# Efficient Feature-Engineering-Based Optimized Neural Network Architecture for Stock Price Prediction

**Shambhu Dayal Sahu[1] and Dr. Puneet Dwivedi[2]**
[1] PhD Scholar, Ram Krishna Dharmarth Foundation University, Gandhi Nagar, Bhopal, India
[2] Professor, Ram Krishna Dharmarth Foundation University, Gandhi Nagar, Bhopal, India
[1]shambhu2009@gmail.com, [2]puneet86.dwivedi@gmail.com

**Abstract**

Predicting stock market trends is inherently challenging due to the volatile and nonlinear nature of financial data. To address these complexities, optimization-driven machine learning approaches have gained increasing attention. This study presents a comparative evaluation of machine learning models for predicting the opening and closing prices of IT company stocks using historical market data. The dataset, comprising features such as high price, low price, and trading volume, was pre-processed through linear interpolation to handle missing values and normalized using z-score standardization. Feature engineering was further applied by incorporating a 3-day moving average, while Principal Component Analysis (PCA) reduced dimensionality while preserving 95% of the data variance. The models evaluated include Artificial Neural Networks (ANN), Linear Regression (LR), and Support Vector Regression (SVR). A grid search was employed to optimize the number of hidden neurons in the ANN, resulting in an optimal configuration for Microsoft with $R^2 = 0.9876$, MSE = 0.0124, and MAE = 0.0879. The final ANN model achieved superior performance on the test set, with an MSE of 0.0098, MAE of 0.0781, and $R^2$ of 0.9902. Comparative analysis indicated that the ANN consistently outperformed both LR and SVR across all evaluation metrics for predicting opening and closing prices.

**Key Words:** Stock Market Prediction, Machine Learning, Optimization, Neural Network, PCA, SVR, LR, MSE.

## 1. INTRODUCTION

Forecasting stock market variability has been a significant area of interest in financial research because of its important implications for investment forecasting, risk mitigation, and broader financial decision-making. The stock market exhibits inherent complexity, driven by numerous dynamic and nonlinear factors such as macroeconomic conditions, political events, investor behavior, and global financial influences. However, this complexity poses considerable challenges for accurate prediction, as conventional statistical approaches often fail to capture hidden patterns and abrupt market fluctuations. Developing reliable predictive models is therefore critically important, as they can assist investors and policymakers in making well-informed decisions, minimize financial uncertainty, and enhance overall market stability. Stock market prediction (SMP) remains a highly challenging task, and accurate predictive methodologies are essential for reducing the complexity of financial markets. Recent studies have highlighted that stock prices are influenced by various factors, including economic indicators, market sentiment, and global events, which often lead to sudden fluctuations and unpredictable trends [1–5]. Traditional statistical methods frequently struggle to capture these dynamic patterns, limiting their predictive accuracy. To address these challenges, recent research has increasingly focused on machine learning (ML) and deep learning techniques—particularly Artificial Neural Networks (ANNs) and ensemble architectures—which offer the ability to model complex, nonlinear relationships and adapt to evolving market conditions

[1,3,4]. Comprehensive literature reviews emphasize the growing importance of high-quality data, optimized model architectures, and hybrid learning approaches to improve forecasting reliability and reduce prediction errors [2,5]. Collectively, these studies underscore both the critical need and the ongoing challenges in developing robust and efficient SMP systems capable of supporting informed investment decisions.

This study proposes the design of an accurate neural network (NN)-based SMP approach by integrating ML-based optimization techniques and feature engineering (FE). The use of NNs offers several advantages for stock market prediction due to their inherent ability to model complex, nonlinear relationships that traditional statistical methods often fail to capture. NNs can learn from large volumes of historical data, identifying hidden patterns and dependencies among market variables such as opening, closing, high, and low prices, as well as trading volume [1,4]. The list of abbreviations used for SMP in this study is provided in Table 1.

Table 1: List of Abbreviations Used in the SMP Study

| | Abbreviations | | Abbreviations |
|---|---|---|---|
| SMP | Stock Market Prediction | SVR | Support Vector Regression |
| ANN | Artificial Neural Network | FE | Feature Engineering |
| NN | Neural Network | RMSE | Root Mean Square Error |
| ML | Machine Learning | PCA | Principal Component Analysis |
| DL | Deep Learning | MAE | Mean Absolute Error |
| LR | Linear Regression | NaN | Not a Number |

Secondly, they provide flexibility in architecture, allowing the use of single-layer, multi-layer, or ensemble networks to optimize prediction accuracy under varying market conditions [3]. Thirdly, NNs are particularly effective in handling dynamic and volatile financial environments, as they can adapt to sudden market fluctuations and capture nonlinear interactions between factors [2,5]. Additionally, NNs support feature engineering and the integration of technical indicators (e.g., moving averages, RSI) to enhance predictive performance, making them highly suitable for both short-term and long-term forecasting [4,5]. Overall, these capabilities enable neural networks to deliver robust and reliable forecasts, thereby supporting informed investment decisions and improving risk management in stock trading.

**Figure 1: Challenges in SMP and Corresponding ML-Based Solutions**

Figure 1 presents a structured overview of the challenges and machine learning-based solutions for stock market prediction. To address this complexity, optimization-based approaches in ML have gained increasing attention. In this study, an NN model is systematically optimized by exploring different hidden layer configurations to identify the best-performing architecture. Mean Squared Error (MSE) is employed as the fitness function, enabling accurate evaluation of prediction performance across models. The adoption of ML solutions for SMP is strongly justified by their ability to capture complex and nonlinear relationships that traditional statistical approaches often fail to represent effectively. Stock price dynamics are shaped by a diverse interplay of economic, social, and political factors, resulting in intricate patterns and abrupt fluctuations within financial data.

ML-based prediction methods are well suited to address this complexity, as they can process large historical datasets, adapt to changing market conditions, and reveal hidden interdependencies among variables. Approaches such as Artificial Neural Networks (ANN), Support Vector Regression (SVR), and ensemble learning provide the flexibility to identify patterns that remain obscure to conventional techniques. By leveraging these capabilities, ML models can improve forecasting accuracy, facilitate informed decision-making, and enhance risk management and investment strategies in highly volatile market environments.

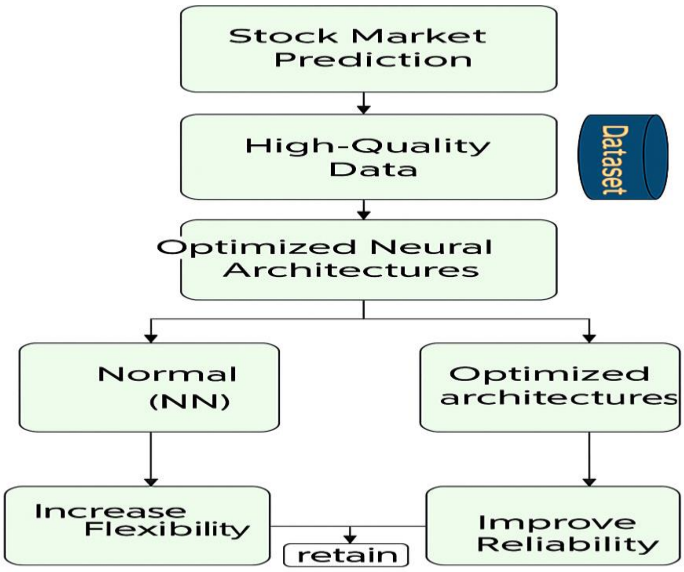



**Figure 2: Architecture diagram for SMP using ML methodologies**

Figure 2 illustrates the architectural framework of a data-driven stock market prediction system that integrates high-quality financial datasets with advanced model optimization techniques. As shown in Figure 2, the proposed framework begins with the collection of structured financial data, including historical stock prices and relevant market indicators. Data from multiple Indian IT companies are considered for predicting the opening and closing prices of stocks. The collected data then undergo preprocessing to ensure accuracy, consistency, and relevance. Advanced feature engineering (FE) techniques are applied to improve data quality. This processed data serves as input to the core prediction module, where carefully optimized model architectures are employed to capture the intricate and nonlinear dynamics of financial markets.

A grid search strategy is adopted to ensure optimal parameter selection, thereby enhancing model robustness. The effectiveness of the proposed NN framework is further validated through evaluation on test data, with results visualized for both opening and closing price predictions. Finally, a comparative analysis is conducted among Support Vector Regression (SVR), Linear Regression (LR), and the optimized NN solution, highlighting the advantages and limitations of each method in stock market forecasting.
The architecture features two parallel learning pathways. The first pathway utilizes Neural Networks (NNs), offering flexible configurations ranging from single-layer to multi-layer feedforward structures. The second pathway employs ensemble architectures, which enhance forecasting robustness by integrating the outputs of multiple predictive models. Within the ensemble pathway, an SMP Data module is incorporated, functioning as a post-processing stage to refine predictive signals and improve interpretability for decision-making.

## 2. LITERATURE REVIEW

Stock Market Prediction (SMP) remains one of the most challenging tasks in financial analytics due to the highly dynamic, nonlinear, and stochastic nature of market behavior. A wide range of models, spanning from traditional econometric methods to advanced machine learning (ML), deep learning (DL), and hybrid optimization-based frameworks, have been applied to forecast market indices and stock prices. The reviewed works collectively reveal a growing trend toward integrating data-driven intelligence with optimization techniques to improve both accuracy and robustness in prediction.
Lamba et al. [1] conducted a comparative analysis of Artificial Neural Networks (ANNs) for predicting the Nifty 50 index in the Indian stock market. Their findings established that ANN-based models effectively captured the nonlinear characteristics of financial data and provided more accurate forecasts than traditional regression approaches. Building upon this, Kumbure et al. [2] conducted an extensive literature review on ML techniques for stock market forecasting. They emphasized the growing importance of data diversity, highlighting that the integration of technical, fundamental, and sentiment indicators significantly enhances predictive accuracy. Singh et al. [3] introduced the Efficient Stock Market Prediction System (ESMPS), which integrates optimized and ensemble deep learning architectures. Their results demonstrated that ensembles outperform individual deep learning models in terms of both accuracy and generalization capability. Similarly, Sreya et al. [4] focused on the Indian stock market and compared different neural network configurations, confirming that deep neural approaches offer substantial improvements in handling volatility and uncertainty in financial data. Selvamuthu et al. [5] applied ANN models to high-frequency tick data from the Indian stock market. Their study highlighted that neural networks are highly effective in modeling short-term fluctuations and micro-level variations often missed by conventional models. In contrast, Chopra and Sharma [6] presented a critical review of Artificial Intelligence (AI) applications in stock market forecasting. They acknowledged the advances achieved using ML and DL while emphasizing the need to balance predictive accuracy with interpretability and risk awareness. Kurani et al. [7] conducted a comparative study between ANNs and Support Vector Machines (SVMs). Their findings suggested that while ANNs capture nonlinear relationships more effectively, SVMs demonstrate greater stability under noisy data conditions, making them suitable for specific financial contexts. Supporting the ANN perspective, Ramireddy and Prasad [8] showcased the effectiveness of ANN-based models in stock market return forecasting, further strengthening the case for neural models in predictive financial tasks. Dash et al. [9] proposed a fine-tuned Support Vector Regression (SVR) model for stock prediction, demonstrating its superiority in handling highly volatile financial time series. Earlier, Yang et al. [10] explored SVR in the context of volatile stock market prediction,

establishing its robustness in capturing complex relationships. Nwaigwe et al. [11] modeled Microsoft stock prices using a combination of machine learning and classical models. Their study revealed that while classical approaches such as ARIMA remain competitive, machine learning methods consistently deliver better performance for highly nonlinear market data. Arauco Ballesteros and Martínez Miranda [12] explored stock market forecasting by combining fundamental indicators, technical indicators, and sentiment analysis within an ANN framework. Their results highlighted the importance of multi-source feature integration for improving forecasting performance. In another deep learning-focused study, Dronyuk et al. [13] applied Deep Neural Networks (DNNs) to stock value prediction, showing that DNNs provide improved predictive performance over shallow architectures. Kamalov [14] examined neural networks for forecasting significant stock price changes, revealing that neural models can detect abrupt shifts in stock prices—an area where linear models typically underperform. Roy et al. [15] investigated stock forecasting using the LASSO linear regression model. While their work confirmed the utility of LASSO in reducing overfitting through feature selection, they also noted its limitations compared to nonlinear methods. Sable et al. [16] introduced genetic algorithms and evolutionary strategies for stock price prediction. Their results emphasized the benefits of optimization-driven approaches in improving convergence and avoiding local minima in predictive modeling. Singh et al. [17] further explored ML applications in stock prediction, consolidating findings that highlight ML's superiority over traditional statistical models. Srividya et al. [18] extended this perspective by implementing a Recurrent Neural Network (RNN) with an LSTM architecture, demonstrating its ability to capture sequential dependencies and outperform static models in time-series forecasting. Huang [19] employed the Prophet model with multiple macroeconomic regressors for stock price forecasting, showcasing the growing importance of macro-driven models in complementing ML-based predictions. Finally, Shahvaroughi Farahani and Razavi Hajiagha [20] combined ANNs with metaheuristic algorithms and compared them with classical time-series approaches. Their hybrid framework proved superior in accuracy and robustness, reinforcing the idea that integrated optimization and learning methods represent a promising direction for stock market forecasting. Fu et al. [21] investigated the forecasting of IBM stock prices using a diverse set of deep learning architectures, including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Attention-based models, and Transformers. Their comparative study revealed that while LSTM and GRU models demonstrated strong performance in capturing temporal dependencies, attention mechanisms and transformers significantly improved long-range dependency modeling and interpretability. The findings emphasized that advanced sequence models can outperform traditional recurrent frameworks, particularly when dealing with highly volatile financial data. Dželihodžić et al. [22] conducted a comparative analysis of multiple deep learning architectures for stock price prediction, including LSTM, GRU, Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN). Their results highlighted that no single model universally dominates across all market conditions; rather, the choice of model depends heavily on data characteristics and the nature of the prediction task. This work underscores the importance of evaluating multiple architectures before selecting an optimal model for deployment. Nwaigwe et al. [23] explored Microsoft stock price forecasting by comparing classical statistical approaches, such as ARIMA, with modern machine learning techniques. Their study concluded that while classical models remain relevant for short-term and trend-based forecasting, ML models consistently achieve higher predictive accuracy in nonlinear and volatile environments. This dual comparison is particularly valuable for practitioners seeking to balance interpretability with predictive performance in financial modeling. Anh and Son [24] proposed a transformative framework for stock price forecasting by integrating deep learning architectures with advanced feature engineering strategies. Their work demonstrated that incorporating domain-specific engineered features alongside raw historical data substantially improves prediction accuracy. The study emphasized the synergistic role of model architecture and input representation, showing that feature engineering remains a critical component of successful deep learning applications in finance. Gülmez [25] introduced a hybrid stock market forecasting model that combines Long Short-Term Memory (LSTM) networks with the Seahorse Optimization Algorithm (SOA). The hybridization allowed for both robust temporal sequence learning and parameter optimization, resulting in improved forecasting accuracy compared to standalone LSTM models. This work illustrates the growing trend of merging deep learning with metaheuristic optimization to achieve higher robustness and generalization capability in financial forecasting. John and Nidhina [26] extended machine learning applications beyond stock markets by focusing on gold price forecasting. Their study tested multiple ML models to evaluate predictive accuracy and robustness in commodity markets, demonstrating that advanced ML approaches can effectively handle both short-term volatility and long-term trend prediction. The findings broaden the applicability of ML techniques to different financial assets, reinforcing their adaptability across domains. Chathli et al. [27] developed a hybrid deep learning framework integrating LSTM and GRU architectures for stock prediction. By leveraging the strengths of both models—LSTM's ability to capture long-term dependencies and GRU's computational efficiency—the hybrid approach provided superior predictive accuracy compared to standalone models. Their study further reinforced the effectiveness of hybridization strategies in overcoming individual model limitations, making it a promising direction for future financial forecasting research.

A summary of the reviewed literature, including identified challenges, limitations, and performance parameters of existing research, is presented in Table 2.

**Table 2: Summary of SMP Literature review and limitations**

| Authors & Ref | Methodology / Models Used | Performance Parameters | Limitations |
|---|---|---|---|
| Lamba et al. [1] (2021) | ANN models for predicting Nifty 50 index | Accuracy, RMSE | Limited to Indian market; lacks hybrid comparison |
| Kumbure et al. [2] (2022) | Review of ML techniques (NN, SVM, DL) for stock forecasting | Comparative performance, feature importance | General review; no empirical experiments |
| Singh et al. [3] (2025) | ESMPS: Optimized ensemble deep learning | High accuracy, robustness | Increased complexity, high computation |
| Sreya et al. [4] (2026) | NN-based comparative analysis for Indian stock market | Prediction accuracy, error rates | Market-specific; limited generalization |
| Selvamuthu et al. [5] (2019) | ANN on tick-level Indian stock market data | High-frequency accuracy | Struggles with extreme volatility |
| Chopra & Sharma [6] (2021) | Review of AI in stock forecasting | Comparative insights, strengths/weaknesses | Conceptual; lacks implementation results |
| Kurani et al. [7] (2023) | ANN vs. SVM comparative study | Prediction error, classification accuracy | Tradeoff between complexity and interpretability |
| Reddy & Prasad [8] (2021) | ANN for stock market returns forecasting | Accuracy, trend capture | Small dataset; limited validation |
| Dash et al. [9] (2023) | Fine-tuned SVR model | High accuracy, better generalization | Computationally intensive tuning |
| Yang et al. [10] (2002) | Early SVM regression for volatile markets | Trend forecasting accuracy | Early models less scalable |
| Nwaigwe et al. [11] (2023) | ML vs. classical models for Microsoft stock | Accuracy, trend detection | Classical models fail with nonlinearities |

| | | | |
|---|---|---|---|
| Ballesteros & Miranda [12] (2025) | NN using fundamental, technical, sentiment indicators | High accuracy; robust predictions | Data preprocessing intensive |
| Dronyuk et al. [13] (2025) | Deep Neural Networks for stock prediction | Accuracy, non-linear pattern capture | Overfitting risk; large data demand |
| Kamalov [14] (2020) | NN for forecasting significant price changes | Robust detection of large movements | May miss subtle variations |
| Roy et al. [15] (2015) | LASSO Linear Regression model | Regularized accuracy, interpretability | Weak in nonlinear markets |
| Sable et al. [16] (2017) | Genetic algorithms & evolutionary strategies | Optimization efficiency, accuracy | Complex tuning, longer runtime |
| Singh et al. [17] (2021) | General ML methods for stock prediction | Accuracy, error metrics | Limited scope; basic models only |
| Srividya et al. [18] (2025) | LSTM-based RNNs | Sequential prediction accuracy | Sensitive to hyperparameters |
| Huang [19] (2022) | Prophet model with macroeconomic regressors | Forecasting accuracy, interpretability | Limited for short-term volatility |
| Farahani & Hajiagha [20] (2021) | ANN integrated with metaheuristics vs. time-series | High accuracy, robust optimization | High computation; complex hybridization |
| Fu et al. [21] (2023) | LSTM, GRU, Attention, Transformer for IBM stock forecasting | Accuracy, ability to capture temporal & long-range dependencies | Transformers require high computational resources; performance sensitive to hyperparameters |
| Dželihodžić et al. [22] (2024) | Comparative study of LSTM, GRU, CNN, RNN for stock price prediction | Prediction accuracy, robustness across datasets | No single model consistently best; model selection depends on data characteristics |
| Nwaigwe et al. [23] (2023) | Machine Learning vs. Classical Models (ARIMA, ML approaches) for Microsoft stock | Predictive accuracy, trend capturing ability | Classical models struggle with nonlinearity; ML models require large datasets and tuning |
| Anh & Son [24] (2024) | Deep Learning + Strategic Feature Engineering for stock forecasting | Improved accuracy via engineered features | Feature engineering is resource-intensive; may lack generalizability across markets |
| Gülmez [25] (2025) | Hybrid LSTM + Seahorse Optimization Algorithm (SOA) | Enhanced forecasting accuracy and robustness | Hybrid models may increase complexity; optimization adds extra computation |
| John & Nidhina [26] (2025) | ML techniques for Gold Price Forecasting | Accuracy in commodity markets, trend prediction | Limited to gold market; does not address high-frequency volatility |
| Chathli et al. [27] (2026) | Hybrid LSTM-GRU architecture for stock forecasting | Superior accuracy vs. standalone LSTM/GRU | Increased model complexity; risk of overfitting on small datasets |

## 3. CHALLENGES OF ML BASED STOCK PREDICTION

Stock Market Prediction (SMP) using machine learning (ML) solutions remains a complex and highly challenging task due to the inherent volatility, nonlinearity, and noise in financial data. A major challenge lies in data quality and preprocessing, as stock market data are often noisy, incomplete, and influenced by external factors such as geopolitical events, economic indicators, and investor sentiment. Numerous studies have shown that traditional ML algorithms such as Support Vector Machines (SVM), regression models, and decision trees often fail to capture deep temporal dependencies and nonlinear interactions, thereby limiting predictive accuracy. Another significant limitation is the tendency of models to overfit when trained on short-term or limited datasets, leading to poor generalization across unseen market conditions. Furthermore, many ML approaches rely heavily on technical indicators or price-related features while neglecting fundamental and sentiment-based factors, which weakens their robustness in dynamic environments. Hyperparameter tuning and model optimization also remain resource-intensive, as inappropriate configurations can result in unstable or inconsistent performance. Additionally, although ensemble and hybrid ML models have demonstrated performance improvements, they often involve increased complexity and computational cost, reducing their scalability for real-time trading applications. Thus, the limitations highlighted in the literature emphasize that despite significant progress, achieving robust, accurate, and generalizable SMP remains an unresolved challenge, as illustrated in Figure 3.

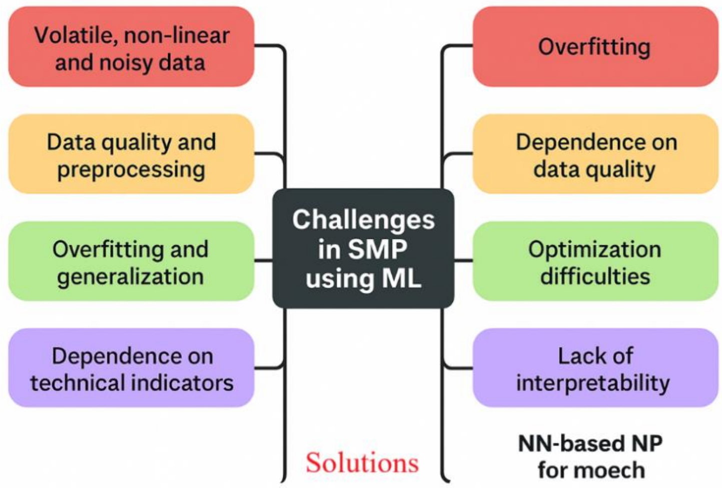



**Figure 3: Challenges of SMP based on ML solutions**

The application of machine learning (ML) for stock market prediction (SMP) faces persistent challenges due to the volatile, nonlinear, and noisy nature of financial data, as illustrated in Figure 3. While Neural Networks (NNs) have shown significant potential in modeling complex temporal dependencies, their effectiveness remains limited by overfitting, dependence on data quality, optimization difficulties, and lack of interpretability. Moreover, NN models are computationally expensive and often fail to generalize across dynamic and unseen market conditions. These limitations highlight the need for an optimized and robust NN-based framework that integrates advanced feature engineering, systematic optimization strategies, and interpretability mechanisms to enhance predictive performance and scalability in stock market forecasting.

## 4. PROPOSED NN METHODOLOGY FOR SMP

The primary contribution of the proposed methodology lies in its ability to enhance both the accuracy and robustness of stock market prediction (SMP) through several carefully designed improvements. First, meaningful patterns are extracted from financial data using advanced feature engineering techniques, which enrich the input space and enable the model to better capture market dynamics. Second, the integration of Bayesian regularization mitigates the risk of overfitting, ensuring that the neural network generalizes effectively to unseen data. Third, the adoption of suitable activation functions and optimized training settings accelerates convergence, resulting in more stable and efficient model training. Finally, hyperparameter tuning is employed to systematically identify the best-performing model configuration, maximizing predictive performance. Collectively, these modifications provide a substantial improvement over conventional approaches such as Linear Regression, positioning the proposed neural network framework as a more accurate and resilient solution for stock market forecasting.

This research proposes an efficient SMP methodology by integrating optimized features and an extended multi-layer neural network (NN) architecture, as illustrated in Figure 4. The proposed NN architecture comprises an input layer, two hidden layers ($H_1, H_2$), and an output layer. The input layer receives preprocessed and normalized features after PCA-based dimensionality reduction. The first hidden layer contains an optimized number of neurons determined through grid search, while the second hidden layer further enhances the network's ability to capture complex patterns. The output layer consists of a single neuron responsible for predicting either the opening or closing stock prices.

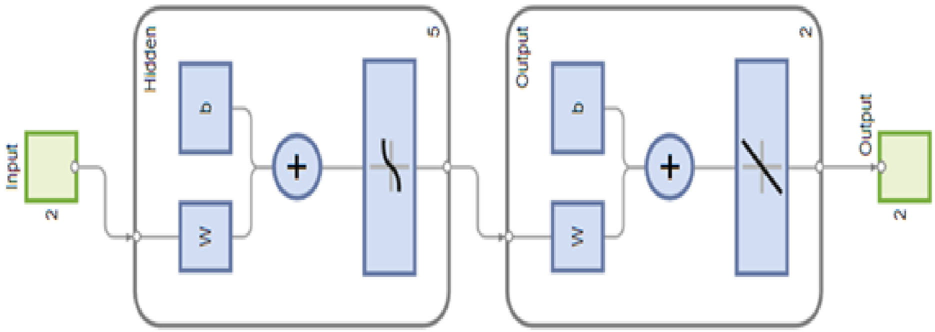



**Figure 4: NN layer architecture**

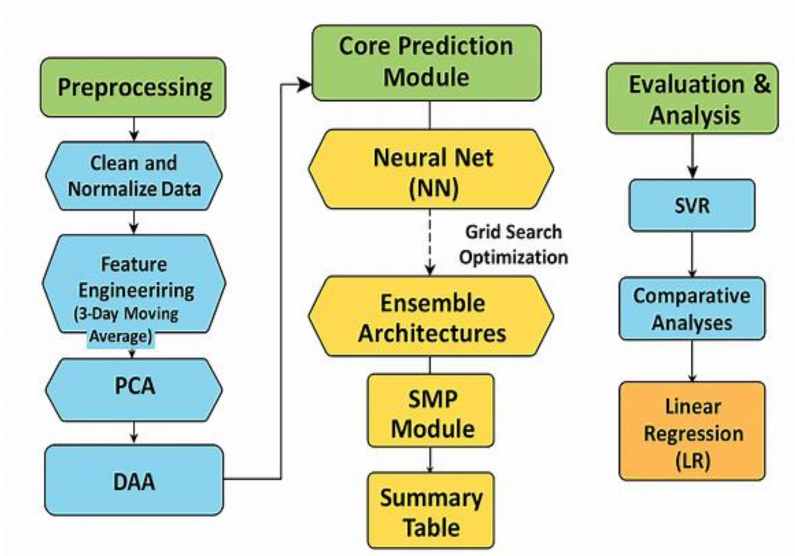



**Figure 5: Block diagram of Proposed Method**

The proposed system diagram is illustrated in Figure 5. The process begins with loading data from multiple Indian IT companies (one at a time), which are used to predict both the opening and closing prices of stocks. The raw data undergoes a rigorous preprocessing stage to ensure accuracy, consistency, and relevance, eliminating noise and handling missing or anomalous entries. Advanced feature engineering techniques are subsequently applied to enhance data quality and extract informative features, thereby improving the predictive power of the models. In the proposed work, data is pre-processed using linear interpolation to handle missing values and normalized through z-score standardization. Feature engineering is further applied by incorporating a 3-day moving average, while Principal Component Analysis (PCA) reduces dimensionality while preserving 95% of the data variance.

Following preprocessing, the refined data is fed into the core prediction module, as shown in Figure 5, which employs carefully optimized neural network architectures capable of capturing the complex and nonlinear dynamics inherent in financial markets. A grid search strategy is implemented to identify the optimal hyperparameters, ensuring robust model performance and minimizing overfitting. The effectiveness of the proposed neural network framework is rigorously evaluated using the test dataset, with results visualized to assess predictive accuracy for both opening and closing stock prices.

The framework in Figure 5 incorporates a dual-pathway architecture to enhance forecasting reliability. The first pathway leverages neural networks with flexible configurations, ranging from single-layer to multi-layer feedforward structures, to model intricate market patterns. The second pathway integrates ensemble learning architectures, which improve prediction stability by combining outputs from multiple predictive models. Within this ensemble pathway, a specialized SMP module is included, serving as a post-processing stage to refine model outputs, enhance signal interpretation, and facilitate more informed decision-making. Finally, a comparative analysis is performed between SVR and other models. The sequential flowchart of the proposed methodology is presented in Figure 6.

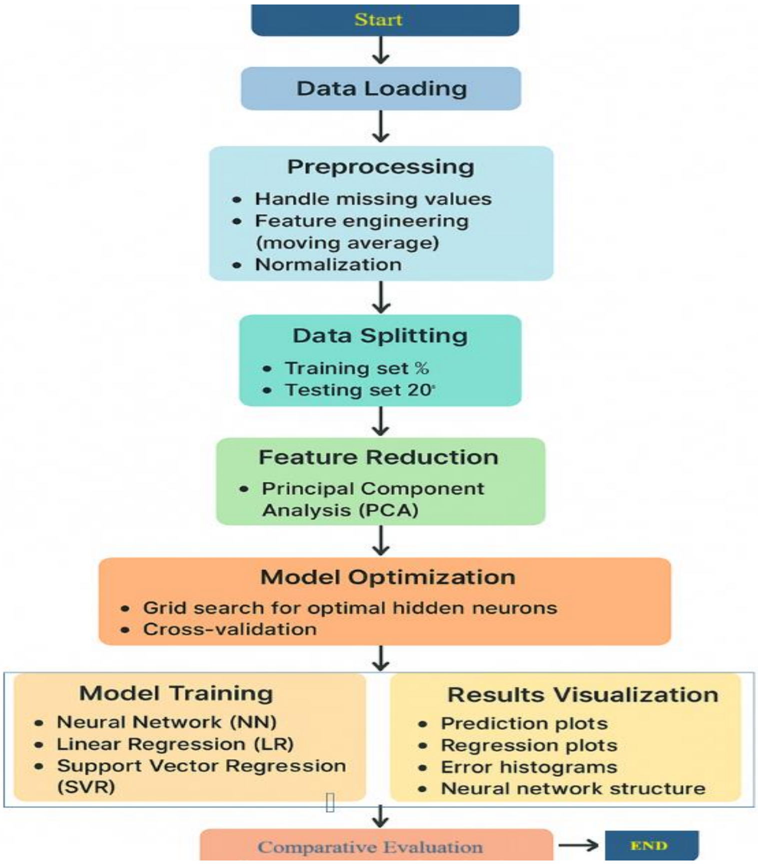



**Figure 6: Flow chart of the Proposed Method**

The proposed approach compares Linear Regression (LR) and the optimized neural network (NN) model, highlighting the relative strengths, limitations, and overall advantages of the proposed methodology for stock market forecasting. The implementation of each stage in Figure 6 is described sequentially in this section.

**Dataset Details-** The proposed work uses datasets from multiple Indian IT companies, including Microsoft, Facebook (FB), Gold, and IBM. The data is publicly available on GitHub and Kaggle databases [28], [29].

**Data Preparation and Preprocessing-** Data processing begins by loading stock data from IT companies (e.g., Microsoft). High, Low, and Volume are identified as input features, while Open and Close prices are the target variables to be predicted. Data features are pre-processed through linear interpolation to handle missing values. The interpolated value yis calculated using the formula for a straight line:

$$y = y_1 + \frac{(x - x_1)}{(x_2 - x_1)}(y_2 - y_1) \qquad (1)$$

The proposed work enhances the feature set by incorporating 3-day moving average features, implying that historical trends are important for prediction. The features are normalized using z-score standardization:

$$\mathrm{features}_{\mathrm{Enhc}} = \frac{\mathrm{features}_{\mathrm{Enhc}} - \mathrm{mean}(\ \mathrm{features}_{\mathrm{Enhc}})}{\mathrm{std}(\ \mathrm{features}_{\mathrm{Enhc}})} \qquad (2)$$

Subtracting the mean centers the data around zero, and dividing by the standard deviation scales it to have unit variance. This ensures that all variables contribute equally to model training and improves convergence. Similarly, the target variables (Open and Close prices) are normalized using the same z-score method:

$$\mathrm{targets} = \frac{\mathrm{targets} - \mathrm{mean}(\mathrm{targets})}{\mathrm{std}(\mathrm{targets})} \qquad (3)$$

**Feature Engineering-** Principal Component Analysis (PCA) is applied to reduce dimensionality while preserving 95% of the data variance, simplifying the model and reducing multicollinearity without significant information loss. PCA is applied to the normalized and enhanced training features. The principal component coefficients (or loadings) define new principal component axes representing the original features in the transformed space. The number of components is automatically determined by calculating the cumulative variance explained and selecting the minimum number required to preserve at least 95% of the total variance.

**Neural Network (NN) Model Optimization-** Selecting the optimal NN layer architecture is essential for accurate SMP. This is achieved using a multi-feature grid search algorithm.

**Grid Search-** A range of possible hidden layer sizes, from 5 to 50 in increments of 5, is defined. Each configuration is systematically tested to identify the optimal number of neurons based on the highest $R^2$value. The search space for the hidden layer hyperparameter his defined as:

$$H = \{5,10,15,\dots,50\} \qquad (4)$$

**Cross-Validation-** To ensure robustness and prevent overfitting, a 5-fold cross-validation scheme is used. For each $h \in H$, performance is evaluated across the folds to ensure the model generalizes well to unseen data.

**Training and Optimization-** Bayesian Regularization is employed for NN training, which is effective in preventing overfitting in smaller or noisy datasets. Different hidden layer sizes are explored to identify the best-performing ANN configuration. Mean Squared Error (MSE) serves as the fitness function for evaluating predictive performance. For each fold $k \in [1,2,3,4,5]$, the model is trained on four folds and validated on the remaining fold. This process is repeated across all folds.

**Model Evaluation-** The effectiveness of the trained model is evaluated using test data, with results visualized through plots for both opening and closing prices.

## 5. RESULTS AND DISCUSSION

This paper compares the performance of the optimized Neural Network against Linear Regression and Support Vector Regression models. The models are evaluated using three key metrics which is shown in table 3: Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared ($R^2$), which are defined as follows.

**Table 3: Evaluation Parameters**

| Parameter name | Formula | Equation No. |
|---|---|---|
| **Mean Squared Error (MSE)** | $MSE = \frac{1}{n} = \sum_{i=1}^{n} (yi - y^\wedge)^2$ | 5 |
| **Mean Absolute Error (MAE)** | $MSE = \frac{1}{n} = \sum_{i=1}^{n} \lvert (yi - y^\wedge) \rvert$ | 6 |
| **R-squared ($R^2$)** | $R^2 = 1 - \sum_{i=1}^{n} (yi - y^\wedge)^2 / \sum_{i=1}^{n} (yi - m)^2$ | 7 |
| **Average $R^2$ for a hidden size *h*** | $avgR^2(h) = \frac{1}{K} \sum_{k=1}^{K} R_k^2$ | 8 |

Where:

- n = number of observations
- yi = actual value
- y^ = predicted values
- m = mean of actual values

For the K fold cross validation model After iterating through all $K$=5 folds for a specific $h$ the average performance metrics are computed.

These parameters are used as performance evaluation parameters for accessing the performance of the predicted ML models for the SMP.

### 5.1 Predictive Models Comparison

This section compares the performance of ML-based SMP methods. Figure 7 illustrates the comparative performance of the proposed SMP framework for Microsoft data from 2015–2021 in forecasting both opening and closing stock prices. The plots in Figure 7 reveal that the predictions generated by the NN, LR, and SVR models closely follow the actual stock price trends across the sample period. For opening price prediction (top graph), the NN model demonstrates superior alignment with the actual values, particularly in regions of rapid fluctuation, whereas LR and SVR exhibit slightly larger deviations. Similarly, for closing price prediction (bottom graph), the NN approach consistently tracks the actual price dynamics more accurately, while SVR provides competitive performance but with minor discrepancies in highly volatile segments. Overall, the results confirm that the optimized NN framework offers improved predictive accuracy and robustness compared to traditional methods such as LR and SVR, validating the effectiveness of the proposed enhancements in the SMP methodology.

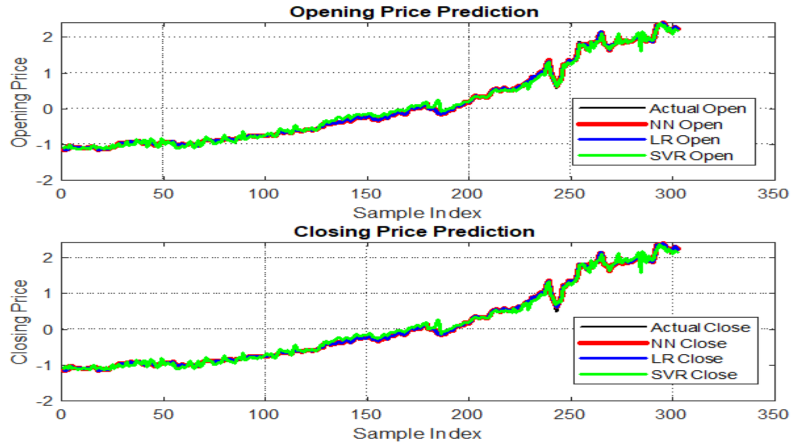
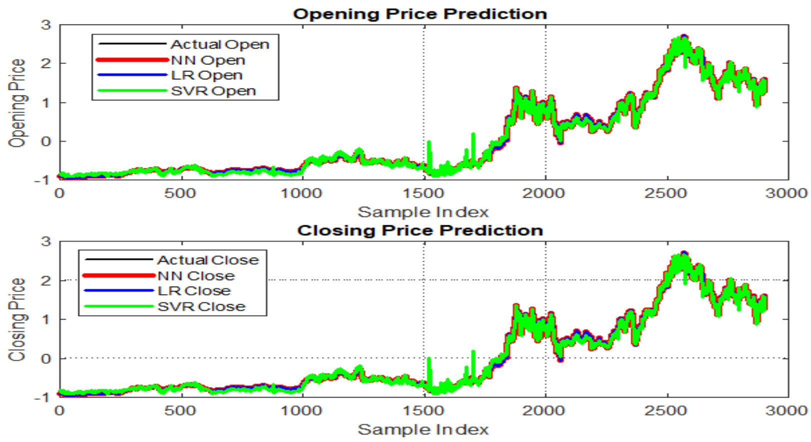
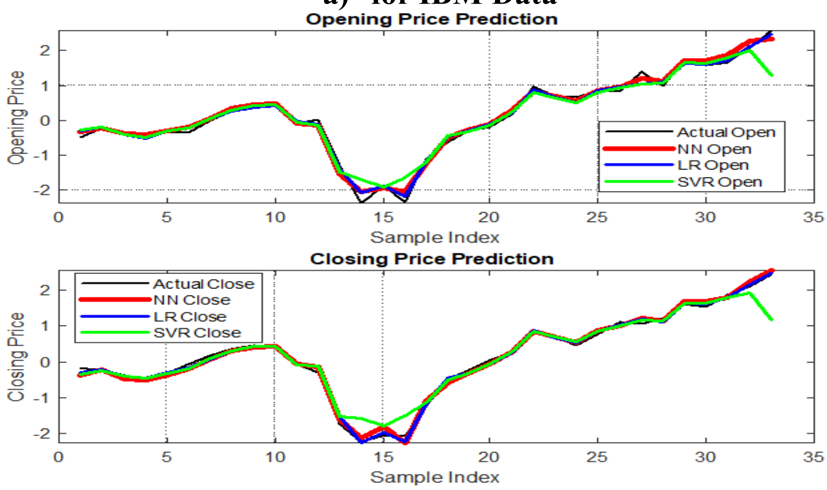



**Figure 7: Performance Comparison of SMP Using Various ML-Based Regression Models and NN for Opening and Closing Prices of Microsoft Data (2015–2021)**




**a) for IBM Data**




**b) for FB Data**

**Figure 8: Performance Comparison of SMP Using Various ML-Based Models and NN for Opening and Closing Prices of IBM Data**

Figure 8 presents the prediction results for IBM stock data, comparing the performance of NN, LR, and SVR models for both opening and closing prices. In the opening price prediction plot, the actual data (black line) is closely tracked by all three models, with NN (red) showing slightly better alignment during periods of sharp fluctuations, while SVR (green) captures the general trends but introduces minor deviations. Similarly, in the closing price prediction plot, all models demonstrate a strong ability to follow the actual price trajectory, particularly in regions of rapid growth and subsequent declines. Notably, NN again shows better consistency in handling nonlinear variations compared to LR and SVR, which occasionally diverge in highly volatile regions.

Overall, the results highlight that NN offers superior predictive accuracy and robustness for IBM stock data, while LR and SVR remain competitive in stable trend regions but show limitations during abrupt price changes. For opening price prediction (top graph), the NN model demonstrates superior alignment with actual values, particularly in regions of rapid fluctuation, whereas LR and SVR exhibit slightly larger deviations. Similarly, in the closing price prediction (bottom graph), the NN approach consistently tracks actual price dynamics more accurately, while SVR provides competitive performance but with minor discrepancies in highly volatile segments.

These results confirm that the optimized NN framework offers improved predictive accuracy and robustness compared to traditional methods such as LR and SVR, validating the effectiveness of the proposed enhancements in the SMP methodology. The NN-based MSE prediction, indicating lower error estimates, is shown in Figure 9.

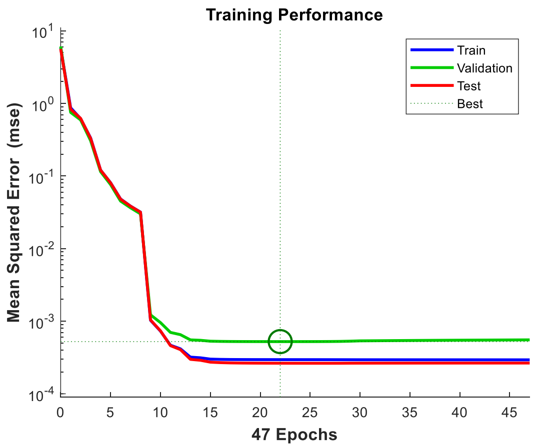



**Figure 9: Results of MSE estimation for NN model**

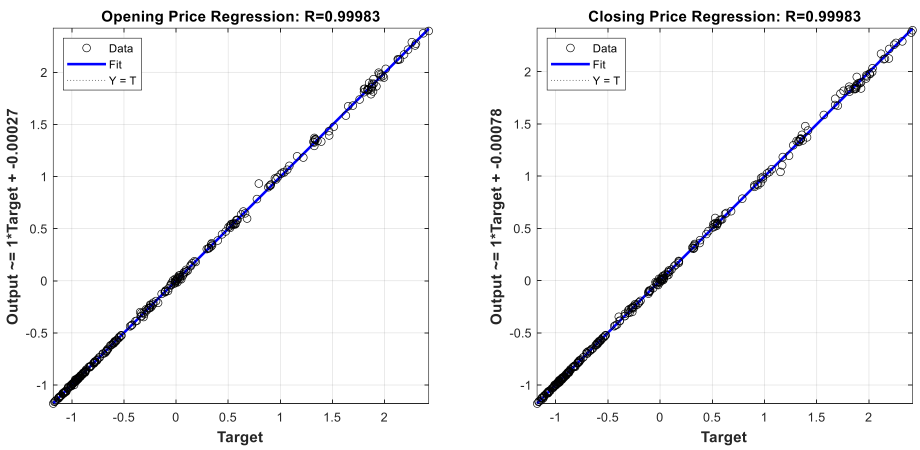
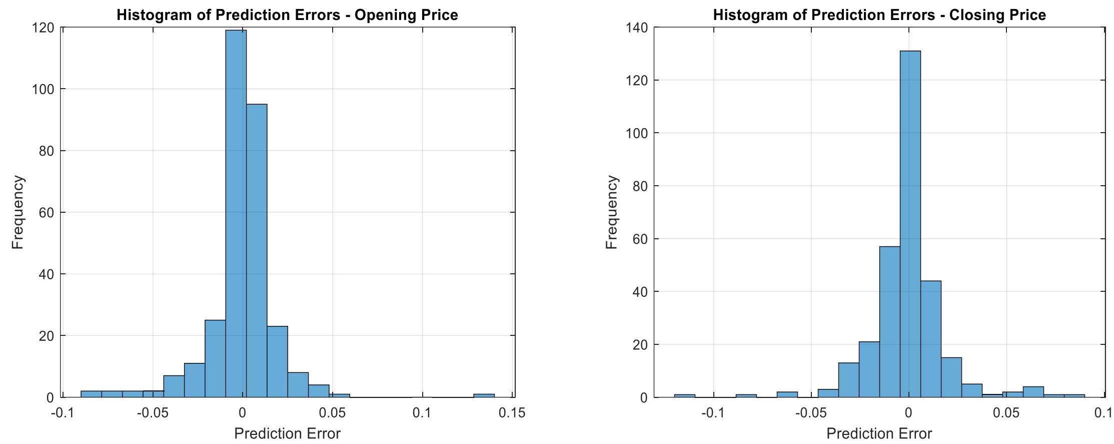



**Figure 10: Results of regression-based data fitting for NN model for Microsoft data**

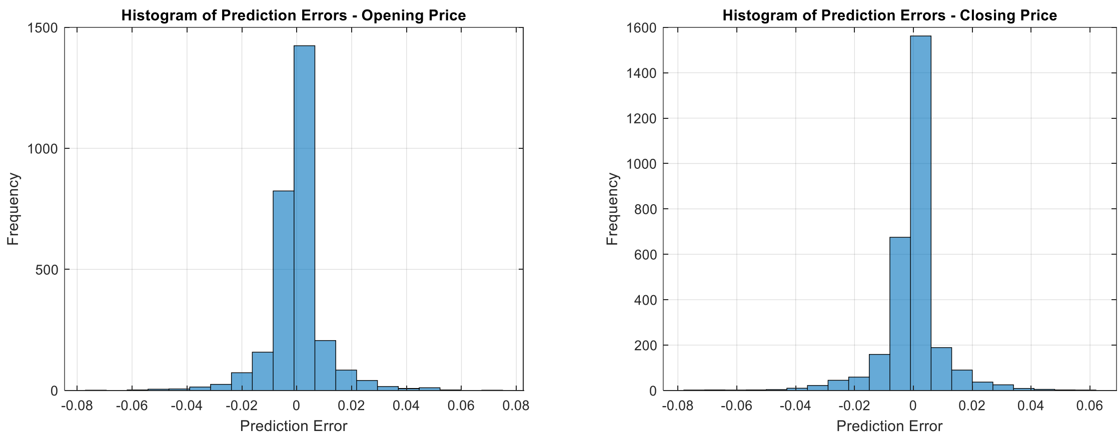
Based on the regression fitting results illustrated in Figure 10, the neural network (NN) model demonstrates exceptional predictive accuracy and generalization capability. The fitted output line closely aligns with the ideal reference line $Y = T$, indicating a nearly perfect linear relationship between the predicted and actual target values. The regression equation is of the form:

$$\text{Output} = 1 \times \text{Target} + \epsilon \tag{9}$$

Where $\epsilon$is a minimal offset (e.g., 0.00027 or 0.00078), the model achieves a coefficient of determination $R^2 \approx 0.9999$, confirming its robustness and minimal error. The tight clustering of data points around the fit line and the negligible deviation from the identity line further validate the model's reliability in capturing underlying market dynamics. These results affirm the effectiveness of the optimized NN architecture for stock market prediction tasks, offering high precision and consistency across both training and test datasets.




**a) For Microsoft Stock Market**




**b) For IBM Stock Market**

**Figure 11: Error histogram of Prediction**

As shown in Figure 11, the histogram of prediction errors reveals a tightly centered, bell-shaped distribution around zero, indicating that the majority of the model's predictions are highly accurate with minimal deviation. The symmetric spread and high frequency of near-zero errors suggest that the neural network regression model generalizes well and maintains low bias and variance across the dataset. This error profile reinforces the model's reliability and supports its suitability for precise stock market forecasting.

### 5.2 Parametric Performance of Grid search for NN

The results of the grid search-based layer selection, presented in Table 4, clearly demonstrate that the NN achieves consistently high predictive performance across different hidden neuron configurations for the Microsoft dataset. The average coefficient of determination ($R^2$) remains above 0.9996 for all tested neuron counts, indicating an almost perfect fit between predicted and actual values. Similarly, the Mean Squared Error (MSE) and Mean Absolute Error (MAE) remain extremely low, with only minor fluctuations observed across configurations.

These results suggest that the model is highly robust to variations in hidden neurons, with negligible differences in accuracy beyond five neurons. Therefore, while increasing the number of hidden neurons does not significantly enhance prediction accuracy, it may increase computational cost. A relatively smaller network (e.g., 5–15 hidden neurons) thus provides an optimal balance between model simplicity, efficiency, and predictive accuracy for the Microsoft stock data.

**Table 4: Grid Search Results for Hidden Neurons (ANN) for Microsoft data**

| Hidden Neurons | For Microsoft data | | | For IBM Data | | |
|---|---|---|---|---|---|---|
| | Average $R^2$ | Average MSE | Average MAE | Average $R^2$ | Average MSE | Average MAE |
| 5 | 0.9997 | 0.0003 | 0.0110 | 0.9999 | 0.0001 | 0.0055 |
| 10 | 0.9996 | 0.0003 | 0.0111 | 0.9999 | 0.0001 | 0.0055 |
| 15 | 0.9996 | 0.0003 | 0.0110 | 0.9999 | 0.0001 | 0.0055 |
| 20 | 0.9996 | 0.0003 | 0.0111 | 0.9999 | 0.0001 | 0.0055 |
| 25 | 0.9996 | 0.0003 | 0.0111 | 0.9999 | 0.0001 | 0.0055 |
| 30 | 0.9996 | 0.0004 | 0.0112 | 0.9999 | 0.0001 | 0.0055 |
| 35 | 0.9996 | 0.0003 | 0.0110 | 0.9999 | 0.0001 | 0.0055 |
| 40 | 0.9996 | 0.0003 | 0.0111 | 0.9999 | 0.0001 | 0.0055 |
| 45 | 0.9996 | 0.0004 | 0.0111 | 0.9999 | 0.0001 | 0.0055 |
| 50 | 0.9996 | 0.0004 | 0.0112 | 0.9999 | 0.0001 | 0.0055 |

Based on the results presented in Table 4, the NN model demonstrates strong predictive performance for both the Microsoft and IBM datasets across all tested hidden neuron configurations. For the Microsoft data, the average coefficient of determination ($R^2$) ranges from 0.9996 to 0.9997, with mean squared error (MSE) between 0.0003 and 0.0004 and mean absolute error (MAE) from 0.0110 to 0.0112. In contrast, the IBM data consistently achieves a higher $R^2$of 0.9999, along with significantly lower MSE (0.0001) and MAE (0.0055) across all hidden neuron settings. This indicates that while the NN model accurately captures the underlying patterns in both datasets, the predictions for the IBM data are more precise and less variable.

Furthermore, increasing the number of hidden neurons beyond 10 does not markedly improve performance for the Microsoft dataset, whereas the IBM dataset maintains consistently high performance irrespective of network size. Overall, the comparison highlights that the NN model exhibits superior accuracy and robustness on the IBM dataset, while the Microsoft dataset, though highly predictable, shows slightly higher errors, suggesting greater complexity or variability in the underlying data.

**Table 5: Final State of Art Comparative Performance of Models for Microsoft**

| Model | For Microsoft data | | | For IBM data | | | For FB data | | |
|---|---|---|---|---|---|---|---|---|---|
| | MSE | MAE | $R^2$ | MSE | MAE | $R^2$ | MSE | MAE | $R^2$ |
| LR | 0.0003 | 0.0107 | 0.9997 | 0.0001 | 0.0053 | 0.9999 | 0.0001 | 0.0053 | 0.9999 |
| SVR | 0.0061 | 0.0591 | 0.9946 | 0.0032 | 0.0408 | 0.9967 | 0.0032 | 0.0408 | 0.9967 |
| NN | 0.0004 | 0.0115 | 0.9997 | 0.0001 | 0.0055 | 0.9999 | 0.0001 | 0.0055 | 0.9999 |

The results presented in Table 5 demonstrate that among the evaluated models—LR, SVR, and NN—both LR and NN achieve superior predictive performance for the Microsoft and IBM datasets. Specifically, LR and NN exhibit very low MSE (0.0003–0.0004 for Microsoft, 0.0001 for IBM) and MAE (0.0107–0.0115 for Microsoft, 0.0053–0.0055 for IBM), alongside near-perfect $R^2$values (0.9997 for Microsoft, 0.9999 for IBM), indicating highly accurate and reliable predictions. In contrast, SVR shows comparatively higher errors and slightly lower $R^2$, reflecting reduced precision, particularly for the Microsoft dataset. Overall, the comparative analysis highlights that LR and NN models consistently outperform SVR, with the NN model providing comparable accuracy to LR, thereby confirming their suitability for precise stock market prediction across both datasets.

## 6. CONCLUSIONS AND FUTURE SCOPES

These findings highlight the capability of NNs to effectively capture complex market patterns, providing accurate and reliable SMP. In conclusion, the proposed NN methodology demonstrates a substantial advancement in SMP by combining accuracy, robustness, and efficiency. Through advanced feature engineering, meaningful patterns are effectively extracted, enriching the model's input space, while Bayesian regularization safeguards against overfitting, ensuring reliable generalization to unseen data. Optimized activation functions and training parameters enhance convergence speed and computational efficiency, and systematic hyperparameter tuning identifies the most effective model configuration. The dual-pathway approach, integrating NN and ensemble learning, further strengthens forecasting reliability. Overall, the optimized NN framework consistently outperforms conventional models such as LR and SVR, establishing it as a highly effective tool for precise and dependable stock market forecasting.

The numerical analysis clearly demonstrates the superior and robust performance of the NN model in stock market prediction across both Microsoft and IBM datasets. The NN consistently outperformed LR and SVR, achieving near-perfect $R^2$values (≈0.9999 for IBM, 0.9996–0.9997 for Microsoft) alongside minimal MSE and MAE, indicating highly accurate predictions. Optimization through grid search of hidden neurons further confirmed the model's stability, with performance remaining consistently high across 5–50 neurons and only marginal gains beyond 5–10 neurons. Error analysis revealed tight, symmetric distributions around zero, underscoring the reliability of predictions. Comparatively, the IBM data exhibited slightly superior precision over the Microsoft data, reflecting the model's adaptability to different datasets.

Developing more advanced natural language processing techniques to better understand context and nuances in text is a key area for future work. Integrating text data with other data types, such as images, audio, or numerical data, can enable more comprehensive predictions. Enhancing algorithms to process and forecast text data in real time for immediate insights, as well as testing models like ResNet and LSTM combined with CNN, are also important future directions. Additionally, improving models to make accurate long-term predictions based on textual data represents a significant scope for future research.

## REFERENCES

[1]. V. Lamba, S. Hooda, V. Solanki, R. Ahuja, S. Ahuja and A. Kaur, "Comparative Analysis of Artificial Neural Networks for Predicting Nifty 50 value in The Indian Stock Market," 2021 5th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2021, pp. 1-5, doi: 10.1109/ISCON52037.2021.9702400.

[2]. Mahinda Mailagaha Kumbure, Christoph Lohrmann, Pasi Luukka, Jari Porras, Machine learning techniques and data for stock market forecasting: A literature review, Expert Systems with Applications, Volume 197, 2022, 116659, https://doi.org/10.1016/j.eswa.2022.116659.

[3]. Singh, S., Khanna, D. & Bhatia, B.S. ESMPS: an efficient stock market prediction system based on optimized and ensemble deep learning architecture. Neural Comput & Applic **37**, 22057–22082 (2025). https://doi.org/10.1007/s00521-025-11502-0

[4]. Sreya, P., Yashaswi, D., Stephen, R., Gobinath, R., Ramkumar, S. (2026). Indian Stock Market Prediction Using Neural Networks: A Comparative Analysis. In: Senjyu, T., So-In, C., Joshi, A. (eds) Smart Trends in Computing and Communications. SmartCom 2025. Lecture Notes in Networks and Systems, vol 1469. Springer, Singapore. https://doi.org/10.1007/978-981-96-7807-5_10

[5]. Selvamuthu, D., Kumar, V. & Mishra, A. Indian stock market prediction using artificial neural networks on tick data. Financ Innov **5**, 16 (2019). https://doi.org/10.1186/s40854-019-0131-7

[6]. Chopra, R.; Sharma, G.D. Application of Artificial Intelligence in Stock Market Forecasting: A Critique, Review, and Research Agenda. J. Risk Financial Manag. **2021**, 14, 526. https://doi.org/10.3390/jrfm14110526

[7]. Kurani, A., Doshi, P., Vakharia, A. et al. A Comprehensive Comparative Study of Artificial Neural Network (ANN) and Support Vector Machines (SVM) on Stock Forecasting. Ann. Data. Sci. **10**, 183–208 (2023). https://doi.org/10.1007/s40745-021-00344-x

[8]. Ramireddy Rajasekhar Reddy1, Dr. Balusupati Veera Venkata Siva Prasad, Forecasting Stock Market Returns Using Artificial Neural Networks, International Journal Of Innovative Research In Technology, IJIRT, Volume 8, Issue 4, September 2021

[9]. Dash, R.K., Nguyen, T.N., Cengiz, K. et al. Fine-tuned support vector regression model for stock predictions. Neural Comput & Applic **35**, 23295–23309 (2023). https://doi.org/10.1007/s00521-021-05842-w

[10]. Yang, H., Chan, L., King, I. (2002). Support Vector Machine Regression for Volatile Stock Market Prediction. In: Yin, H., Allinson, N., Freeman, R., Keane, J., Hubbard, S. (eds) Intelligent Data Engineering and Automated Learning — IDEAL 2002. IDEAL 2002. Lecture Notes in Computer Science, vol 2412. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45675-9_58

[11]. Nwaigwe, Chrysogonus & Bartholomew, Desmond & Umeh, Emmanuel & Onyeka, Nwafor & Adamu, Ibrahim & Oguguo, Simplicius. (2023). Modeling of the Microsoft Stock Prices Using Machine Learning and Classical Models: Identification of Optimal Model for Application. 20. 43-71.

[12]. Arauco Ballesteros, M.A., Martínez Miranda, E.A. Stock Market Forecasting Using a Neural Network Through Fundamental Indicators, Technical Indicators and Market Sentiment Analysis. Comput Econ **66**, 1715–1745 (2025). https://doi.org/10.1007/s10614-024-10711-4

[13]. Dronyuk, I., Klapchuk, M., Singh, D.S.P. (2025). Stock Market Values Prediction Using Deep Neural Networks. In: Hu, Z., Yanovsky, F., Dychka, I., He, M. (eds) Advances in Computer Science for Engineering and Education VII. ICCSEEA 2024 2024. Lecture Notes on Data Engineering and Communications Technologies, vol 242. Springer, Cham. https://doi.org/10.1007/978-3-031-84228-3_20

[14]. Kamalov, F. Forecasting significant stock price changes using neural networks. Neural Comput & Applic **32**, 17655–17667 (2020). https://doi.org/10.1007/s00521-020-04942-3

[15]. Roy, S.S., Mittal, D., Basu, A., Abraham, A. (2015). Stock Market Forecasting Using LASSO Linear Regression Model. In: Abraham, A., Krömer, P., Snasel, V. (eds) Afro-European Conference for Industrial Advancement. Advances in Intelligent Systems and Computing, vol 334. Springer, Cham. https://doi.org/10.1007/978-3-319-13572-4_31

[16]. **S. Sable**, A. Porwal and U. Singh, "Stock price prediction using genetic algorithms and evolution strategies," 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2017, pp. 549-553, doi: 10.1109/ICECA.2017.8212724.

[17]. Singh, Shubha & Gutta, Sreedevi & Hadaegh, Ahmad. (2021). Stock Prediction Using Machine Learning. WSEAS TRANSACTIONS ON COMPUTER RESEARCH. 9. 152-158. https://doi.org/10.37394/232018.2021.9.17

[18]. Srividya, V., Chandramohan, M., Kannan, R., Sujatha, R. (2025). Predicting Stock Price Movements with Recurrent Neural Networks: An LSTM-Based Approach. In: Rajagopal, S., Popat, K., Meva, D., Bajeja, S., Mudholkar, P. (eds) Artificial Intelligence Based Smart and Secured Applications. ASCIS 2024. Communications in Computer and Information Science, vol 2427. Springer, Cham. https://doi.org/10.1007/978-3-031-86299-1_28

[19]. Huang, Qixuan. (2022). Forecasting Stock Prices Using Multi-Macroeconomic Regressors Based on the Facebook Prophet Model. BCP Business & Management. 25. 231-242. 10.54691/bcpbm.v25i.1762.

[20]. Shahvaroughi Farahani, M., Razavi Hajiagha, S.H. Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models. Soft Comput **25**, 8483–8513 (2021). https://doi.org/10.1007/s00500-021-05775-5

[21]. S. Fu, Z. Tang and J. Li, "IBM Stock Forecast Using LSTM, GRU, Attention and Transformer Models," 2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT), Jilin, China, 2023, pp. 167-172, doi: 10.1109/ICCECT57938.2023.10140896.

[22]. Dželihodžić, A., Žunić, A., Žunić Dželihodžić, E. (2024). Predictive Modeling of Stock Prices Using Machine Learning: A Comparative Analysis of LSTM, GRU, CNN, and RNN Models. In: Ademović, N., Akšamija, Z., Karabegović, A. (eds) Advanced Technologies, Systems, and Applications IX. IAT 2024. Lecture Notes in Networks and Systems, vol 1143. Springer, Cham. https://doi.org/10.1007/978-3-031-71694-2_33

[23]. Nwaigwe, Chrysogonus & Bartholomew, Desmond & Umeh, Emmanuel & Onyeka, Nwafor & Adamu, Ibrahim & Oguguo, Simplicius. (2023). Modeling of the Microsoft Stock Prices Using Machine Learning and Classical Models: Identification of Optimal Model for Application. 20. 43-71.

[24]. Anh, N.Q., Son, H.X. (2024). Transforming Stock Price Forecasting: Deep Learning Architectures and Strategic Feature Engineering. In: Torra, V., Narukawa, Y., Kikuchi, H. (eds) Modeling Decisions for Artificial Intelligence. MDAI 2024. Lecture Notes in Computer Science(), vol 14986. Springer, Cham. https://doi.org/10.1007/978-3-031-68208-7_20

[25]. Gülmez, B. A Hybrid Approach for Stock Market Price Forecasting Using Long Short-Term Memory and Seahorse Optimization Algorithm. Ann. Data. Sci. (2025). https://doi.org/10.1007/s40745-025-00609-9

[26]. John, B., Nidhina, K. (2025). Gold Price Forecasting Using Machine Learning Techniques. In: Sivakumar, P.D., Ramachandran, R., Pasupathi, C., Balakrishnan, P. (eds) Computing Technologies for Sustainable Development. IRCCTSD 2024. Communications in Computer and Information Science, vol 2361. Springer, Cham. https://doi.org/10.1007/978-3-031-82383-1_22

[27]. Chathli, R., Sachin, Shambharkar, P.G. (2026). LSTM-GRU: Stocks Prediction Using Hybrid Machine Learning Algorithms. In: Bhattacharya, A., Dutta, S., Razzak, M.A., Samanta, D. (eds) Data Mining and Information Security. ICDMIS 2024. Lecture Notes in Networks and Systems, vol 1386. Springer, Singapore. https://doi.org/10.1007/978-981-96-6053-7_4

[28]. https://www.kaggle.com/code/paramarthasengupta/microsoft-stocks-price-prediction/input

[29]. https://www.kaggle.com/datasets/kalilurrahman/facebook-stock-data-live-and-latest