by [2]. Optimizers are techniques utilized to adjust the model's features, including parameters like learning rate and weights, to minimize losses. An epoch represents one complete pass of the entire training dataset [26] by the LSTM model. The training process continues until the validation loss stops improving for a specified number of epochs, as determined by early stopping, or until the maximum number of iterations is reached.

TABLE I. LIST OF STIS

STI	Description	Indicator Type
MACD	Moving average convergence divergence	Momentum
RSI	Relative strength index	Momentum
STOCH	Stochastic Oscillator	Momentum
CCI	Commodity Channel Index	Momentum
ADX	Average Directional Index	Momentum
ROC	Rate of change	Momentum
WILLR	William percent R	Momentum
ATR	Average True Range	Volatility
NATR	Normalized Average True Range	Volatility
TRANGE	True Range	Volatility

Source:[46]

The model's parameter settings are established through trial-and-error experiments to optimize performance. The implementation is carried out in a Python 3.7 environment using Keras, a high-level API built on Google's TensorFlow framework. Table II provides an overview of the parameter settings used in the experiments.

B. Portfolio Optimization

1) Assumptions: In this study, several key assumptions are necessary for the analysis of portfolio performance. Although these assumptions may seem idealistic compared to real-world conditions, they serve to simplify the complexities associated with investment, such as costs and trading prices. By making these assumptions, the study can focus more effectively on comparing the relative performance of portfolios.

- No transaction cost and tax.
- The stocks are sold and bought at the closing price.
- Investors are not risk-averse.
- A 3-year average return on treasury bills is considered as risk-free return.

Once an LSTM-based prediction model has been developed for predicting close values, the next stage involves the preselection of assets for portfolio creation and optimization. Since the test period consists of approximately three years, datasets from April 2021 – March 2024 (Test period) are considered for portfolio creation as well. Average returns for a period of three years are calculated for all 50 stocks based on the predicted close values. Then, they are arranged in descending order of their predicted returns to select the top return-generating stocks.

2) Top N stocks: Several studies have shown that holding too many stocks makes it difficult to manage and keep them under control, particularly for individual investors. Building a portfolio with fewer than ten stocks is taken into consideration in several portfolio optimization research [52]. A portfolio with an average of seven stocks performs better than other portfolios with a variable number of stocks, according to [37]. Wang et al. [3] noted that a portfolio with ten stocks is ideal, as it outperforms portfolios with any other number of stocks. According to [40], the optimal number of stocks for an individual investor's portfolio construction is seven. As a result, this study selected a group of the top 10,9,8 and seven stocks for portfolio creation. Six different objectives-based portfolios are created. This means that for each objective, five different portfolios with the number of stocks N =10,9,8 and 7 and, all NIFTY stocks with N=50 are formed to evaluate the performance of different-sized portfolios.

TABLE II. PARAMETER SETTINGS

Parameters	Values
Optimizer	Adam
Epochs	100
Batch size	64
Step size	30
Drop out	0.2
Activation function	ReLU (Hidden layer) Linear (Output layer)

Source:[46]

3) Portfolio objectives

- Objective 1: Minimum volatility portfolio
- Objective 2: Maximum returns portfolio
- Objective 3: Maximum Sharpe ratio with No Constraints
- Objective 4: Maximum Sharpe ratio with Constraints [L2 regularizer, gamma=2, $\sum W = 1$]
- Objective 5: Uncorrelated assets portfolio
- Objective 6: Equally weighted portfolio

4) Comparison criteria: The portfolios created are compared on the following metrics of return and risk [39], [56], [57].

a) Sharpe ratio: The Sharpe ratio is a widely used industry standard for assessing investment risk adjustment return in finance. It is computed by deducting the risk-free investment return from the stock or investment portfolio's actual return and dividing the result by the stock or portfolio's standard deviation.

$$\text{sharpe ratio} = \frac{R_p - R_f}{\sigma_P} \quad (2)$$

where, R_P is the Return of the portfolio, R_f is the Risk-free rate, σ_P is the Standard deviation of the portfolio. A Sharpe ratio >1 is always preferred, implying that the investment generates one unit of excess return for every unit of risk taken.

b) Sortino ratio: Sharpe ratio considers both upside and downside risks since standard deviation calculates deviation from mean returns. This deviation could either be positive or negative. Conversely, the Sortino ratio [55]—a variant of the Sharpe ratio—only accounts for negative or downward volatility. Upside volatility is generally considered a benefit of investing and is not dangerous [55]. Consequently, the total standard deviation in the Sharpe ratio is replaced by this downside risk or volatility in the Sortino ratio. A higher Sortino ratio is always desired by the investor since it indicates the return per unit of downside risk.

$$sortino\ ratio = \frac{R_p - R_f}{DR} \quad (3)$$

where, R_P is the Return of the portfolio, R_f is the Risk-free rate and DR is the Downside risk.

c) Cumulative returns: Cumulative returns measure the total growth of an investment from the start to the end of a given period. It represents how much a portfolio has increased in value, assuming daily compounding of returns. The study uses the *cumprod()* function in Python. This cumulative product function calculates the running product of the growth factors or daily growth over the entire period. This effectively compounds the daily returns, simulating how an investment grows day by day.

d) Annual returns (CAGR): Annual returns convert total cumulative growth into an annual growth rate. It accounts for the effect of compounding, showing the consistent annual rate that equates to the total growth observed. The study uses 252 Trading Days as the standard number of trading days annually to annualize the returns.

$$\text{Annual return} = (1 + \text{Cumulative return})^{252/\text{ Trading days}} - 1 \quad (4)$$

Here, the number of trading days is calculated as 252 days* 3 years

e) Volatility: Volatility is a measure of the risk factor of the portfolio and is calculated as the standard deviation of the portfolio's returns by using the covariance matrix of the asset returns and the optimized portfolio weights.

f) Beta: Beta is the measure of the systematic risk of a portfolio or stock. It indicates the relative volatility of a portfolio as against the benchmark or index. It helps investors understand how much risk they are taking in comparison to the market. Investors looking for higher returns with higher risk might prefer high-beta stocks, while those seeking stability might opt for low-beta stocks.

The study used the PyportfolioOpt package in Python to create these portfolios. This package generated weights based on the given objective functions. For every N number of stocks, six different portfolios corresponding to each objective are formed. After the weights have been assigned, portfolio returns are calculated based on these allotted weights and the market returns, which are NIFTY returns in this case. Next, *cum.prod()* function is used to calculate cumulative returns and CAGR is calculated from that value. Volatility is measured as the standard deviation of the returns. The study considered a rate of 6.85% as the average risk-free rate of the past three years and 252 as the average trading days for calculating Sharpe and Sortino ratios.

IV. Experimental Results and Discussion

A. Generation of PC's

In the first experiment, the study attempts to develop a stacked LSTM for the prediction of NIFTY 50 stocks. Historical values and STIs of the past 30 days are used as input variables to predict the close values of the 31st day. Since there are 18 input variables and 12 years of data, as mentioned in the methodology section, the study uses PCA to generate PCs to reduce dimensionality. The PCA run for all the 50 stocks generated 5 or 6 PCs. To evaluate the variance associated with each PC, the explained variance ratio was calculated. This ratio is obtained by dividing the variance of each component by the total variance. The variance ratio of each PC of the top 5 NIFTY stocks, in terms of market capitalization, is explained in Table III. It is clear that more than 95% of the information was preserved even after PCA.

The number of generated PCs determines the architecture of the stacked LSTM. Since the number of PC's are 5 and 6, the input layer will also have the same number of nodes. The architecture of the developed LSTMs is illustrated in Fig. 1.

B. Learning Curves

After obtaining the PCs, the datasets are separated as training data (70%; 2,035 days), validation data (10%;291 days), and testing data (20%; 580 days) to train, validate and test the proposed LSTM.

The study employed training and validation curves to chart the model's performance, indicated by loss (MSE), on both the training and validation datasets across epochs. During training, the model aims to minimize the loss function, which measures the disparity between the predicted values and the actual values in the training dataset. Fig. 2 illustrates the learning curve of the LSTM model applied to the top five stocks, The X-axis represents the number of training epochs, and the Y-axis represents the loss value, indicating the model's error in prediction.

TABLE III. Explained Variance Ratio

PCs	HDFC	RIL	ICICI	INFY	L&T
0	0.5912883	0.5722852	0.5063156	0.5477141	0.5843656
1	0.1384574	0.1520931	0.1390794	0.1500667	0.1258094
2	0.08379	0.0760954	0.1303963	0.0789791	0.1029366
3	0.0500956	0.0707812	0.0786461	0.0639012	0.0575905
4	0.0427779	0.0428246	0.0574367	0.0537327	0.0469288
5	0.0384285	0.0343057	0.0338025	0.0424565	0.0334498
6	0.0261741	0.0154152	0.0179384	0.0298618	
Total	0.9710117	0.9638004	0.9636151	0.9667119	0.9510807

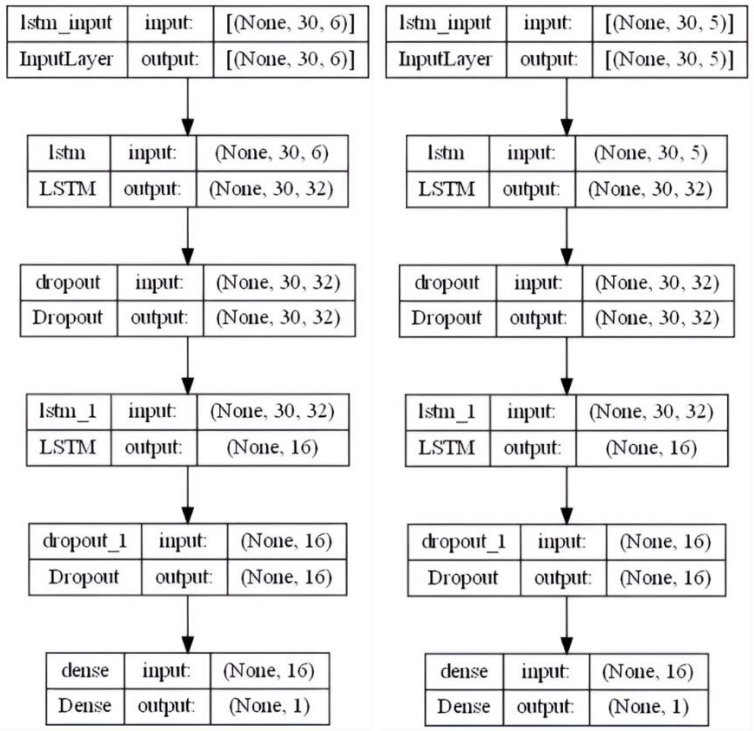
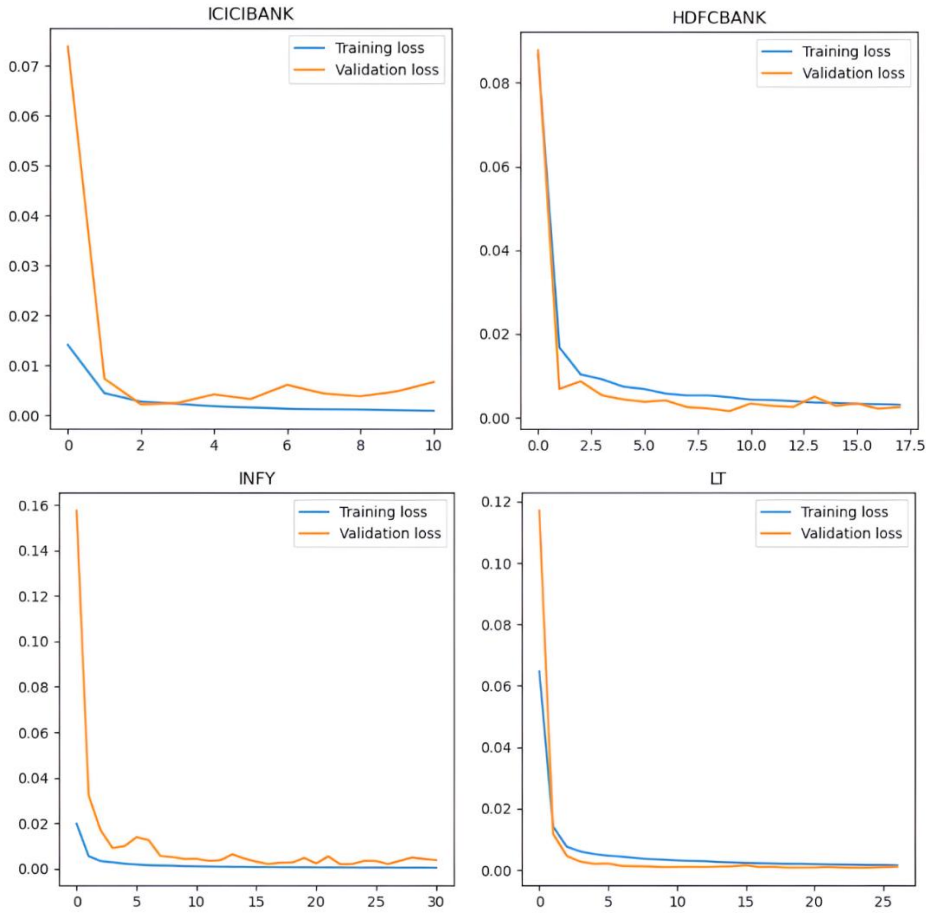


Fig. 1. Architecture of the LSTM.



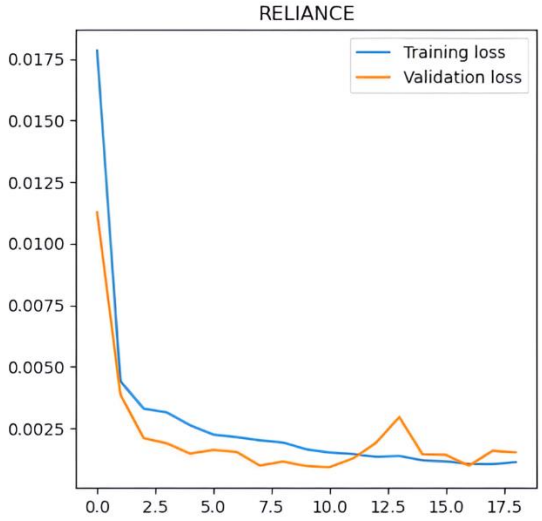


Fig. 2. Learning curves.

The study made the following observations

- Good Generalization: The learning curves for both training and validation loss decrease rapidly at the start and remain closely aligned throughout the training. The parallel behavior of the training and validation losses suggests that the model is generalizing well and not overfitting or underfitting the training data.
- Convergence: Both curves stabilize and converge to a low value towards the end of the epochs, indicating that the model has reached a point where additional training does not significantly change the loss, demonstrating a well-trained model.

C. Metric Evaluation

Appropriate assessment metrics are needed to validate the deep learning models. This study chooses the following indicators: Mean of absolute error (MAE) and root mean square error (RMSE). They have been extensively used in the literature [56], [57], [58]. Lower MAE and RMSE values would increase the prediction accuracy. Accuracy is measured as [58]. The equations are as follows.

$$\text{PM}\Sigma\text{E}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}|yi\ -\ \hat{y}i|^{\ 2}} \tag{5}$$

Where n is the number of data points, yi is the actual value of the i^{th} data point, ŷi is the predicted value of the i^{th} data point, Σ represents summation, | | represents absolute value.

$$\text{A}\chi\chi\upsilon\rho\alpha\chi\psi = 1-\text{MAE} \tag{6}$$

$$\text{MAE}=\frac{1}{n}\sum_{i=1}^{n}|yi\ -\ \hat{y}i| \tag{7}$$

Results show that the model obtained an average accuracy of 90.58%, RMSE of 0.1057, and MAE of 0.0942. ADANIENT, from the Metals and Mining Industry, and EICHERMOT, belonging to the Automobile industry, recorded the lowest and highest accuracies of 53.12% and 97.33%, respectively. Industry-wise analysis shows that the Oil, Gas & Consumable Fuels industry has the highest and Metals and Mining has the lowest accuracy.

D. Asset Preselection

In the second experiment, the focus is on creating and optimizing stock portfolios. The process involves filtering the top stocks based on their average returns over a three-year test period. The returns are calculated on the basis of their adjusted closing values. Table IV shows that the top 10 stocks include COAL INDIA, which achieved the highest average return at 38.72%, followed by SUN PHARMA with 25.22%, and ONGC with 23.83%.

Interestingly,30% of these stocks are in the energy sector (COALINDIA, BPCL, ONGC) and another 30% in the automobile sector (HEROMOTOCO, EICHERMOT, BAJAJ-AUTO), with 20% in banking and financial services (INDUSINDBK, AXIS BANK) and 10% each in pharmaceuticals (SUN PHARMA) and FMCG(ITC), despite financial services (37.72%) and information technology (14.11%) constituting the majority share in NIFTY50.

E. Portfolio Formation

This study created portfolios with the number of stocks N= 10, 9, 8 and 7. With each N, six objectives-based portfolios are created. For optimization purposes, the PyportfolioOpt package is used to achieve optimum weight allocation in each scenario. These portfolios are compared on the basis of the six key factors: Sharpe ratio, Sortino ratio, Annual returns, Cumulative returns, Annual volatility, and Beta. Furthermore, optimal portfolios are also compared against the market benchmark NIFTY 50 and portfolio with N=50 to evaluate the performance of portfolios with and without asset preselection.

1) Performance of different-Sized portfolios: For Objective 1, the top-performing portfolios are analyzed with a focus on their respective metrics. The top 7 portfolios stand out with a Sharpe ratio of 1.75 and a Sortino ratio of 3.15, indicating high risk-adjusted returns and effective downside risk management. These portfolios achieve an annual return of 29.67% and a cumulative return of 114.27%.

TABLE IV. AVERAGE RETURNS OF TOP 10 STOCKS

Stock	Annual returns
COALINDIA	38.7258
SUN PHARMA	25.2218
ONGC	23.8309
INDUSINDBK	22.3735
AXIS BANK	22.164
ITC	21.5993
HEROMOTOCO	19.5488
EICHERMOT	17.0569
BAJAJ-AUTO	16.6815
BPCL	15.8402

TABLE V. OBJECTIVE-WISE ANALYSIS

OBJECTIVE 1

	Sharpe ratio	Sortino ratio	Beta	Annual Volatility	Annual Returns %	Cumulative Returns
Top 7	1.75	3.15	0.75	14	29.67	114.27
Top 8	1.72	3.13	0.76	13.9	29.02	111.13
Top 9	1.93	3.41	0.74	13.6	31.41	122.78
Top 10	1.87	3.33	0.74	13.5	30.53	118.46
All 50	1.46	3.04	0.63	10.26	20.62	73.32

OBJECTIVE 2

	Sharpe ratio	Sortino ratio	Beta	Annual Volatility	Annual Returns	Cumulative Returns
Top 7	1.63	3.35	0.82	20.51	46.43	205.99
Top 8	1.65	3.40	0.80	17.79	41.48	176.69
Top 9	1.51	3.14	0.91	19.58	41.32	175.76
Top 10	1.44	3.06	0.90	19.41	38.83	162.01
All 50	1.15	2.52	0.99	14.57	25.29	93.69

OBJECTIVE 3

	Sharpe ratio	Sortino ratio	Beta	Annual Volatility	Annual Returns	Cumulative Returns
Top 7	2.35	3.93	0.72	17.2	44.51	194.41
Top 8	2.35	3.93	0.72	17.2	44.51	194.41
Top 9	2.51	4.17	0.72	15.7	44.03	191.57
Top 10	2.51	4.17	0.72	15.7	44.03	191.57
All 50	2.73	4.47	0.73	14.3	43.85	190.49

OBJECTIVE 4

	Sharpe ratio	Sortino ratio	Beta	Annual Volatility	Annual Returns	Cumulative Returns
Top 7	2.14	3.41	0.81	18	42.06	179.99
Top 8	2.11	3.39	0.81	17.7	40.85	173.06
Top 9	2.24	3.61	0.79	16.4	40.9	173.31
Top 10	2.2	3.55	0.80	16.3	39.86	167.48
All 50	2.08	3.32	0.94	15.5	36.62	149.69

OBJECTIVE 5

	Sharpe ratio	Sortino ratio	Beta	Annual Volatility	Annual Returns	Cumulative Returns
Top 7	1.66	3.6	0.67	15.83	37.47	154.32
Top 8	1.6	3.44	0.71	15.29	35.1	141.65
Top 9	1.89	3.91	0.72	15.78	42.47	182.37
Top 10	1.73	3.55	0.80	16.09	34.6	165.55
All 50	1.19	2.64	0.92	13.88	25.07	92.72

OBJECTIVE 6

	Sharpe ratio	Sortino ratio	Beta	Annual Volatility	Annual Returns	Cumulative Returns
Top 7	1.78	2.83	0.89	15.92	31.06	121.08
Top 8	1.7	2.76	0.90	15.7	28.98	110.96
Top 9	1.8	2.94	0.88	15.27	30.11	116.39
Top 10	1.74	2.79	0.88	15.18	28.75	109.82
All 50	1.44	2.23	0.95	13.8	20.77	73.91

Expanding the portfolio to the top 9 increases the Sharpe ratio to 1.93 and the Sortino ratio to 3.41, with annual returns rising to 31.41% and cumulative returns to 122.78%, suggesting that including more assets up to this point optimizes performance. However, the top 10 portfolios show a slight decrease in the Sharpe ratio to 1.87 and Sortino ratio to 3.33, with annual returns at 30.53% and cumulative returns at 118.46%, indicating a slight reduction in performance efficiency. The beta values remain relatively low across these portfolios, suggesting they maintain a lower level of market risk while delivering strong returns.

Under Objective 2, the top 7 portfolios have a Sharpe ratio of 0.63 and an impressive Sortino ratio of 3.35, indicating excellent management of downside risk. These portfolios achieve the highest annual returns of 46.43% and cumulative returns of 205.99%. The top 8 portfolios, with a higher Sharpe ratio of 1.65 and Sortino ratio of 3.40, offer annual returns of 41.48% and cumulative returns of 176.69%, providing better overall risk-adjusted performance. The top 9 portfolios show a slight decrease in the Sharpe ratio to 1.51 and Sortino ratio to 3.14, with annual returns at 41.32% and cumulative returns at 175.76%, suggesting a minor decline in performance. The top 10 portfolios further decrease in performance with a Sharpe ratio of 1.44 and Sortino ratio of 3.06, achieving annual returns of 38.83% and cumulative returns of 162.01%. The beta values indicate that these top portfolios are slightly more volatile, which aligns with their higher returns.

For Objective 3, the portfolios excel in risk-adjusted returns and overall performance. The top 7 and top 8 portfolios share the highest Sharpe and Sortino ratios of 2.35 and 3.93, respectively, achieving annual returns of 44.51% and cumulative returns of 194.41%. Interestingly, the top 9 and top 10 portfolios, with slightly higher Sharpe ratios of 2.51 and Sortino ratios of 4.17, deliver similar annual returns of 44.03% and cumulative returns of 191.57%, suggesting that an increase in the number of assets does not significantly impact performance. The beta values are slightly higher, reflecting moderate volatility but efficient risk management.

In Objective 4, the top 7 portfolios exhibit a high Sharpe ratio of 2.14 and a Sortino ratio of 3.41, with annual returns of 42.06% and cumulative returns of 179.99%. The top 8 portfolios maintain a Sharpe ratio of 2.11 and a Sortino ratio of 3.39, with annual returns of 40.85% and cumulative returns of 173.06%. Expanding to the top 9 portfolios slightly increases the Sharpe and Sortino ratios to 2.24 and 3.61, respectively, while maintaining annual returns of 40.9% and cumulative returns of 173.31%. The top 10 portfolios show a slight decrease in performance with a Sharpe ratio of 2.2 and Sortino ratio of 3.55, achieving annual returns of 39.86% and cumulative returns of 167.48%. The beta values indicate that these portfolios are more volatile but compensate with higher returns, reflecting efficient risk management.

Objective 5 portfolios show significant variation in performance metrics. The top 9 portfolios stand out with a higher Sharpe ratio of 1.89 and a Sortino ratio of 3.91, achieving the highest annual returns of 42.47% and cumulative returns of 182.37%. The top 7 portfolios have a Sharpe ratio of 1.66 and a high Sortino ratio of 3.6, with annual returns of 37.47% and cumulative returns of 154.32%. The top 8 portfolios maintain a Sharpe ratio of 1.6 and Sortino ratio of 3.44, achieving annual returns of 35.1% and cumulative returns of 141.65%. The top 10 portfolios show a slight decline in performance with a Sharpe ratio of 1.73 and Sortino ratio of 3.55, achieving annual returns of 34.6% and cumulative returns of 165.55%. The beta values suggest moderate volatility, consistent with the achieved returns.

For Objective 6, the performance metrics highlight a steady trend. The top 7 portfolios achieve a Sharpe ratio of 1.78 and a Sortino ratio of 2.83, with annual returns of 31.06% and cumulative returns of 121.08%. The top 8 portfolios maintain a Sharpe ratio of 1.7 and Sortino ratio of 2.76, achieving annual returns of 28.98% and cumulative returns of 110.96%. The top 9 portfolios show a slight increase in the Sharpe ratio to 1.8 and Sortino ratio to 2.94, with annual returns of 30.11% and cumulative returns of 116.39%. The top 10 portfolios exhibit a Sharpe ratio of 1.74 and Sortino ratio of 2.79, achieving annual returns of 28.75% and cumulative returns of 109.82%. The beta values indicate that these portfolios are relatively more volatile, reflecting their higher returns. Overall, the top portfolios under Objective 6 maintain good performance with balanced risk management despite diminishing returns with larger portfolio sizes. Fig. 3 shows a performance comparison of the best portfolios from each objective.

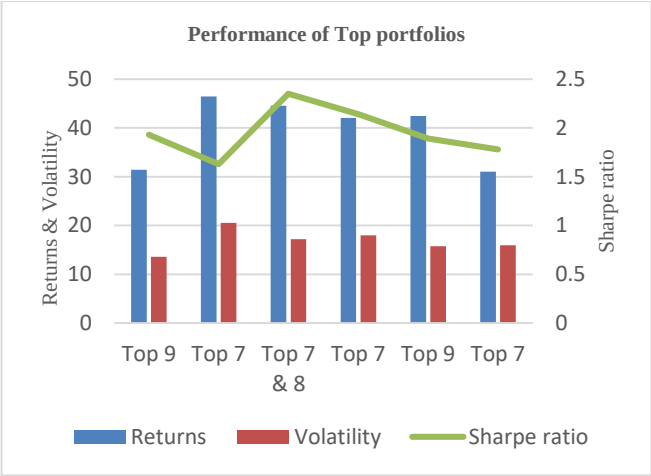


Fig. 3. Performance comparison of Top portfolios.

2) Best performing portfolio:'Investors in the financial markets rarely pursue a risk-minimization approach; instead, they are more than willing to take on more considerable risks if the accompanying profits are even higher' [59]. Consequently, the study considers returns as the primary criteria for selecting the best portfolio. Though the top 7 stocks under objective 2 (maximum returns) generated the highest return of 46.43%, the volatility is 20.51%, which is higher than any other portfolio. Therefore, based on the detailed analysis across various objectives, the portfolio that consistently delivered the best performance is the top 9 and 10 portfolios under Objective 3, which is the maximum Sharpe ratio with no constraints. Though nine stocks were initially added, the weights were allocated to only 5 stocks [Bajaj Auto, Coal India, ITC, ONGC, and Sun pharma] and the rest were given zero weights. Weights were allocated in such a way that it maximizes the objective function of the Sharpe ratio. Hence, effectively only five stocks constituted the optimal portfolio.

Fig. 4 demonstrates the weight allocation for portfolio construction. It is also interesting to note that these 5 selected stocks are not the top 5 return-generating stocks but are randomly selected by the software.

Portfolio performance metrics:

- Sharpe Ratio: 2.51
- Sortino Ratio: 4.17
- Beta: 0.72
- Annual Volatility: 15.7%
- Annual Returns: 44.03%
- Cumulative Returns: 191.57%

The high ratios indicate that this portfolio achieved the best risk-adjusted returns and managed downside risk effectively.

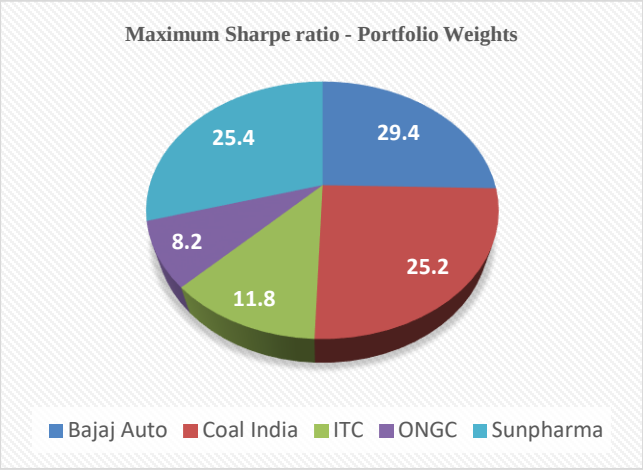


Fig. 4. Weight allocation of optimal portfolio.

The beta value indicates a moderate level of market risk, suggesting that the portfolio is not overly volatile while still capturing substantial returns. Annual Returns of 44.03% and Cumulative Returns of 191.57% reflect the highest annual and cumulative returns, demonstrating the portfolio's superior performance over the period. Hence, the study concludes that a five-stock portfolio provides the best performance across all analyzed metrics. They achieve the highest risk-adjusted returns, manage downside risk effectively, and deliver the highest annual and cumulative returns.

3) Pre-selection V/s All 50: Table V shows a comparison of portfolios constructed after pre-selection and without pre-selection. The pre-selected stocks consist of top 7,8,9, and 10 stocks from NIFTY 50, whereas the alternate portfolios consist of all NIFTY 50 stocks. It is evident that the former achieved better returns than the latter in terms of annual returns. Fig. 5 shows the excess returns earned by the top-performing portfolios as compared to NIFTY all 50 stocks from each objective. The optimal portfolios obtained an excess return of 10.79%, 21.14%, 0.66%,5.44 %, 17.4 %, and 10.29 %. For Objective 3, the portfolio aimed to maximize the Sharpe ratio without constraints but only achieved a 0.66% outperformance in annual returns. This underperformance suggests that the unconstrained approach might have led to a skewed portfolio, possibly concentrating too heavily on high-risk stocks. While the goal was to achieve the best risk-adjusted returns, the lack of risk controls likely reduced the portfolio's ability to generate higher excess returns. In contrast, Objective 4, with constraints, achieved a much higher excess return of 17.4%, highlighting the importance of controlled risk management when optimizing for the Sharpe ratio. Nevertheless, the top portfolios earned an average excess return of 10.95% as against NIFTY all 50 portfolios. Thus, the experiment proves that pre-selection of your assets can help better your investment fortunes than too much diversification. Over-diversification reduces your risk but also brings down one's returns.

4) Superiority over benchmark models: The study evaluated the performance of the proposed portfolio optimization method against the NIFTY 50 index, a well-established market benchmark. This benchmark includes a diverse range of leading companies in the Indian stock market. By using the NIFTY 50 as a reference, the study can compare the proposed portfolio allocations to the performance of a globally recognized benchmark index.

a) All 50 v/s Index: This analysis involves the creation of six different portfolios, each based on a distinct objective, utilizing all NIFTY 50 stocks. These portfolios were then compared to the NIFTY Index in terms of returns and volatility. The results demonstrate that the newly constructed portfolios significantly outperformed the NIFTY Index (Fig. 6). Specifically, the portfolios showed higher returns, with "All 50 (3)" achieving the highest return of 43.85% and "All 50 (4)" following with 36.62%. Even the lowest-performing portfolio, "All 50 (1)," delivered a return of 20.62%, surpassing the NIFTY Index's return of 14.85%. In terms of volatility, while some portfolios exhibited higher volatility than the NIFTY Index (13.75%), such as "All 50 (4)" with 15.5% and "All 50 (2)" with 14.57%, others managed to maintain lower or

comparable volatility levels, like "All 50 (1)" with 10.26%. This indicates that the newly created portfolios not only provided superior returns but also effectively managed risk, outperforming the NIFTY Index overall.

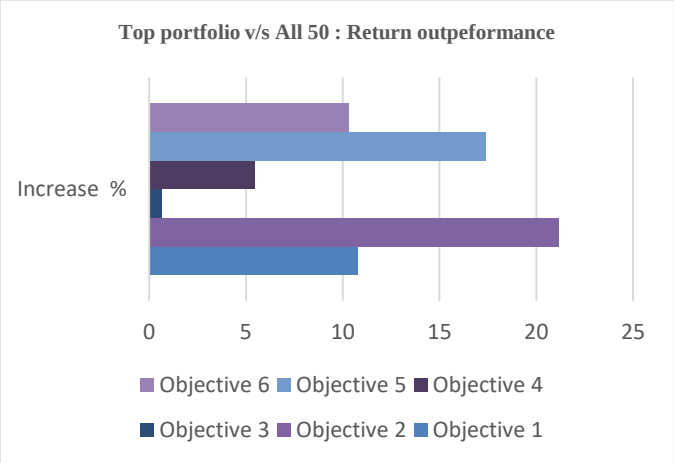


Fig. 5. Return outperformance of top portfolios.

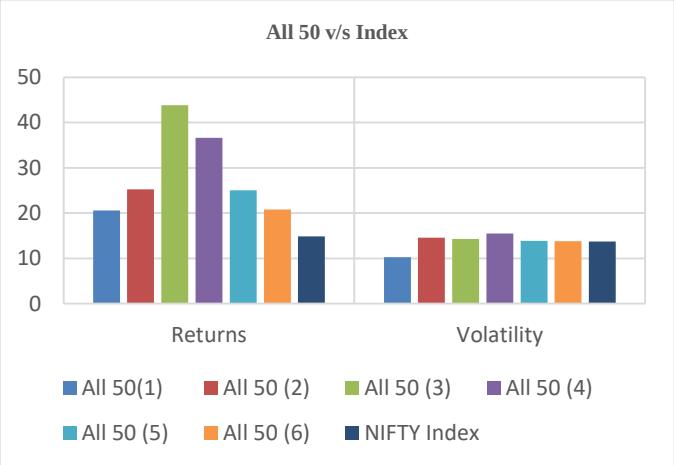


Fig. 6. Comparison of returns and volatility.

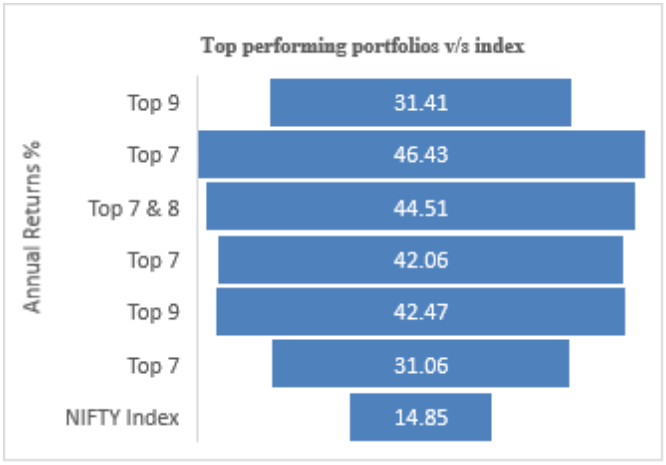


Fig. 7. Performance comparison of top portfolios and NIFTY index.

b) Top performing portfolios v/s index: This analysis compared the annual returns of top-performing portfolios from each objective with the NIFTY index returns over the past three years. Fig. 7 shows that the top 7 portfolios in objective 2 could generate returns as high as 46%, an excess return of 31.58% more than the NIFTY index. Even the portfolio with the lowest returns of 29.02% (Top 8 in objective 1) earns 14.7 % more than the benchmark index, demonstrating consistent and notable outperformance across different strategies. All the top-performing portfolios earn an average excess return of 27.51%. The analysis conclusively demonstrates that the top-performing portfolios provide superior returns and significantly exceed the NIFTY index's benchmark performance. This suggests that the strategies employed in these portfolios are highly effective, yielding substantial excess returns even in the least performing portfolio among the top contenders. Investors may find these strategies attractive for achieving higher returns compared to the standard benchmark, emphasizing the value of strategic portfolio selection and management.

F. Discussion

This work aims to extend the existing literature on deep learning-based prediction and asset pre-selection for portfolio optimization. An LSTM-based prediction model was developed using data from 12 years of historical and technical indicators. This model was applied to forecast NIFTY stocks for a test period of 580 days. Then, the predicted returns were used to filter the top ten assets for portfolio creation. The major findings of the study are:

- Results show that the model obtained an average accuracy of 90.58%, RMSE of 10.57%, and MAE of 9.42%. The authors compare these results to [62],[63], which reported an accuracy of 72% and 90%, respectively.
- This study attempted to select high-quality assets from the Indian stock market through deep learning-based forecasting and build a competitive portfolio for improved returns. Results indicated that portfolios coupled with pre-selected assets generated better results than the portfolios with the entire NIFTY 50 stocks. Preselection helps filter out underperforming or overly volatile assets, leading to a more robust and resilient portfolio that aligns with specific investment objectives. This is consistent with the studies of [63],[64], and [65].
- It is evident from the study that a portfolio consisting of 5 stocks provides the optimal balance between diversification, risk management, and return maximization, which is consistent with the results of [62] but contradicting [3], [37] and [40]. While diversification is crucial to reduce unsystematic risk, excessive diversification beyond nine stocks leads to diminishing returns and unnecessary complexity. For instance, the top 10 and all 50 portfolios have significantly lower returns and Sharpe ratios compared to the top 7, 8 and 9 portfolios.

V. Conclusion

With advancements in machine learning and deep learning, though predicting asset returns has become feasible, these prediction results are not yet effectively utilized in practice for portfolio creation and optimization. As a result, many portfolios fail to fully capitalize on the available predictive insights, limiting their potential for improved performance and risk management. The challenge lies in effectively utilizing these predicted returns to construct an optimal investment portfolio. Considering this, this research seeks to tackle the issue by exploring how to integrate advanced forecasting information into the portfolio selection process.

The study has dual stages. In the first stage, the study developed a stacked LSTM capable of forecasting close values of all NIFTY 50 stocks by following a sliding window approach of 30 days. The model obtained an average accuracy of 90%. The second stage is asset pre-selection, where the top ten stocks, based on their predicted returns, were filtered for portfolio creation. Five portfolios each per objectives were created resulting in a total of 30 different portfolios. The results concluded that portfolios constituting five stocks result in best returns as high as 44%. Investors should avoid expanding their portfolios beyond nine stocks, as excessive diversification can lead to diminishing returns and unnecessary complexity. The proposed portfolios beat the benchmark NIFTY index as well as portfolios with no asset pre-selection, comprising all 50 stocks.

The findings of this study indicate that the proposed two-stage portfolio optimization method has the potential to construct a promising investment strategy due to its trade-off between historical and future information on assets. The results demonstrate the reliability and effectiveness of the asset selection approach in identifying high-performing assets, providing competitive risk-adjusted returns for portfolio optimization, beneficial for both portfolio managers and individual investors. Using real-time market predictions, the algorithm enables investors to choose assets with higher returns and apply the model, which accounts for recent data dynamics in expected return and risk. This makes the approach more practical. Consequently, the proposed method offers a systematic decision-making framework that assists in determining which assets to hold and their investment proportions to achieve the maximum risk-adjusted return and optimal risk-return balance. The study hence concludes that combining forecasting theory with portfolio selection could improve portfolio returns.

VI. Limitations and Future Scope

The assumptions used to test the portfolios do not accurately reflect their performance in real-world conditions. Real-world investments incur costs such as taxes, transaction fees, indivisibility of assets, and unexpected transaction prices. However, these assumptions do not impact the relative performance when compared to benchmarked portfolios. The portfolio construction in this study considered only stocks. Future studies could include assets from different classes to evaluate their performance.

Despite the improved performance, the proposed model used only historical values and STIs as input values for LSTM forecasting. Future studies could explore the integration of other sources of data, such as news articles and social media sentiment analysis, to improve the model's predictive power. Including exogenous factors, such as interest rates, inflation rates, and exchange rates, could also provide more comprehensive forecasting results. Moreover, the proposed model was validated only on the NIFTY stocks and on a single time-frame, limiting the generalizability of the results to other stock markets and time frames. Future work could explore the model's performance across different markets and under varying market conditions, such as highly volatile markets or those experiencing sudden shocks. This would help assess the model's robustness and applicability in diverse financial environments, providing insights into its potential for broader use in real-world scenarios.

References

[1] H. M. Markowitz, “Portfolio selection,” Journal of finance, vol. 7, no. 1, pp. 71–91, 1952.

[2] Z. Zhang, S. Zohren, and S. Roberts, “Deep Learning for Portfolio Optimization,” Journal of Financial Data Science, vol. 2, no. 4, pp. 8–20, 2020, doi: 10.3905/jfds.2020.1.042.

[3] 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, p. 113042, 2020, doi: 10.1016/j.eswa.2019.113042.

[4] Z. Zhou, Z. Song, T. Ren, and L. Yu, “Two-Stage Portfolio Optimization Integrating Optimal Sharp Ratio Measure and Ensemble Learning,” IEEE Access, vol. 11, no. December 2022, pp. 1654–1670, 2023, doi: 10.1109/ACCESS.2022.3232281.

[5] Y. H. Chou, S. Y. Kuo, and Y. T. Lo, “Portfolio Optimization Based on Funds Standardization and Genetic Algorithm,” IEEE Access, vol. 5, pp. 21885–21900, 2017, doi: 10.1109/ACCESS.2017.2756842.

[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 Forecast, vol. 31, no. 2, pp. 550–560, 2015, doi: https://doi.org/10.1016/j.ijforecast.2014.10.003.

[7] Y. Zhang, X. Li, and S. Guo, “Portfolio selection problems with Markowitz’s mean–variance framework: a review of literature,” Fuzzy Optimization and Decision Making, vol. 17, no. 2, pp. 125–158, 2018, doi: 10.1007/s10700-017-9266-z.

[8] M. Johnson, “Forecasting in Emerging Markets: Challenges and Solutions,” Journal of Emerging Market Finance, vol. 18, no. 1, pp. 1–14, 2019.

[9] L. T. Quang, “Application of Artificial Intelligence-Genetic Algorithms to Select Stock Portfolios in the Asian Markets,” International Journal of Advanced Computer Science and Applications, vol. 13, no. 12, pp. 469–476, 2022, doi: 10.14569/IJACSA.2022.0131257.

[10] J. Sen, A. Dutta, and S. Mehtab, “Stock Portfolio Optimization Using a Deep Learning LSTM Model,” 2021 IEEE Mysore Sub Section International Conference, MysuruCon 2021, pp. 263–271, 2021, doi: 10.1109/MysuruCon52639.2021.9641662.

[11] Z. Wang, K. Li, S. Q. Xia, and H. Liu, “Economic Recession Prediction Using Deep Neural Network,” Journal of Financial Data Science, vol. 4, no. 3, pp. 108–127, 2022, doi: 10.3905/jfds.2022.1.097.

[12] J. B. Heaton, N. G. Polson, and J. H. Witte, “Deep learning for finance: deep portfolios,” Appl Stoch Models Bus Ind, vol. 33, no. 1, pp. 3–12, 2017, doi: 10.1002/asmb.2209.

[13] A. M. Rather, “LSTM-based Deep Learning Model for Stock Prediction and Predictive Optimization Model,” EURO Journal on Decision Processes, vol. 9, no. May, p. 100001, 2021, doi: 10.1016/j.ejdp.2021.100001.

[14] F. Ploessl, T. Just, and L. Wehrheim, “Cyclicity of real estate-related trends: topic modelling and sentiment analysis on German real estate news,” Journal of European Real Estate Research, vol. 14, no. 3, pp. 381–400, 2021, doi: 10.1108/JERER-12-2020-0059.

[15] A. Chamekh, M. Mahfoudh, and G. Forestier, “Sentiment Analysis Based on Deep Learning : A Comparative Study,” Lecture Notes in Computer

Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 13369 LNAI, pp. 498–507, 2022, doi: 10.1007/978-3-031-10986-7_40.

[16] Q. Liu, Z. Tao, Y. Tse, and C. Wang, "Stock market prediction with deep learning: The case of China," Financ Res Lett, vol. 46, no. June, p. 102209, 2022, doi: 10.1016/j.frl.2021.102209.

[17] M. Dixon, D. Klabjan, and J. H. Bang, "Classification-based financial markets prediction using deep neural networks," Algorithmic Finance, vol. 6, no. 3–4, pp. 67–77, 2017, doi: 10.3233/AF-170176.

[18] H. Gunduz, Y. Yaslan, and Z. Cataltepe, "Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations," Knowl Based Syst, vol. 137, pp. 138–148, 2017, doi: 10.1016/j.knosys.2017.09.023.

[19] G. Taroon, A. Tomar, C. Manjunath, M. Balamurugan, B. Ghosh, and A. V. N. Krishna, "Employing Deep Learning in Intraday Stock Trading," Proceedings - 2020 5th International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN 2020, pp. 209–214, 2020, doi: 10.1109/ICRCICN50933.2020.9296174.

[20] B. Moews and G. Ibikunle, "Predictive intraday correlations in stable and volatile market environments: Evidence from deep learning," Physica A: Statistical Mechanics and its Applications, vol. 547, no. xxxx, p. 124392, 2020, doi: 10.1016/j.physa.2020.124392.

[21] P. Mehta, S. Pandya, and K. Kotecha, "Harvesting social media sentiment analysis to enhance stock market prediction using deep learning," PeerJ Comput Sci, vol. 7, pp. 1–21, 2021, doi: 10.7717/peerj-cs.476.

[22] A. Shah, M. Gor, M. Sagar, and M. Shah, "A stock market trading framework based on deep learning architectures," Multimed Tools Appl, vol. 81, no. 10, pp. 14153–14171, 2022, doi: 10.1007/s11042-022-12328-x.

[23] P. Ghosh, A. Neufeld, and J. K. Sahoo, "Forecasting directional movements of stock prices for intraday trading using LSTM and random forests," Financ Res Lett, vol. 46, no. June, p. 102280, 2022, doi: 10.1016/j.frl.2021.102280.

[24] M. S. Sivri and A. Ustundag, "An adaptive and enhanced framework for daily stock market prediction using feature selection and ensemble learning algorithms," Journal of Business Analytics, vol. 00, no. 00, pp. 1–21, 2023, doi: 10.1080/2573234X.2023.2263522.

[25] A. W. Li and G. S. Bastos, "Stock market forecasting using deep learning and technical analysis: A systematic review," IEEE Access, vol. 8, pp. 185232–185242, 2020, doi: 10.1109/ACCESS.2020.3030226.

[26] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and S. Shahab, "Deep learning for stock market prediction," Entropy, vol. 22, no. 8, 2020, doi: 10.3390/E22080840.

[27] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," Comput Ind Eng, vol. 143, no. March, p. 106435, 2020, doi: 10.1016/j.cie.2020.106435.

[28] I. Valova, N. Gueorguieva, T. Aayushi, P. Nikitha, and H. Mohamed, "Hybrid Deep Learning Architectures for Stock Market Prediction," Proceedings of the World Congress on Electrical Engineering and Computer Systems and Science, pp. 1–8, 2023, doi: 10.11159/cist23.121.

[29] P. N. Kolm, R. Tütüncü, and F. J. Fabozzi, "60 Years of portfolio optimization: Practical challenges and current trends," Eur J Oper Res, vol. 234, no. 2, pp. 356–371, Apr. 2014, doi: 10.1016/J.EJOR.2013.10.060.

[30] F. D. Freitas, A. F. De Souza, and A. R. de Almeida, "Prediction-based portfolio optimization model using neural networks," Neurocomputing, vol. 72, no. 10–12, pp. 2155–2170, 2009, doi: 10.1016/j.neucom.2008.08.019.

[31] C. Hao, J. Wang, W. Xu, and Y. Xiao, "Prediction-Based Portfolio Selection Model Using Support Vector Machines," in 2013 Sixth International Conference on Business Intelligence and Financial Engineering, 2013, pp. 567–571. doi: 10.1109/BIFE.2013.118.

[32] C. F. Huang, "A hybrid stock selection model using genetic algorithms and support vector regression," Appl Soft Comput, vol. 12, no. 2, pp. 807–818, Feb. 2012, doi: 10.1016/J.ASOC.2011.10.009.

[33] S. Il Lee and S. J. Yoo, "Threshold-based portfolio: the role of the threshold and its applications," Journal of Supercomputing, vol. 76, no. 10, pp. 8040–8057, 2020, doi: 10.1007/s11227-018-2577-1.

[34] Y. Ma, R. Han, and W. Wang, "Portfolio optimization with return prediction using deep learning and machine learning," Expert Syst Appl, vol. 165, no. September 2020, p. 113973, 2021, doi: 10.1016/j.eswa.2020.113973.

[35] G. Li, "Information sharing and stock market participation: Evidence from extended families," Review of Economics and Statistics, vol. 96, no. 1, pp. 151–160, 2014.

[36] Y. Zhang, X. Li, and S. Guo, "Portfolio selection problems with Markowitz's mean–variance framework: a review of literature," Fuzzy Optimization and Decision Making, vol. 17, no. 2, pp. 125–158, 2018, doi: 10.1007/s10700-017-9266-z.

[37] 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, 2019, doi: 10.1016/j.eswa.2018.08.003.

[38] Y. Ma, R. Han, and W. Wang, "Prediction-Based Portfolio Optimization Models Using Deep Neural Networks," IEEE Access, vol. 8, pp. 115393–115405, 2020, doi: 10.1109/ACCESS.2020.3003819.

[39] V. D. Ta, C. M. Liu, and D. A. Tadesse, "Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading," Applied Sciences (Switzerland), vol. 10, no. 2, 2020, doi: 10.3390/app10020437.

[40] W. Chen, H. Zhang, M. K. Mehlawat, and L. Jia, "Mean–variance portfolio optimization using machine learning-based stock price prediction," Appl Soft Comput, vol. 100, p. 106943, 2021, doi: 10.1016/j.asoc.2020.106943.

[41] G. Kumar, S. Jain, and U. P. Singh, "Stock market forecasting using computational intelligence: A survey," Archives of computational methods in engineering, vol. 28, no. 3, pp. 1069–1101, 2021.

[42] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," Neural Comput Appl, vol. 32, no. 13, pp. 9713–9729, 2020, doi: 10.1007/s00521-019-04504-2.

[43] K. Chen, Y. Zhou, and F. Dai, "A LSTM-based method for stock returns prediction : A case study of China stock market," in IEEE International Conference on Big Data (Big Data), 2015, pp. 2823–2824. [Online]. Available: http://arxiv.org/abs/1506.00019

[44] X. Zhong and D. Enke, "Predicting the daily return direction of the stock market using hybrid machine learning algorithms," Financial Innovation, vol. 5, no. 1, 2019, doi: 10.1186/s40854-019-0138-0.

[45] L. Cao and F. E. H. Tay, "Financial forecasting using support vector machines," Neural Computing and Application, no. 10, pp. 184–192, 2001, doi: 10.1016/S0925-2312(03)00372-2.

[46] A. Sebastian and V. Tantia, "Transforming Finance With Deep Learning Predictions," in Navigating the Future of Finance in the Age of AI, 2024, ch. 12, pp. 227–252. doi: 10.4018/979-8-3693-4382-1.ch012.

[47] A. Sebastian and V. Tantia, "From data to decisions: Harnessing AI and big data for advanced business analytics," in Social Reflections of Human-Computer Interaction in Education, Management, and Economics, 2024, ch. 6, pp. 97–124. doi: 10.4018/979-8-3693-3033-3.ch006.

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

[49] J. Shen and M. O. Shafiq, "Short-term stock market price trend prediction using a comprehensive deep learning system," J Big Data, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00333-6.

[50] S. Chen and L. Ge, "Exploring the attention mechanism in LSTM-based Hong Kong stock price movement prediction," Quant Finance, vol. 19, no. 9, pp. 1507–1515, 2019, doi: 10.1080/14697688.2019.1622287.

[51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, vol. 15, pp. 1929–1958, 2014.

[52] S. Almahdi and S. Y. Yang, "An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown," Expert Syst Appl, vol. 87, pp. 267–279, 2017, doi: 10.1016/j.eswa.2017.06.023.

[53] V. Mohan, J. G. Singh, and W. Ongsakul, "Sortino Ratio Based Portfolio Optimization Considering EVs and Renewable Energy in Microgrid

Power Market," IEEE Trans Sustain Energy, vol. 8, no. 1, pp. 219–229, 2017, doi: 10.1109/TSTE.2016.2593713.

[54] Z. Zhou, Z. Song, T. Ren, and L. Yu, "Two-Stage Portfolio Optimization Integrating Optimal Sharp Ratio Measure and Ensemble Learning," IEEE Access, vol. 11, no. December 2022, pp. 1654–1670, 2023, doi: 10.1109/ACCESS.2022.3232281.

[55] F. A. Sortino, The Sortino Framework for Constructing Portfolios: Focusing on Desired Target ReturnTM to Optimize Upside Potential Relative to Downside Risk. Elsevier, 2009.

[56] O. B. Ansari and F. M. Binninger, "A deep learning approach for estimation of price determinants," International Journal of Information Management Data Insights, vol. 2, no. 2, p. 100101, 2022, doi: 10.1016/j.jjimei.2022.100101.

[57] I. H. Shakri, "Time series prediction using machine learning: a case of Bitcoin returns," Studies in Economics and Finance, vol. 39, no. 3, pp. 458–470, 2022, doi: 10.1108/SEF-06-2021-0217.

[58] G. Ding and L. Qin, "Study on the prediction of stock price based on the associated network model of LSTM," International Journal of Machine Learning and Cybernetics, vol. 11, no. 6, pp. 1307–1317, 2020, doi: 10.1007/s13042-019-01041-1.

[59] P. Singh and M. Jha, "Portfolio Optimization Using Novel EW-MV Method in Conjunction with Asset Preselection," Comput Econ, 2024, doi: 10.1007/s10614-024-10583-8.

[60] P. Singh and M. Jha, "Portfolio Optimization Using Novel EW-MV Method in Conjunction with Asset Preselection," Comput Econ, no. 0123456789, 2024, doi: 10.1007/s10614-024-10583-8.

[61] A. Chaweewanchon and R. Chaysiri, "Markowitz Mean-Variance Portfolio Optimization with Predictive Stock Selection Using Machine Learning," International Journal of Financial Studies, vol. 10, no. 3, 2022, doi: 10.3390/ijfs10030064.

[62] W. Chen, H. Zhang, M. K. Mehlawat, and L. Jia, "Mean–variance portfolio optimization using machine learning-based stock price prediction," Appl Soft Comput, vol. 100, p. 106943, 2021, doi: 10.1016/j.asoc.2020.106943.