International Journal of Applied Information Systems (IJAIS) ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12 - No. 47, March 2025 - www.ijais.org

Enhancing Financial Time Series Predictions with a Hybrid BNN-LSTM Approach

Anika Tahsin Biva
Department of Applied Mathematics
University of Dhaka
Dhaka 1000, Dhaka, Bangladesh

A.B.M. Shahadat Hossain
Department of Applied Mathematics
University of Dhaka
Dhaka 1000, Dhaka, Bangladesh

Md. Shafiul Alom Khan
Department of Applied Mathematics
University of Dhaka
Dhaka 1000, Dhaka, Bangladesh

Iqbal Habib
Department of Statistics and Data Science
Jahangirnagar University
Dhaka 1342, Dhaka, Bangladesh

ABSTRACT

Accurate forecasting of stock market indices is vital for guiding investment strategies and mitigating financial risks. This study proposes a novel hybrid Bayesian Neural Network-Long Short-Term Memory (BNN-LSTM) model to enhance the predictive accuracy of Dow Jones Industrial Average (DJIA) closing price forecasts. By integrating the uncertainty quantification capabilities of Bayesian Neural Networks with the sequential learning strengths of Long Short-Term Memory networks, the hybrid model addresses the challenges of modeling complex, nonlinear, and time-dependent financial data. Comparative experiments were conducted using Bayesian Neural Networks, LSTM, Random Forest (RF), Gradient Boosting Machine (GBM), and the hybrid BNN-LSTM model on historical DJIA data spanning January 1, 2005, to December 31, 2022, for training, and January 1, 2023, to January 31, 2024, for testing. The hybrid BNN-LSTM consistently outperformed all competing models across multiple evaluation metrics. These results underscore the model's superior ability to capture complex market dynamics and its robustness in forecasting financial time series. This study contributes a powerful tool for financial decision-making and sets the foundation for future advancements in hybrid deep learning models for stock market analysis.

General Terms:

Hybrid Model, Closing Price Forecasting

Keywords:

Stock market forecasting, hybrid BNN-LSTM model, neural networks, deep learning in finance, time series prediction

1. INTRODUCTION

Financial markets play a pivotal role in the global economy by influencing investment decisions, corporate strategies, and economic policies. Accurate forecasting of stock market indices is essential for investors, analysts, and policymakers to make informed decisions and manage risks effectively. Stock market data is inherently complex which is characterized by nonlinearity, volatility, and intricate temporal dependencies [23]. Because of these issues, simple statistical models and other standard methods of forecasting can't fully explain how financial time series behave.

The application of machine learning techniques for stock market forecasting has gained significant attention in recent years, with various models being employed to enhance prediction accuracy. According to Ticknor[1], a Bayesian regularized artificial neural network improves accuracy and generalization in stock price forecasting by reducing overfitting and performing effectively without extensive preprocessed data. Selvamuthu et al. [2] highlight the challenges of predicting dynamic financial time series data and demonstrate that neural networks with various learning algorithms achieve up to 99.9% accuracy on tick data but show reduced accuracy on 15-minute datasets. Althelaya et al.[3] looked at how bidirectional and stacked LSTM systems could be used to predict financial time series. They proved that these models are better than simple LSTMs and shallow neural networks. This change is very important for getting rid of the vanishing gradient problem. Yan et al.[4] propose a Bayesian-regularised ANN optimized with particle swarm optimization (PSO), demonstrating improved reliability and accuracy in forecasting Shanghai composite index prices by reducing overfitting and optimizing model parameters effectively. Park et al.[5] modeled a stock market prediction system called LSTM–Forest which combines LSTM and random forest to improve performance and make the results easier to understand. In trading tests, LSTM–Forest achieved lower prediction errors and higher profits compared to basic models and earlier deep learning methods. Vazirani et al.[6] conducted a comparative analysis of various machine learning models for stock market prediction and proposed a new linear regression-based hybrid model that significantly reduced prediction errors and improves efficiency in forecasting. Bola et al.[7] compare Artificial Neural Networks (ANN) and Bayesian Networks (BN) for forecasting the Nigerian Stock Exchange index and they found that BN outperforms ANN in short-term predictions by demonstrating effective forecasting without extensive market data. Omar et al.[8] propose machine learning models which included AR-DNN and AR-RF for stock index forecasting and they demonstrated their superior performance over traditional models, especially during high stock price fluctuations caused by Covid-19, with practical implications for investors and policymakers. Garg[9] uses Bayesian regularization neural networks to predict gold prices over a ten-year period, achieving a mean percentage error of 1% when comparing forecasted prices with actual values.

Vullam et al.[10] propose a hybrid model combining Generative Adversarial Networks (GANs), reinforcement learning, and Bayesian optimization for stock market prediction, showing superior performance over existing models like Stock-GAN and Multi-Model Hybrid Prediction Algorithm (MM-HPA). Hajiaghajani[11] proposes a hybrid system combining Bayesian networks and the Markov model for predicting daily stock market trends, leveraging Bayesian networks to define variable relationships and the Markov model to forecast market trends. The system demonstrates high efficiency in evaluation. Pandya et al.[12] propose a hybrid model combining ARIMA and LSTM

for stock market prediction, comparing its performance with standalone ARIMA and LSTM models. The effectiveness of the hybrid model is tested across five businesses from different industries, with performance evaluated using various error metrics. Hossain and Kaur[13] compare XGBoost and LSTM for stock price forecasting, highlighting XGBoost's strength in tabular data processing and LSTM's ability to capture time dependencies, with potential for future research into hybrid models combining both strengths.
Adeleye et al.[14] review various machine learning models for stock market forecasting, comparing traditional time-series models like ARIMA and MACD with advanced models like SVM, ANN, and ensemble methods, highlighting the importance of feature selection, model accuracy, and strategies to mitigate overfitting. Wu [15] presents a stock trading action prediction model based on neural networks and Bayesian optimization, achieving superior performance compared to ResNet and XGBoost models, with a utility score higher by 2111 and 3179, respectively, using the Jane Street dataset. Maeda et al.[16] propose a Bayesian convolutional neural network (CNN) to predict short-term stock price trends, effectively addressing prediction uncertainty and outperforming conventional CNN and logistic regression models in terms of reliability. Su and Zhao[17] propose a Bayesian financial panel data model based on neural network algorithms to improve the accuracy of stock price forecasting, highlighting the challenges and effectiveness of BP neural networks in stock market analysis.
Alam et al. [18] introduce a robust hybrid LSTM-DNN model for stock market prediction, validated across 26 real-life datasets, achieving exceptional performance with an average R-squared score of 0.98606, MAE of 0.0210, and MSE of 0.00111, setting a new standard in stock price forecasting. Satyaveer et al.[19] propose a novel hybrid ARFIMA-LSTM model combined with news sentiment analysis for stock market prediction, outperforming traditional models like SVM, Random Forest, ARIMA, KNN, GRNN, and LSTM in accuracy.
The main contribution of this study is the development of a novel hybrid Bayesian Neural Network-Long Short-Term Memory (BNN-LSTM) model for accurately forecasting the closing price of the Dow Jones Industrial Average (DJIA). By combining the uncertainty quantification of BNN with the sequential learning strength of LSTM, the proposed model outperforms traditional machine learning techniques (including BNN, LSTM, Random Forest, and Gradient Boosting Machine) in terms of predictive accuracy, as demonstrated through comprehensive evaluation using various metrics. This study highlights the model's effectiveness in capturing complex market dynamics and its potential for real-world financial time series forecasting.
Section 2 details the data collection methods , data preprocessing, and the description of the ML models using in this study. Section 3 focuses on the implementation of the ML models and the detailed analysis of the results. Section 4 summarizes the findings from the study and the conclusions drawn from the implementations and results of the machine learning models. Limitations encountered during the research are acknowledged, which may influence future studies.

2. BACKGROUND THEORY & METHODOLOGY

Figure 1 presents an overview of the methodology employed in this study.

2.1 Data Collection and Preparation

The closing price of DJIA were collected from Yahoo Finance using the 'pandas' and 'yfinance' libraries of Python.
Initially, the dataset was loaded into a data frame, and the 'Date' column was parsed to ensure it was correctly recognized as a DateTime object.

2.1.1 Data Scaling. To prepare the data for neural network input, the closing prices were scaled using the MinMaxScaler. This normalization process transformed the data to a specific range [0,1], enhancing convergence speed and performance stability. The scaling transformation is given by:

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

where $X_{\max}$ and $X_{\min}$ are the maximum and minimum values of the training data respectively [20].

2.1.2 Data Parsing and Splitting. The dataset was divided into training and test sets, with data up to December 31, 2022, used for training and data from January 1, 2023, onward used for testing. This division is essential to simulate real-world forecasting where future data points are unknown during model training.
Let X_t represent the time series data at time t. The training set $\{X_t\}_{t=1}^{n}$ and the test set $\{X_t\}_{t=n+1}^{N}$ are defined as:

$$\{X_t\}_{t=1}^{n} \quad \text{for} \quad t \leq \text{2022-12-31} \tag{2}$$

$$\{X_t\}_{t=n+1}^{N} \quad \text{for} \quad t \geq \text{2023-01-01} \tag{3}$$

2.2 Model Development

2.2.1 Hybrid BNN LSTM. The Hybrid Bayesian Neural Network (BNN) and Long Short-Term Memory (LSTM) model combines the benefits of LSTM's sequential data processing and Bayesian Neural Network's probabilistic uncertainty estimation. LSTM networks are designed to capture temporal dependencies in time series data by processing sequences step-by-step. The output of the LSTM model is a hidden state h_t that represents the learned temporal features. Bayesian Neural Networks (BNNs) treat the weights of the network as probabilistic variables, providing a probability distribution over the predictions rather than a single point estimate. This helps quantify uncertainty in predictions[21].
The LSTM component of the hybrid model is responsible for capturing temporal dependencies in time series data. The LSTM components at time step t are:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{4}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5}$$
$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{6}$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{7}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{8}$$
$$h_t = o_t \cdot \tanh(c_t) \tag{9}$$

where f_t is forget gate, i_t is input gate, $\tilde{c}_t$ is candidate cell state, c_t is cell state, o_t is output gate, h_t is hidden state, W_f are the weights, σ is the sigmoid activation, and x_t is the input at time t. The hidden state h_t captures the temporal dependencies of the input sequence at time t.
In the BNN component, the weights W are treated as random variables with prior distributions. The model predicts a distribution of outputs rather than a single value. The output at time step t is:

$$y_t = f(x_t, W) = W \cdot x_t + b \tag{10}$$

where $W \sim \mathcal{N}(0, \sigma^2)$ is the prior distribution for the weights[22].
Given the observed data D, we compute the posterior distribution $p(W|D)$ over the weights using Markov Chain Monte Carlo (MCMC).
The model doesn't output a single value but rather a distribution over predictions, providing uncertainty estimates.

International Journal of Applied Information Systems (IJAIS) ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12 - No. 47, March 2025 - www.ijais.org

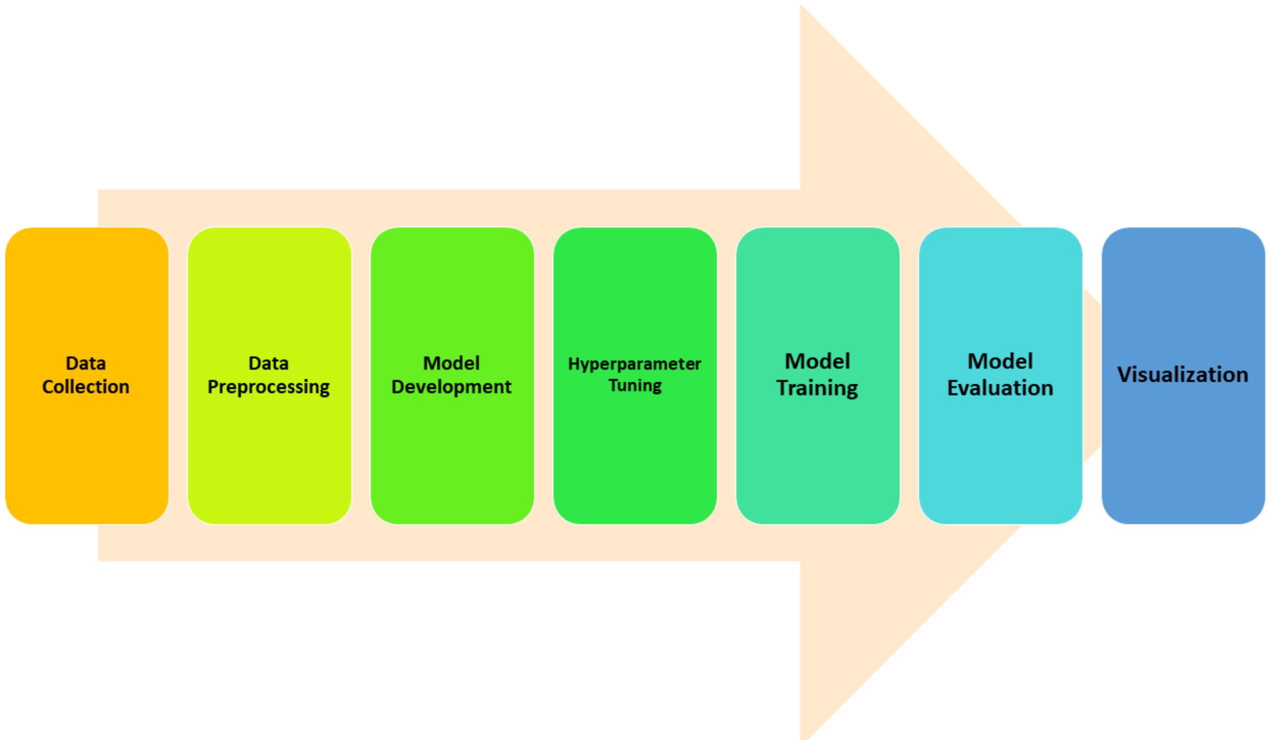

Fig. 1. Methodology

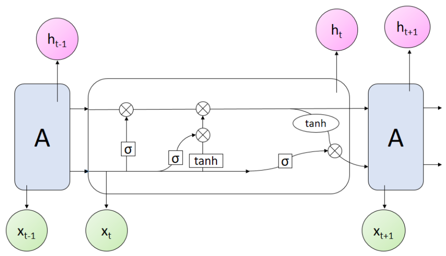

Fig. 2. LSTM Architecture

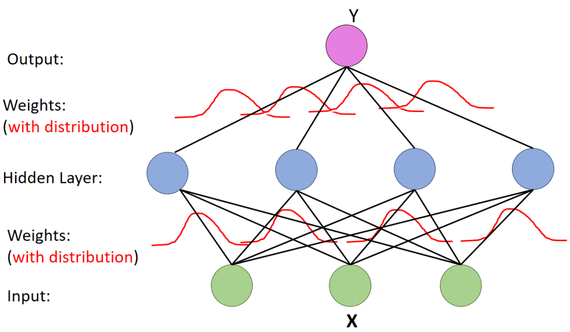

Fig. 3. BNN Architecture

In the hybrid model, the LSTM component's output h_t is passed to the BNN to make a probabilistic prediction:

$$y_t = f(h_t, \theta_{BNN}) \tag{11}$$

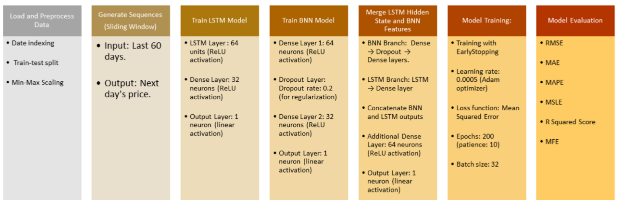

Fig. 4. Hybrid BNN LSTM Methodology

where θ_{BNN} are the parameters of the Bayesian Neural Network. The output y_t is not a point prediction but a probability distribution that represents the uncertainty of the prediction.

Figure 4 depicts the methodology of the Hybrid BNN-LSTM model utilized in this study. The training of the hybrid model requires a loss function that combines the prediction error and the uncertainty regularization term for the BNN. The loss function is:

$$\mathcal{L}(\theta_{LSTM}, \theta_{BNN}) = \frac{1}{N}\sum_{t=1}^{N}(y_t^{true} - y_t^{pred})^2 + \lambda \cdot \mathcal{R}(\theta_{BNN}) \tag{12}$$

where y_t^{true} is the actual target value at time step t, y_t^{pred} is the predicted output from the hybrid model, $\mathcal{R}(\theta_{BNN})$ is the L2 regularization term for the BNN , λ is a regularization parameter that controls the strength of the penalty on the BNN.
The LSTM component is trained first to learn the temporal dependencies in the time series data. After the LSTM is trained, its output h_t is passed through the BNN. The BNN learns the distribution of the weights using Bayesian inference (via MCMC). The parameters of both the LSTM and BNN, θ_{LSTM} and θ_{BNN}, are optimized together using gradient-based methods like stochastic gradient descent, minimizing the combined loss function.

International Journal of Applied Information Systems (IJAIS) ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12 - No. 47, March 2025 - www.ijais.org

2.2.2 Random Forest(RF). Random Forest (RF) is an ensemble learning algorithm that constructs multiple decision trees and aggregates their outputs to predict a continuous target variable for regression. It is particularly effective in reducing overfitting and improving predictive accuracy.

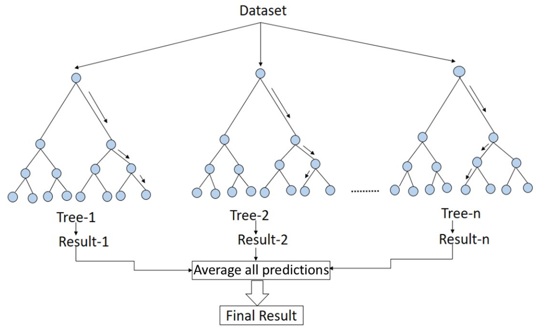

Fig. 5. RF Methodology

RF creates T decision trees, each trained on a bootstrap sample D_b drawn from the original dataset D. For a dataset with n samples:

$$D_b = \{(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)\}, \quad k = n. \tag{13}$$

At each node split, the algorithm selects a random subset of features m (where $m < p$, and p is the total number of features) to determine the best split:

$$m = \frac{p}{3}.$$

This randomness reduces correlation between trees.
Each decision tree $h_t(x)$ independently predicts a numerical output y_t for the input x. The prediction is based on the average value of the training samples in the leaf node where x lands:

$$y_t = \frac{1}{|N|} \sum_{i \in N} y_i \tag{14}$$

where N represents the set of samples in the leaf node. The final prediction $\hat{y}$ is the average of all tree predictions:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} h_t(x). \tag{15}$$

2.2.3 Gradient Boosting Machines(GBM). GBM builds a strong predictive model by sequentially improving predictions through the addition of decision trees for regression. Each tree in the sequence tries to correct the errors made by the previous trees, and the final model is an ensemble of these weak learners. Unlike Random Forest, which builds trees independently, GBM constructs trees one at a time, where each tree is trained to correct the errors made by the previous one. This process is known as boosting.
The GBM algorithm starts with an initial prediction, usually the mean of the target variable y_i for regression tasks:

$$F_0(x) = \frac{1}{N} \sum_{i=1}^{N} y_i, \tag{16}$$

where N is the number of training samples.
At each iteration m, a new decision tree $h_m(x)$ is fitted to the residuals (the difference between the current prediction and actual values):

$$r_i^{(m)} = y_i - F_{m-1}(x_i), \tag{17}$$

where $r_i^{(m)}$ is the residual for the i-th sample at iteration m, and $F_{m-1}(x_i)$ is the prediction from the previous iteration.
A decision tree $h_m(x)$ is fitted to the residuals:

$$h_m(x) = \arg\min_h \sum_{i=1}^{N} \left[r_i^{(m)} - h(x_i) \right]^2. \tag{18}$$

This tree minimizes the squared error of the residuals, effectively learning to predict the residuals.
The model is updated by adding the new tree's prediction to the previous model:

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x), \tag{19}$$

where η is the learning rate, controlling the contribution of each tree.

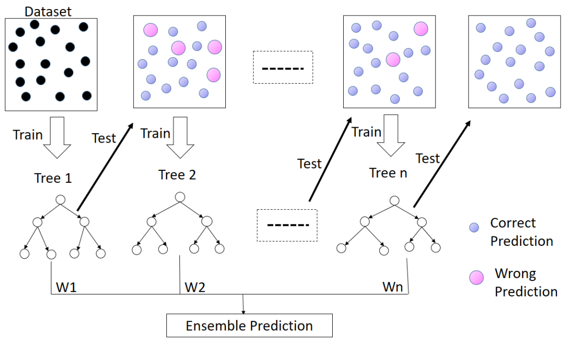

Fig. 6. GBM Methodology

The final prediction is the sum of all the predictions from each tree:

$$\hat{y}_i = F_M(x_i) = \sum_{m=1}^{M} \eta \cdot h_m(x_i), \tag{20}$$

where M is the total number of trees, and η is a user-defined learning rate that controls the step size in the boosting process.

2.3 Evaluation Metrics

After training, the model's predictions are evaluated against the test set. The predicted values and actual values are inverse transformed to their original scale to facilitate comparison. Evaluation metrics are computed to quantify the model's accuracy. The predictions of the models on the test dataset were evaluated using several key metrics:

(i) **Root Mean Square Error (RMSE)** : RMSE measures the square root of the average squared differences between predicted and actual values, emphasizing larger errors.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{21}$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of observations.

(ii) **Mean Absolute Error (MAE)**: MAE calculates the average absolute differences between predicted and actual values, reflecting overall prediction accuracy.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{22}$$

(iii) **Mean Absolute Percentage Error (MAPE)**: MAPE represents the average percentage error between predicted and actual values, scaled by actual values.

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{23}$$

(iv) **Mean Squared Logarithmic Error (MSLE)**: MSLE penalizes the squared differences between the logarithms of predicted and actual values, focusing on relative differences.

$$\text{MSLE} = \frac{1}{N} \sum_{i=1}^{N} (\log(Y_i + 1) - \log(\hat{Y}_i + 1))^2 \tag{24}$$

(v) **R-squared Score (R^2)**: R^2 indicates how well the predictions explain the variance in the actual data, with 1 being a perfect fit.

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{N} (Y_i - \bar{Y})^2} \tag{25}$$

(vi) **Mean Forecast Error (MFE)** MFE measures the average signed difference between predicted and actual values, showing bias in over- or underestimation.

$$\text{MFE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i) \tag{26}$$

3. RESULTS AND DISCUSSION

The training data are the closing prices of the DJIA from January 1, 2005, to December 31, 2022. This study forecast the closing price from January 1, 2023, to January 31, 2024, using this training data. Thus, 5% of the data is utilized as test data, and 95% of the data is used as training data.

Fig. 7. DJIA Closing Price

The test period's lowest, mean, and median values are all much greater than the training period's, indicating a robust growing tendency in the DJIA over time. The test dataset's dropped standard deviation suggests that recent market fluctuations have been less erratic than past patterns. While the training period has greater skewness, reflecting past market volatility, the test period's mean and median are closer, indicating a more stable market performance. The test period's maximum value exceeds the training period's peak, indicating that the DJIA has recently hit new highs. This suggests that the DJIA has experienced consistent growth over time, with the most recent period showing more stable and less volatile performance. This pattern can be an indication of market dynamics or economic resiliency, which have propelled the index's steady increases(refer Table 1 and Figure 7).

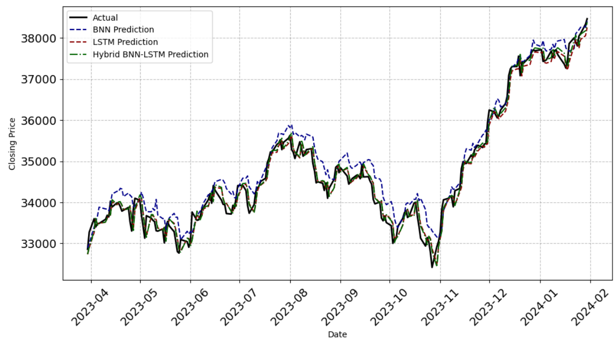

Fig. 8. BNN, LSTM and Hybrid BNN-LSTM Forecasting

Fig. 9. RF Forecasting

Fig. 10. GBM Forecasting

The Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) measure the overall and average magnitude of errors in predictions, respectively. A lower value indicates better accuracy. The Hybrid BNN-LSTM model achieves the lowest RMSE (217.2332) and MAE (170.3741), demonstrating its ability to produce predictions with the smallest errors. LSTM follows closely with an RMSE of 223.5647 and an MAE of 179.1898, while BNN shows the highest RMSE (376.6339) and MAE (301.6685), reflecting relatively poor predictive accuracy. RF and GBM exhibit moderate RMSE $(303.9989$ and $304.5793)$ but achieve remarkably low MAE values $(102.1724$ and 103.8916, respectively). However, their inconsistency between RMSE and MAE suggests that these models may struggle with larger deviations in predictions(refer Table 2, Figure 8,Figure 9,Figure 10).

The Mean Absolute Percentage Error (MAPE) and Mean Squared Logarithmic Error (MSLE) focus on relative and logarithmic differences, providing insights into the proportional accuracy of the models. The Hybrid BNN-LSTM model records the lowest MAPE (0.0049%) and MSLE $(4.0058e-05)$, high-

International Journal of Applied Information Systems (IJAIS) ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12 - No. 47, March 2025 - www.ijais.org

Table 1. Statistics of DJIA Closing Price

	Observations	Min	Max	Mean	Median	SD
Training	4531	6547.0498	36799.6484	18065.4062	15967.0303	7789.7708
Test	270	31819.1406	38467.3086	34389.9147	33981.9336	1481.2066

Table 2. Summary of Model Performance.

	RMSE	MAE	MAPE	MSLE	R Squared	MFE
BNN	376.6339	301.6685	0.0088%	0.0001	0.9342	-0.7377
LSTM	223.5647	179.1898	0.0052%	$4.2147e-05$	0.9768	0.0422
Hybrid BNN-LSTM	217.2332	170.3741	0.0049%	$4.0058e-05$	0.9781	-0.0292
RF	303.9989	102.1724	0.2722%	$6.6318e-05$	0.9577	97.0628
GBM	304.5793	103.8916	0.2771%	$6.6578e-05$	0.9576	96.8417

lighting its robustness in minimizing relative errors across the dataset. LSTM also performs well with a MAPE of 0.0052% and MSLE of $4.2147e-05$, making it a strong contender. In contrast, BNN exhibits higher values for both metrics (MAPE: 0.0088%, MSLE: 0.0001), and RF and GBM show significantly higher MAPE (0.2722% and 0.2771%, respectively), indicating a weaker ability to manage proportional prediction accuracy(refer Table 2, Figure 8,Figure 9,Figure 10).

R-squared measures how well the model explains the variance in the target variable, with values closer to 1 indicating better performance. The Hybrid BNN-LSTM achieves the highest R-squared value (0.9781), suggesting it captures nearly all variations in the data. LSTM follows closely with an R-squared of 0.9768, while RF and GBM perform moderately well, with R-squared values of 0.9577 and 0.9576, respectively. BNN has the lowest R-squared (0.9342), indicating that it struggles to capture the underlying patterns in the data(refer Table 2, Figure 8,Figure 9,Figure 10).

Mean Forecast Error (MFE) evaluates the bias of predictions, with values close to zero indicating minimal systematic error. The Hybrid BNN-LSTM model achieves an MFE of -0.0292, the closest to zero, implying it is nearly unbiased. LSTM also performs well, with an MFE of 0.0422, slightly overestimating the predictions. In contrast, RF and GBM show significant positive bias, with MFE values of 97.0628 and 96.8417, respectively, indicating consistent over-predictions. BNN exhibits a noticeable negative bias (-0.7377), indicating under-predictions(refer Table 2, Figure 8,Figure 9,Figure 10).

The Hybrid BNN-LSTM model consistently outperforms the other models across all metrics. Its low RMSE, MAE, MAPE, and MSLE demonstrate superior accuracy and robustness, while its high R-squared value reflects excellent explanatory power. The nearly unbiased MFE further confirms its reliability in producing accurate forecasts. While the LSTM model also performs well, its slightly higher errors and marginally lower R-squared make it less optimal compared to the Hybrid BNN-LSTM. RF and GBM, despite having competitive R-squared values, exhibit significant biases and higher proportional errors, making them less reliable. Lastly, the BNN model, with the highest errors and lowest R-squared, is the least effective for this forecasting task. Thus, the Hybrid BNN-LSTM is the most accurate and dependable model for predicting the Dow Jones closing prices(refer Table 2, Figure 8,Figure 9,Figure 10).

4. CONCLUSION

This study presents a comprehensive analysis of different machine learning models for forecasting the closing price of the Dow Jones Industrial Average (DJIA), including Bayesian Neural Network (BNN), Long Short-Term Memory (LSTM), Hybrid BNN-LSTM, Random Forest (RF), and Gradient Boosting Machine (GBM). Based on rigorous evaluation using RMSE, MAE, MAPE, MSLE, R-squared, and MFE, the Hybrid BNN-LSTM model demonstrated superior performance compared to all other models, achieving the lowest error metrics and the highest R-squared. This highlights the advantage of combining the strengths of both Bayesian Neural Networks and LSTM in capturing complex temporal dependencies and uncertainty in stock market data. The results indicate that the Hybrid BNN-LSTM model can be a reliable tool for financial forecasting, offering significant improvements in prediction accuracy.

There are some drawbacks with this study, even though the results are good. First, this study failed to incorporate enough economic indicators, trading volumes, or sentiment research in this model. These would have made it work better. Second, the models were trained with data from 2005 to 2022, but they were only tried for a short time, from 2023 to 2024. This means that the results might not work in other business or market situations. Third, the Hybrid BNN-LSTM model did a good job, but it needs a lot of computer power and might not work well in real time without being made more efficient. Lastly, the study only used the DJIA stock market measure to test the forecasting models. They may have done better or worse with other markets or financial assets. These areas could be studied further in the future to enhance the accuracy and usefulness of the models.

5. REFERENCES

[1] Ticknor, J. L. (2013). *A Bayesian regularized artificial neural network for stock market forecasting. Expert Systems With Applications, 40(14), 5501–5506.* `https://doi.org/10.1016/j.eswa.2013.04.013`

[2] *Selvamuthu, D., Kumar, V., & Mishra, A. (2019). Indian stock market prediction using artificial neural networks on tick data. Financial Innovation, 5(1).* `https://doi.org/10.1186/s40854-019-0131-7`

[3] *Althelaya, K. A., El-Alfy, E.-S. M., & Mohammed, S. (2018). Evaluation of bidirectional LSTM for short-and long-term stock market prediction. 2018 9th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 151-156.* `https://doi.org/10.1109/IACS.2018.8355458`

[4] *Yan, D., Zhou, Q., Wang, J., & Zhang, N. (2016). Bayesian regularisation neural network based on artificial intelligence optimisation. International Journal of Production Research, 55(8), 2266–2287.* `https://doi.org/10.1080/00207543.2016.1237785`

[5] *Park, H. J., Kim, Y., & Kim, H. Y. (2021). Stock market forecasting using a multi-task approach integrating long short-term memory and the random forest framework. Applied*

Soft Computing, 114, 108106. https://doi.org/10.1016/j.asoc.2021.108106

[6] Vazirani, S., Sharma, A., & Sharma, P. (2020). Analysis of various machine learning algorithm and hybrid model for stock market prediction using python. 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 203–207. https://doi.org/10.1109/ICSTCEE49637.2020.9276859

[7] Bola, A. A., Adesola, A. G., Omidiora, E. O., & Adebisi, A. A. (2013). Forecasting movement of the Nigerian stock exchange all share index using artificial neural and Bayesian networks. Journal of Finance and Investment Analysis, 2(1).

[8] Omar, A. B., Huang, S., Salameh, A. A., Khurram, H., & Fareed, M. (2022). Stock market forecasting using the random forest and deep neural network models before and during the COVID-19 period. Frontiers in Environmental Science, 10. https://doi.org/10.3389/fenvs.2022.917047

[9] Garg, S. (2021). Forecasting of Gold Prices Using Bayesian Regularization Neural Network. In Nath, V., & Mandal, J. (Eds.), Nanoelectronics, Circuits and Communication Systems. Lecture Notes in Electrical Engineering, vol. 692, Springer, Singapore. https://doi.org/10.1007/978-981-15-7486-3_15

[10] Vullam, N., Yakubreddy, K., Vellela, S. S., Basha Sk, K., R. B., V., & Santhi Priya, S. (2023). Prediction and analysis using a hybrid model for stock market. 2023 3rd International Conference on Intelligent Technologies (CONIT), Hubli, India, 1–5. https://doi.org/10.1109/CONIT59222.2023.10205638

[11] Hajiaghajani, A. (2023). Designing a combined Markov-bayesian model in order to predict stock prices in the stock exchange. International Journal of Innovation in Management Economics and Social Sciences, 3(2), 33–41. https://doi.org/10.59615/ijimes.3.2.33

[12] Pandya, A., Kapoor, V., & Joshi, A. (2024). Stock market prediction using ARIMA-LSTM hybrid. Journal of Information and Optimization Sciences, 45(4), 1129–1139. https://doi.org/10.47974/jios-1697

[13] Hossain, S., & Kaur, G. (2024). Stock market prediction: XGBoost and LSTM comparative analysis. 2024 3rd International Conference on Artificial Intelligence for Internet of Things (AIIoT), Vellore, India, 1–6. https://doi.org/10.1109/AIIoT58432.2024.10574794

[14] Adeleye, N. R. A., Tubokirifuruar, N. T. S., Bello, N. B. G., Ndubuisi, N. N. L., Asuzu, N. O. F., & Owolabi, N. O. R. (2024). Machine learning for stock market forecasting: A review of models and accuracy. Finance & Accounting Research Journal, 6(2), 112–124. https://doi.org/10.51594/farj.v6i2.783

[15] Wu, M. (2021). Financial transaction forecasting using neural network and Bayesian optimization. 2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI), Fuzhou, China, 193–196. https://doi.org/10.1109/CEI52496.2021.9574535

[16] Maeda, I., et al. (2020). Learning uncertainty in market trend forecast using Bayesian neural networks. In Bucciarelli, E., Chen, S. H., & Corchado, J. (Eds.), Decision Economics: Complexity of Decisions and Decisions for Complexity. Advances in Intelligent Systems and Computing, vol. 1009, Springer, Cham. https://doi.org/10.1007/978-3-030-38227-8_24

[17] Su, Z., & Zhao, Z. (2022). A Bayesian financial panel data model based on neural network algorithm. 2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC), Tumkur, Karnataka, India, 1–5. https://doi.org/10.1109/ICMNWC56175.2022.10031896

[18] Alam, K., Bhuiyan, M. H., Haque, I. U., Monir, M. F., & Ahmed, T. (2024). Enhancing stock market prediction: A robust LSTM-DNN model analysis on 26 real-life datasets. IEEE Access, 12, 122757–122768. https://doi.org/10.1109/ACCESS.2024.3434524

[19] Satyaveer, P., Patel, H., Chandra, P., Pal, P., & Singh, S. K. (2023). A new hybrid model ARFIMA-LSTM combined with news sentiment analysis model for stock market prediction. 2023 Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 1–5. https://doi.org/10.1109/ICAECT57570.2023.10118349

[20] Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. Applied Soft Computing, 97, 105524. Elsevier.

[21] Izmailov, P., Vikram, S., Hoffman, M. D., & Wilson, A. G. G. (2021). What are Bayesian neural network posteriors really like? In Proceedings of the International Conference on Machine Learning (pp. 4629–4640). PMLR.

[22] Hernández-Lobato, J. M., & Adams, R. (2015). Probabilistic backpropagation for scalable learning of Bayesian neural networks. In Proceedings of the International Conference on Machine Learning (pp. 1861–1869). PMLR.

[23] Biva, A. T., Khan, M. S. A., & Hossain, A. S. (2024). A Comparative Study of ARIMA, Artificial Neural Networks, and Kalman Filter Models for Dhaka Stock Exchange Forecasting. GANIT: Journal of Bangladesh Mathematical Society, 44(1), 29–58.