International Journal of Computer Applications (0975 - 8887)
Volume 186 - No.74, March 2025

Benchmarking Hybrid ANN-LSTM and Physics-Informed Neural Networks for Forecasting Stock Market Prices

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

This study presents a novel approach for forecasting stock market prices by combining Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM) models into a hybrid ANN-LSTM framework. This study focus on forecasting the closing prices of the S&P 500 and Toronto Stock Exchange (TSX) indices, evaluating the performance of the proposed hybrid model against traditional ANN, LSTM, and Physics-Informed Neural Network (PINN) models. The hybrid ANN-LSTM model demonstrates superior forecasting accuracy, outperforming the individual models and the PINN in terms of multiple evaluation metrics. The training dataset spans from January 1, 2005, to December 31, 2020, while the testing period covers January 1, 2021, to January 31, 2024. The results highlight the potential of hybrid deep learning models, specifically the ANN-LSTM combination, in enhancing stock market prediction accuracy, representing a significant advancement over conventional methods.

General Terms

Stock Market Forecasting, Hybrid ANN-LSTM Model

Keywords

Artificial Neural Networks, Long Short-Term Memory, Physics-Informed Neural Networks, Deep Learning in Finance, Time Series Prediction

1. INTRODUCTION

Accurate forecasting of stock market prices is a critical task for investors, policymakers, and researchers due to its impact on financial decision-making [16]. In recent years, machine learning (ML) and deep learning (DL) models have gained significant attention for their potential to predict market trends and asset prices. Among these, Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM) networks have proven to be effective tools for time series forecasting [5]. However, despite their individual strengths, both models exhibit limitations when applied to complex, nonlinear financial data. ANN struggles with capturing temporal dependencies, while LSTM, though capable of handling sequential data, may not fully harness the nuances of market dynamics.

Ajoku et al. [4] developed an ensemble ANN model using ensemble averaging to address high variance issues in stock market forecasting, demonstrating superior predictive accuracy compared to traditional multilayer perceptron models . Srivastava et al. [6] highlighted the use of artificial neural networks with a Backpropagation algorithm and Multilayer Feedforward Network to forecast stock prices, offering a robust approach to uncover hidden patterns in complex market data . Ballesteros et al. [7] demonstrated that incorporating market sentiment into a neural network model, alongside fundamental and technical analysis variables, improves prediction accuracy by 1.5% for 66% of S&P 500 companies, emphasizing tailored variable selection based on market sectors.

Inani et al. [8] conducted a comparative analysis of RW, ARIMA, and ANN models for forecasting the Nifty Fifty index, revealing ANN's superior accuracy using MAE and RMSE metrics, providing valuable insights for investors and financial stakeholders. Singh et al. [1] developed a Long Short-Term Memory (LSTM) framework to forecast stock prices, emphasizing its ability to analyze complex market dynamics and empower investors with data-driven decision-making insights . Qiu et al. [2] proposed an attention-based LSTM model with wavelet denoising to enhance stock price prediction accuracy, achieving superior performance with R-squared values above 0.94 and MSE below 0.05 on S&P 500 and DJIA datasets . Sivadasan et al. [3] demonstrated that GRU and LSTM models, optimized with technical indicators like SMA, EMA, and RSI, significantly improved stock market forecasting accuracy, achieving lower MAPE values compared to existing models. Patel et al. [9] explored various LSTM model architectures for stock price prediction, highlighting that a model using 11 years of data, 60 or 100 previous days, and a $70 : 10 : 20$ data split outperformed others, emphasizing the importance of hyper-parameter tuning.

Ku et al. [10] proposed integrating investor domain knowledge with an LSTM model for stock price prediction, showing superior accuracy and performance in a 100-stock simulation compared to strategies based on random technical indicator selection. Karima et al. [11] applied an LSTM-based recurrent neural network to forecast daily stock prices for four Moroccan companies, achieving promising performance with MSE and MAE validation scores ranging from 0.0412 to 0.1230. Girish et al. [12] explored the use of Long Short-Term Memory (LSTM) models for stock market prediction, emphasizing their ability to capture temporal dependencies and patterns in volatile stock price data. The study highlights LSTM's effectiveness in forecasting future market trends based on historical data and other relevant indicators .

Ge [13] develops a hybrid predictive model combining time series analysis and machine learning techniques, demonstrating superior prediction accuracy and adaptability across various global stock indices, including the S&P 500, NASDAQ 100, and Nikkei 225, with promising results for mitigating losses and enhancing returns. Arora et al. [14] propose a hybrid LSTM-CNN model that integrates temporal and image features from stock time series, demonstrating improved prediction accuracy compared to individual models like LSTM, CNN, and Naïve Bayes, with LSTM showing the best performance. Musa & Joshua [15] demonstrate that the hybrid ARIMA-Artificial Neural Network model outperforms individual ARIMA and ANN models in forecasting Nigerian stock market returns, recommending it for improved accuracy in predictions.

Physics-Informed Neural Networks (PINNs) introduce a novel approach by integrating physical laws into the learning process, potentially enhancing the interpretability and robustness of predictions. Although the application of PINNs in stock market forecasting is still emerging, their ability to incorporate domain knowledge may provide a significant advantage in understanding market dynamics. This is particularly relevant in volatile markets where traditional models may struggle to adapt to rapid changes. The integration of such physics-informed approaches with existing neural network methodologies could pave the way for more resilient forecasting models. Moreover, the literature indicates a growing consensus on the necessity of high-quality, diverse datasets for training these models effectively.

This study proposes a hybrid ANN-LSTM model that combines the strengths of both architectures, aiming to enhance the forecasting accuracy for stock market prices. This hybrid model is applied to forecast the closing prices of two major financial indices: the S&P 500 and the Toronto Stock Exchange (TSX). By integrating ANN's ability to capture patterns with LSTM's proficiency in handling temporal dependencies, the hybrid model aims to provide more robust predictions compared to individual models.

In addition to the hybrid ANN-LSTM, this study also compare its performance with a Physics-Informed Neural Network (PINN), a novel approach that incorporates physical laws into neural network models to improve prediction accuracy. This comparison aims to evaluate the relative advantages of using a hybrid deep learning model over other contemporary techniques, specifically PINN, in stock market forecasting.

The primary contributions of this paper are as follows:

—The development of a hybrid ANN-LSTM model for stock market forecasting, which combines the strengths of both ANN and LSTM to improve predictive performance.
—A detailed comparison of the hybrid ANN-LSTM model with traditional ANN, LSTM, and PINN models on forecasting the closing prices of the S&P 500 and TSX.
—A comprehensive evaluation using multiple performance metrics, including RMSE, MAE, MAPE, MSLE, R-squared Score, and Mean Forecast Error (MFE), to assess the accuracy and reliability of the proposed model.

Section 2 details the data collection methods , data preprocessing, and the description of the ML models using in the 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

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

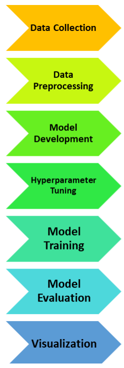

Fig. 1. Methodology

2.1 Data Collection and Preparation

The closing price of S&P 500 and TSX 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 Date-Time 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:

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (1)$$

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

2.1.2 Data Parsing and Splitting. The objective is to develop a model that successfully generalizes to new data while fitting the training set of data.
The dataset was divided into training and test sets, with data up to December 31, 2020, used for training and data from January 1, 2021, 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:

$$\{X_t\}_{t=1}^{n} \quad \text{for} \quad t \leq \text{2020-12-31} \tag{2}$$

$$\{X_t\}_{t=n+1}^{N} \quad \text{for} \quad t \geq \text{2021-01-01} \tag{3}$$

2.2 Model Development

2.2.1 Hybrid ANN-LSTM Model. A Hybrid ANN-LSTM model combines the strengths of both Artificial Neural Networks (ANNs) and Long Short-Term Memory (LSTM) networks to create a model capable of handling both complex non-linearities (using the ANN) and long-term dependencies in sequential data (using the LSTM). This hybrid approach is particularly effective when dealing with time-series forecasting, where the data has both immediate and long-range dependencies.
ANN is good for capturing non-linear relationships between inputs and outputs, and it is typically used for extracting patterns from data [17]. LSTM is specialized for learning temporal dependencies in sequential data. It can capture long-term dependencies and retain memory of past inputs.
In a Hybrid ANN-LSTM model, the ANN component learns the non-linear patterns from the raw input features. The LSTM component processes the sequential output of the ANN, capturing the temporal dependencies.

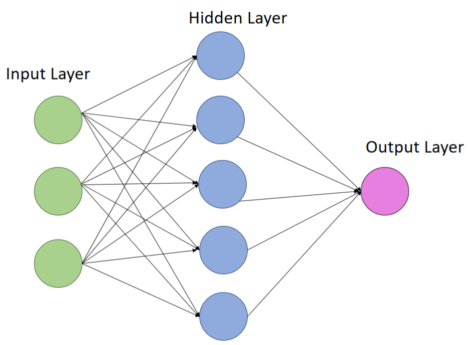

Fig. 2. ANN Architecture

Suppose we have an input vector $\mathbf{x_t} = [x_1, x_2, \ldots, x_n]$ at time step t. In the first step, this input is processed by the ANN component [19]. For an ANN layer with a nonlinear activation function ϕ (ReLU), the output $\mathbf{a_t}$ at time step t is computed as:

$$\mathbf{a_t} = \phi(W_{\text{ann}} \cdot \mathbf{x_t} + b_{\text{ann}}) \tag{4}$$

where W_{ann} is the weight matrix for the ANN, $\mathbf{x_t}$ is the input vector at time step t, b_{ann} is the bias term for the ANN, ϕ is the activation function applied element-wise (e.g., ReLU, sigmoid) [18].

After the ANN layer, we feed the output $\mathbf{a_t}$ into the LSTM layer to capture the temporal dependencies. The LSTM uses the equations described in the previous answer to update its internal state and compute the hidden state $\mathbf{h_t}$ at each time step. Let the LSTM's input at time step t be the output from the ANN: $\mathbf{a_t}$. The LSTM cell components are:

$$f_t = \sigma(W_f \cdot [\mathbf{h}_{t-1}, \mathbf{a_t}] + b_f) \tag{5}$$
$$i_t = \sigma(W_i \cdot [\mathbf{h}_{t-1}, \mathbf{a_t}] + b_i) \tag{6}$$
$$\tilde{C}_t = \tanh(W_C \cdot [\mathbf{h}_{t-1}, \mathbf{a_t}] + b_C) \tag{7}$$
$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{8}$$
$$o_t = \sigma(W_o \cdot [\mathbf{h}_{t-1}, \mathbf{a_t}] + b_o) \tag{9}$$
$$\mathbf{h_t} = o_t \cdot \tanh(C_t) \tag{10}$$

where f_t forget gate, i_t input gate, $\tilde{C}_t$ candidate cell state, C_t cell state, o_t output gate, $\mathbf{h_t}$ hidden state, W_f, W_i, W_C, W_o are the weight matrices for the forget, input, candidate, and output gates respectively, $\mathbf{h}_{t-1}$ is the previous hidden state, $\mathbf{a_t}$ is the ANN output at time step t, b_f, b_i, b_C, b_o are the corresponding bias terms [20].

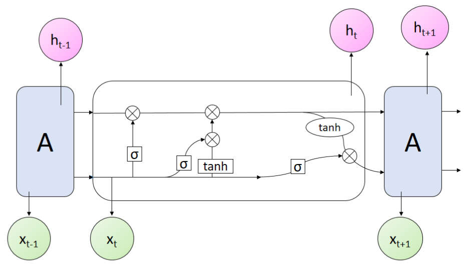

Fig. 3. LSTM Architecture

Once the LSTM processes the sequential data, the output hidden state $\mathbf{h_t}$ is passed through an output layer. This layer is typically a fully connected neural network (ANN) that provides the final prediction for the time series value at time step t.
Let's denote the final output of the network at time step t as $\hat{y}_t$:

$$\hat{y}_t = W_{\text{out}} \cdot \mathbf{h_t} + b_{\text{out}} \tag{11}$$

where W_{out} is the weight matrix for the output layer, b_{out} is the bias term for the output layer.
The hybrid model combines the output from the ANN component and the LSTM component. The final prediction $\hat{y}_t$ is obtained after passing the hidden state $\mathbf{h_t}$ from the LSTM through a fully connected layer.
Figure 4 depicts the methodology of the Hybrid ANN-LSTM model utilized in this study The training of the hybrid model involves minimizing a loss function L, typically the Mean Squared

Error (MSE) for regression tasks, over the training data. The training process uses gradient descent (or variants like Adam) to update all the weights in the model, including those of the ANN and LSTM components.

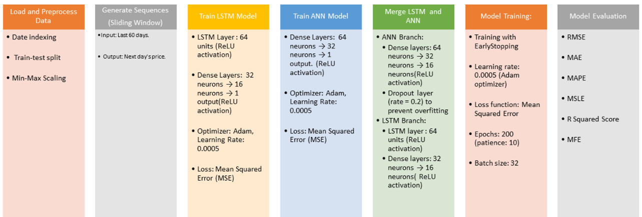

Fig. 4. Hybrid ANN-LSTM Methodology

The loss function L for time series forecasting is usually defined as:

$$L = \frac{1}{T} \sum_{t=1}^{T} (\hat{y}_t - y_t)^2 \tag{12}$$

where T is the number of time steps in the training set, $\hat{y}_t$ is the predicted value at time step t, y_t is the true value at time step t.
This loss is minimized through back propagation, where gradients are computed for the weights of both the ANN and the LSTM components and used to adjust the weights.

2.2.2 Physics Informed Neural Network(PINN) Model. Physics-Informed Neural Networks (PINNs) are a class of machine learning models that integrate physical laws, expressed as partial differential equations (PDEs) or other mathematical constraints, into the loss function of a neural network. This allows PINNs to leverage domain knowledge while training, providing robustness in scenarios with limited or noisy data. In the context of stock market forecasting, the dynamics of prices can be approximated by stochastic differential equations (SDEs) or autoregressive models. For simplicity, let the system be described by a governing equation:

$$\frac{\partial S(t)}{\partial t} + \mathcal{F}(S(t), \theta) = 0, \tag{13}$$

where $S(t)$ is the stock price at time t, $\mathcal{F}$ is a function representing market dynamics (e.g., drift and volatility terms in a stochastic model), θ represents model parameters or coefficients.
For example, $\mathcal{F}$ could take the form of a Black-Scholes equation or a simplified autoregressive model.
A fully connected neural network $u(t; w)$, parameterized by weights and biases w, is employed to approximate $S(t)$. The input to the network is t, and the output is the predicted stock price $S_{\text{NN}}(t)$.
The architecture typically includes:

—Input layer: Represents time t,

—Hidden layers: Capture nonlinear relationships using activation functions like ReLU or tanh,

—Output layer: Predicts stock price $S_{\text{NN}}(t)$.

The loss function in PINNs incorporates two components:

—**Data Loss**: Ensures that the network predictions fit the observed stock prices at discrete time points.

$$\mathcal{L}_{\text{data}} = \frac{1}{N} \sum_{i=1}^{N} \left| S_{\text{NN}}(t_i) - S(t_i) \right|^2, \tag{14}$$

where $S(t_i)$ is the actual observed price at time t_i, and N is the number of observed data points.

—**Physics Loss**: Enforces the satisfaction of the governing equation $\frac{\partial S(t)}{\partial t} + \mathcal{F}(S(t), \theta) = 0$ over a set of collocation points $\{t_c\}$, which may or may not overlap with the observed data points.

$$\mathcal{L}_{\text{physics}} = \frac{1}{M} \sum_{j=1}^{M} \left| \frac{\partial S_{\text{NN}}(t_c^j)}{\partial t} + \mathcal{F}(S_{\text{NN}}(t_c^j), \theta) \right|^2, \tag{15}$$

where M is the number of collocation points.

The total loss is a weighted combination of these components:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{data}} \mathcal{L}_{\text{data}} + \lambda_{\text{physics}} \mathcal{L}_{\text{physics}}, \tag{16}$$

where λ_{data} and λ_{physics} are weights controlling the trade-off between data fitting and physical consistency.

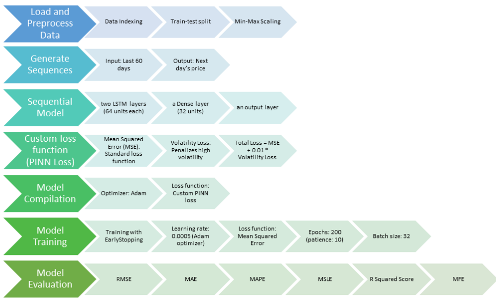

Fig. 5. PINN Methodology

The PINN is trained using gradient-based optimization algorithms, such as Adam , to minimize the total loss $\mathcal{L}_{\text{total}}$. During each training iteration:

—The neural network predicts stock prices $S_{\text{NN}}(t)$.

—The derivatives $\frac{\partial S_{\text{NN}}(t)}{\partial t}$ are computed using automatic differentiation.

—The loss components $\mathcal{L}_{\text{data}}$ and $\mathcal{L}_{\text{physics}}$ are calculated and backpropagated to update the weights w.

Figure 5 illustrates the methodology of the Physics-Informed Neural Network (PINN) implemented in this study. The training procedure of Physics-Informed Neural Networks (PINNs) involves minimizing a composite loss function that integrates data-driven and physics-informed components. First, the network is initialized with random weights, and the input time points (or other independent variables) are fed into the neural network to generate predicted outputs, such as stock prices. The data loss ensures that the predictions closely match observed stock prices at specific data points, while the physics loss enforces the governing equation (e.g., a stochastic or autoregressive model) at collocation points, which may extend beyond the observed data. Automatic differentiation is used to

compute derivatives of the predicted outputs, required for evaluating the physics loss. The total loss, a weighted sum of the data and physics losses, is minimized using gradient-based optimization algorithms like Adam . During each iteration, gradients are computed via backpropagation, and the network's weights are updated to improve both data fitting and adherence to the physical laws. Once the loss converges, the trained PINN can predict future stock prices while respecting the underlying market dynamics encoded in the governing equations.
After training, the PINN is used to forecast stock prices by evaluating $S_{\text{NN}}(t)$ at future time points $t > t_{\text{train}}$. The model inherently respects the underlying physical dynamics encoded in $\mathcal{F}$, which enhances generalization for extrapolative tasks.

2.3 Evaluation

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. It penalizes large errors more heavily than small ones. Lower RMSE indicates better accuracy.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{17}$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of observations [21].

(ii) **Mean Absolute Error (MAE)**: MAE calculates the average of the absolute differences between predicted and actual values. It provides a straightforward measure of the average error without emphasizing outliers. Lower MAE indicates better performance.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{18}$$

(iii) **Mean Absolute Percentage Error (MAPE)**: MAPE is a percentage-based metric that computes the average absolute error as a percentage of the actual values. It is useful for understanding relative errors but can be biased when actual values are close to zero. Lower MAPE indicates better performance [22].

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{19}$$

(iv) **Mean Squared Logarithmic Error (MSLE)**: MSLE focuses on the relative differences between predicted and actual values by comparing their logarithms. It penalizes under-predictions more than over-predictions and is effective when dealing with data spanning multiple scales.

$$\text{MSLE} = \frac{1}{N}\sum_{i=1}^{N}(\log(Y_i + 1) - \log(\hat{Y}_i + 1))^2 \tag{20}$$

(v) **R-squared Score (R^2)**: R^2 represents the proportion of the variance in the actual values that is predictable from the model. It ranges from 0 to 1, where values closer to 1 indicate a better fit. Negative values may occur for poorly fitted models.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{N}(Y_i - \bar{Y})^2} \tag{21}$$

(vi) **Mean Forecast Error (MFE)**: MFE measures the average bias in predictions by computing the mean difference between predicted and actual values. Positive or negative values indicate systematic over-prediction or under-prediction, respectively.

$$\text{MFE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) \tag{22}$$

3. RESULTS & DISCUSSION

The training data are the closing prices of the S&P 500 and TSX from January 1, 2005, to December 31, 2020. The closing price is forecasted from January 1, 2021, to January 31, 2024, using this training data. Thus, 16% of the data is utilized as test data, and 84% of the data is used as training data.

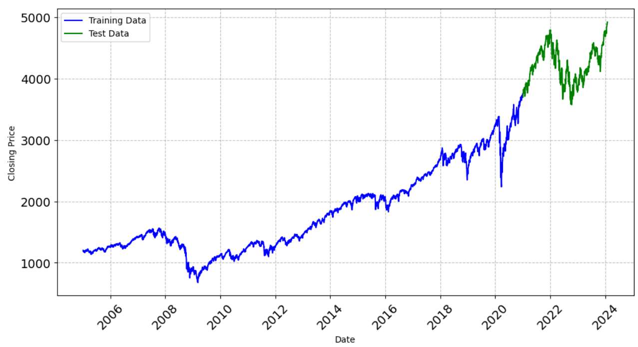

Fig. 6. S&P 500 Closing Price

The test dataset has a higher mean and maximum closing price compared to the training dataset, indicating that the prices in the test period are generally higher than those in the training period. The training dataset has a much lower mean and median value compared to the test period, suggesting that the S&P 500 might have had lower prices in the earlier years (used in training data). The standard deviation (SD) is much higher in the training dataset, which suggests more variability in prices during the training period. The test dataset shows less variability in comparison(refer Table 1 and Figure 6).
The Hybrid ANN-LSTM model consistently outperforms all other models across nearly all metrics, demonstrating the lowest RMSE, MAE, and MFE, as well as the highest R-squared value, which indicates that it provides the most accurate and precise predictions. With an R-squared value of 0.9546, it explains 95.46% of the variance in the data, showcasing a high degree of fit. Furthermore, its MAPE of 0.0115% and MSLE of 0.0002 reflect minimal percentage and logarithmic deviations from the actual values(refer Table 2 and Figure 7).
LSTM follows closely, providing robust results, with a slightly higher RMSE and MAE than the hybrid model but still offering excellent accuracy and a high R-squared value (0.9510). On the other hand, the ANN model exhibits lower performance than both LSTM and Hybrid ANN-LSTM, showing higher RMSE and MAE,

Table 1. Statistics of S&P 500 Closing Price

	Observations	Min	Max	Mean	Median	SD
Training	4028	676.5300	3756.0701	1813.1928	1525.5850	692.1432
Test	773	3577.0300	4927.9302	4233.6370	4223.7002	295.5127

Table 2. Summary of Model Performance(S&P 500)

	RMSE	MAE	MAPE	MSLE	R Squared	MFE
ANN	81.7169	66.2887	0.0158%	0.0004	0.9182	−1.4311
LSTM	63.2422	50.8636	0.0119%	0.0002	0.9510	−0.8194
Hybrid ANN-LSTM	60.8746	48.5111	0.0115%	0.0002	0.9546	−0.8097
PINN	108.5009	96.5399	2.2258%	0.0006	0.8559	−92.6963

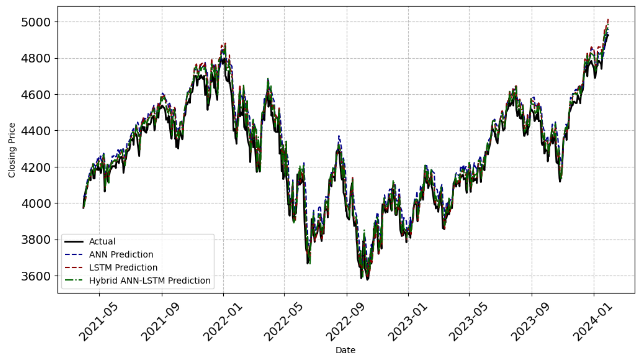

Fig. 7. ANN, LSTM and Hybrid ANN-LSTM Forecasting (S&P 500)

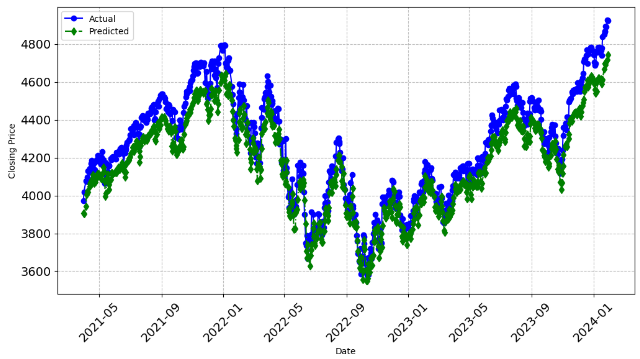

Fig. 8. PINN Forecasting (S&P 500)

though it still maintains reasonable predictive ability(refer Table 2 and Figure 7).

In contrast, the PINN model performs the weakest across all metrics, with the highest RMSE (108.5009), MAE (96.5399), and MAPE (2.2258%), suggesting that while it incorporates physics-based constraints, it struggles to capture the complexities of the S&P 500 data. Additionally, the negative MFE for PINN (-92.6963) highlights a significant underestimation bias, further reinforcing its relatively poor fit compared to the machine learning models(refer Table 2 and Figure 8).

Overall, the Hybrid ANN-LSTM model proves to be the most effective and reliable for forecasting S&P 500 closing prices, offering superior predictive accuracy and model performance across all evaluated criteria(refer Table 2).

The test dataset has higher values for the mean, median, and maximum closing prices, suggesting that the market experienced an

Fig. 9. TSX Closing Price

upward trend during the period represented in the test data. The training dataset has a wider range of variability (higher standard deviation), which indicates that the closing prices fluctuated more during the earlier years in the dataset. The mean and median values in the test dataset are closer to each other compared to the training dataset, indicating that the distribution of closing prices in the test period is less skewed. The minimum price in the test dataset is significantly higher than in the training dataset, reflecting a possible market growth or a different market condition during the test period(refer Table 3 and Figure 9).

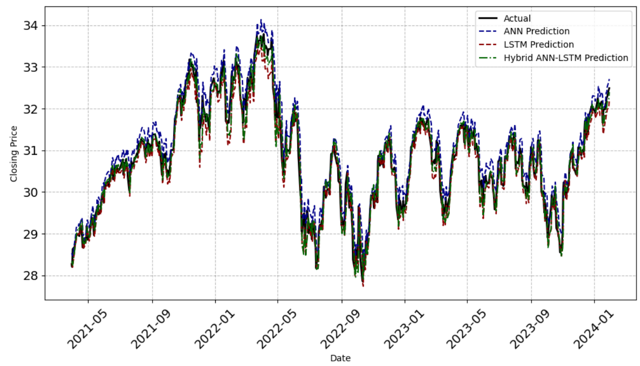

Fig. 10. ANN, LSTM and Hybrid ANN-LSTM Forecasting(TSX)

The Hybrid ANN-LSTM model shows the best performance among the models, with the lowest RMSE (0.2838) and MAE (0.2125), indicating that it offers the most accurate predictions. It also achieves the lowest MAPE (0.0069%), demonstrating that the

Table 3. Statistics of TSX Closing Price

	Observations	Min	Max	Mean	Median	SD
Training	4017	11.5699	27.0	19.7199	19.6900	3.2614
Test	772	26.0200	33.7900	30.5535	30.6800	1.5097

Table 4. Summary of Model Performance(TSX)

	RMSE	MAE	MAPE	MSLE	R Squared	MFE
ANN	0.3825	0.3071	0.0100%	0.0001	0.9039	-0.8834
LSTM	0.3113	0.2382	0.0077%	$9.5643e-05$	0.9364	0.4614
Hybrid ANN-LSTM	0.2838	0.2125	0.0069%	$8.0752e-05$	0.9471	0.1067
PINN	0.5832	0.4586	1.4765%	0.0003	0.7767	-0.4335

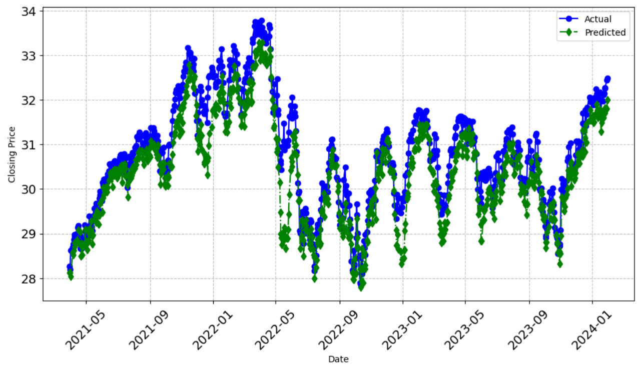

Fig. 11. PINN Forecasting(TSX)

model's errors are minimal relative to the actual values(refer Table 4 and Figure 10).
The R-squared value of 0.9471 suggests that the Hybrid ANN-LSTM model explains 94.71% of the variance in the TSX data, showing a strong fit to the observed values. Additionally, the positive MFE (0.1067) indicates that the model tends to slightly overestimate the closing prices, but this is negligible given its overall accuracy(refer Table 4 and Figure 10).
The LSTM model also performs well, with an RMSE of 0.3113, MAE of 0.2382, and an R-squared value of 0.9364, showing a strong correlation with the actual data, but still falling short of the Hybrid ANN-LSTM model in terms of precision. In comparison, the ANN model has higher RMSE (0.3825) and MAE (0.3071), but still provides reasonable predictions with an R-squared value of 0.9039, though it is less accurate than the LSTM and Hybrid ANN-LSTM models(refer Table 4 and Figure 10).
The PINN model, while incorporating physical constraints, lags behind in terms of performance. It exhibits the highest RMSE (0.5832) and MAE (0.4586), as well as a relatively high MAPE of 1.4765%, suggesting that the model struggles to accurately predict the TSX closing prices. Additionally, the PINN model has the lowest R-squared value of 0.7767, meaning it explains only 77.67% of the data's variance, and the negative MFE (-0.4335) indicates that the model tends to underestimate the closing prices(refer Table 4 and Figure 11).
Overall, the Hybrid ANN-LSTM model proves to be the most effective for forecasting TSX closing prices, outperforming both individual models (ANN and LSTM) and the Physics-Informed Neural Network (PINN) in terms of accuracy, predictive power, and model fit(refer Table 4).

4. CONCLUSIONS

This study aimed to forecast the closing prices of major stock indices, specifically the S&P 500 and TSX, using various machine learning models: Artificial Neural Networks (ANN), Long Short-Term Memory networks (LSTM), Hybrid ANN-LSTM, and Physics-Informed Neural Networks (PINN). Among the models tested, the Hybrid ANN-LSTM approach demonstrated superior forecasting accuracy across both datasets. It outperformed other models, including the individual ANN and LSTM models, by yielding the lowest RMSE, MAE, and MAPE, while providing a high R-squared value. The model's ability to combine the strengths of both ANN and LSTM allows it to effectively capture both short-term and long-term dependencies in stock market data. While the PINN model included physical constraints to improve the interpretability of the results, it did not perform as well as the hybrid approach, likely due to the challenges of incorporating such constraints in stock market predictions. The findings highlight the potential of hybrid models, especially when combining deep learning techniques such as ANN and LSTM, for improving stock market forecasting. This research provides valuable insights for future developments in financial time series prediction and presents a robust framework for tackling the complexity of stock market behavior using machine learning.
Despite its promising results, this research has some limitations. The study focused only on closing prices, without incorporating additional market factors like trading volume or macroeconomic indicators, which could further enhance predictive accuracy. Additionally, the use of historical data from 2005–2024 may not fully account for future market shifts or unprecedented events. Future research could address these limitations by incorporating diverse features, extending the training dataset, and exploring alternative hybrid architectures or refinements to PINN-based approaches.

5. REFERENCES

[1] Singh, K. U., Mishra, U. D., Verma, A., Kumar, L., Singh, T., & Varshney, N. (2024). *Forecasting Stock Prices with Long Short-Term Memory Networks. 2024 International Conference on Automation and Computation (AUTOCOM), Dehradun, India, 646–651.* `https://doi.org/10.1109/AUTOCOM60220.2024.10486109`

[2] *Qiu, J., Wang, B., & Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. PLoS ONE, 15(1), e0227222.* `https://doi.org/10.1371/journal.pone.0227222`

[3] *Sivadasan, E. T., Sundaram, N. M., & Santhosh, R. (2024). Stock market forecasting using deep learning with long short-term memory and gated recurrent unit. Soft Com-*

puting, 28(4), 3267–3282. `https://doi.org/10.1007/s00500-023-09606-7`

[4] *Ajoku, K. K., Nwokonkwo, O. C., John-Otumu, A. M., & Oleji, C. P. (2021). A Model for Stock Market Value Forecasting using Ensemble Artificial Neural Network. Journal of Advances in Computing Communications and Information Technology, 2, 1–13.* `https://doi.org/10.37121/jaccit.v2.162`

[5] *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.*

[6] *Srivastava, Y., Chauhan, A. S., Vaishnav, R., & Bharadwaj, K. (2023). Forecasting Market Price of Stock Using Artificial Neural Network. 2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET), Ghaziabad, India, 544–547.* `https://doi.org/10.1109/ICSEIET58677.2023.10303500`

[7] *Ballesteros, M. A. A., & Miranda, E. A. M. (2024). Stock market forecasting using a neural network through fundamental indicators, technical indicators and market sentiment analysis. Computational Economics.* `https://doi.org/10.1007/s10614-024-10711-4`

[8] *Inani, S. K., Pradhan, H., Arora, S., Nagpal, A., & Junior, P. O. (2023). Forecasting Stock Market Using Artificial Neural Networks: A Performance Analysis. 2023 Global Conference on Information Technologies and Communications (GCITC), Bangalore, India, 1–6.* `https://doi.org/10.1109/GCITC60406.2023.10426212`

[9] *Patel, A., Ukani, V., & Thakkar, P. (2024). Time Series Forecasting in Financial Market: Long Short-Term Memory (LSTM) Approach for Stock Price Prediction. 2024 IEEE Region 10 Symposium (TENSYMP), New Delhi, India, 1–6.* `https://doi.org/10.1109/TENSYMP61132.2024.10752158`

[10] *Ku, C. S., Xiong, J., Chen, Y., Cheah, S. D., Soong, H. C., & Por, L. Y. (2023). Improving Stock Market Predictions: An Equity Forecasting Scanner Using Long Short-Term Memory Method with Dynamic Indicators for Malaysia Stock Market. Mathematics, 11(11), 2470.* `https://doi.org/10.3390/math11112470`

[11] *Karima, L., Mimoun, B., & Jawad, E. G. (2024). Stock Market Forecasting Using Long Short-Term Memory Recurrent Neural Networks: A Case Study of Moroccan Companies. 2024 Sixth International Conference on Intelligent Computing in Data Sciences (ICDS), Marrakech, Morocco, 1–5.* `https://doi.org/10.1109/ICDS62089.2024.10756354`

[12] *Girish, L., Raviprakash, M. L., Thara, D. K., Prathibha, T. S., & Rashmi, T. V. (2023). Stock Market Time Series Forecasting using Long Short-Term Memory. 2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS), Kalaburagi, India, 1–6.* `https://doi.org/10.1109/ICIICS59993.2023.10421259`

[13] *Ge, Q. (2024). Enhancing stock market forecasting: A hybrid model approach for accurate prediction of S&P 500 and CSI 300 future prices. Expert Systems With Applications, 260, 125380.* `https://doi.org/10.1016/j.eswa.2024.125380`

[14] *Arora, A., Upadhyay, Y., Shukla, S., & Saket. (2024). Forecasting Stock Price by LSTM-CNN Hybrid Model and Compares Deep Learning Models. 2024 IEEE International Conference on Smart Power Control and Renewable Energy (ICSPCRE), Rourkela, India, 1–6.* `https://doi.org/10.1109/ICSPCRE62303.2024.10675145`

[15] *Musa, Y., & Joshua, S. (2020). Analysis of ARIMA-Artificial Neural Network Hybrid Model in forecasting of stock market returns. Asian Journal of Probability and Statistics, 42–53.* `https://doi.org/10.9734/ajpas/2020/v6i230157`

[16] *Oukhouya, H., & El Himdi, K. (2023). Comparing Machine Learning Methods—SVR, XGBoost, LSTM, and MLP—For Forecasting the Moroccan Stock Market. IOCMA 2023, MDPI, Basel, Switzerland.*

[17] *Sharma, D. K., Hota, H. S., Brown, K., & Handa, R. (2021). Integration of genetic algorithm with artificial neural network for stock market forecasting. International Journal of System Assurance Engineering and Management, 13(S2), 828–841.*

[18] *Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. arXiv preprint arXiv:1811.12808*

[19] *Kinga, D., Adam, J. B., & others. (2015). A method for stochastic optimization. International Conference on Learning Representations (ICLR), San Diego, California, 6.*

[20] *Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2017). LSTM fully convolutional networks for time series classification. IEEE Access, 6, 1662–1669.*

[21] *Willmott, C. J. (1982). Some comments on the evaluation of model performance. Bulletin of the American Meteorological Society, 63(11), 1309–1313.*

[22] *Kim, S., & Kim, H. (2016). A new metric of absolute percentage error for intermittent demand forecasts. International Journal of Forecasting, 32(3), 669–679.*