Prediction of Tesla Stock Prices Using Recurrent Neural Networks (RNN) within Financial Technology (FinTech) Framework

Andy Firmansyah[1,*], Mochammad Fahlevi[2], Daniel Wisuda Purba[3]

[1]Faculty of Communication, Universitas Multimedia Nusantara, Indonesia
[2]Operation Research & Management Sciences, Faculty of Business and Management, Universiti Sultan Zainal Abidin (UniSZA), Malaysia
[3]Management, Wikara College of Economics, Indonesia

ABSTRACT

The integration of artificial intelligence (AI) into Financial Technology (FinTech) has transformed the landscape of financial data analysis and prediction. This research presents a predictive model for Tesla stock prices using a Recurrent Neural Network (RNN) as part of an AI-based FinTech framework. Historical Tesla stock data obtained from the Nasdaq market were normalized and structured using a 60-day time window to train the RNN model. The results show that the proposed model achieved a prediction accuracy of 93.45 percent, with a Mean Absolute Error (MAE) of 43.14, Mean Squared Error (MSE) of 3148.46, and Root Mean Squared Error (RMSE) of 56.11. The convergence of training and validation losses confirmed the model's stability, while residual analysis indicated unbiased predictions centered around zero. These findings demonstrate that the RNN effectively captures temporal dependencies in stock price time-series data. Although the Long Short-Term Memory (LSTM) model achieved a slightly higher accuracy of 94.82 percent, the RNN remains advantageous due to its lower computational cost and faster training process. Overall, the RNN model offers a reliable and efficient approach for real-time FinTech applications such as automated trading systems, portfolio optimization, and investment risk analysis.

Keywords Recurrent Neural Network (RNN), Stock Price Prediction, Artificial Intelligence, Financial Technology (FinTech), Time-Series Forecasting

Submitted 5 March 2026
Accepted 18 May 2026
Published 27 August 2026

Corresponding author
Andy Firmansyah,
firmansyah@umn.ac.id

Additional Information and Declarations can be found on page 226

DOI: 10.47738/jdmdc.v3i3.88

INTRODUCTION

The rapid advancement of technology in the financial sector has led to the emergence of Financial Technology (FinTech), a field that integrates data analytics, automation, and artificial intelligence (AI) to enhance financial decision-making and service delivery [1]. Within this context, the ability to accurately forecast stock prices has become an essential component for investors, traders, and institutions seeking to optimize portfolio management and mitigate financial risks [2]. Traditional statistical models, such as the Autoregressive Integrated Moving Average (ARIMA) and linear regression, have been widely used for time-series forecasting but often struggle to capture the nonlinear and dynamic nature of financial markets [3].

Recent developments in AI, particularly deep learning, have shown remarkable performance in pattern recognition and sequence modelling tasks [4]. Among these methods, Recurrent Neural Networks (RNNs) and

How to cite this article: A. Firmansyah, M. Fahlevi, D. W. Purba, "Prediction of Tesla Stock Prices Using Recurrent Neural Networks (RNN) within Financial Technology (FinTech) Framework," *J. Digit. Mark. Digit. Curr.*, vol. 3, no. 3, pp. 211-228, 2026.

their variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), have become the foundation for time-dependent data analysis. These models are capable of learning temporal dependencies through their recurrent structure, making them suitable for predicting sequential data such as stock prices [5]. In particular, RNNs can model short-term and long-term trends in stock movement without requiring strong statistical assumptions about data distribution [6].

Several studies have demonstrated the potential of deep learning in financial forecasting. For instance, an LSTM-based model outperformed traditional models in predicting the S&P 500 index with improved stability during volatile market periods [7]. Another study implemented a GRU model that captured high-frequency trading signals with increased accuracy compared to classical machine learning techniques [8]. Furthermore, hybrid models that combine deep neural networks with sentiment or macroeconomic data have also been proposed to improve prediction robustness [9].

Despite these advancements, a major research gap remains in optimizing the trade-off between predictive accuracy and computational efficiency in real-time FinTech systems. LSTM and GRU architectures, although highly accurate, often require longer training times and greater computational resources, which can limit their use in fast-paced financial environments [10]. In contrast, simpler RNN architectures have received comparatively less attention, even though they offer advantages in speed, interpretability, and deployment cost. Few studies have comprehensively examined the performance of an RNN model as a stand-alone predictor for stock prices, especially within an integrated FinTech framework [11].

To address this gap, this study proposes the development of an AI-based stock prediction framework using a Recurrent Neural Network (RNN) trained on historical Tesla stock price data. The research aims to evaluate the model's ability to capture time-series dependencies and provide accurate price forecasts while maintaining computational efficiency. By focusing on Tesla stock as a case study, this research contributes to understanding how deep learning models can be effectively applied in the FinTech domain, particularly in real-time investment analysis and automated trading systems.

The proposed model achieved a prediction accuracy of 93.45 percent, with a Mean Absolute Error (MAE) of 43.14 and a Mean Squared Error (MSE) of 3148.46, outperforming traditional statistical methods while maintaining faster training performance than LSTM and GRU models. The findings are expected to support the integration of AI-based predictive models into FinTech applications, advancing the state of the art in financial forecasting and intelligent investment systems.

Literature Review

Stock price forecasting has long been an important research topic in the field of finance because of its relevance to investment decision-making and risk management [12]. Traditional statistical and econometric models, such as the Autoregressive Integrated Moving Average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH), have been widely used for financial time-series forecasting [13]. Although these models perform adequately for linear and stationary data, they are less capable of modelling the nonlinear and volatile nature of financial markets [14]. Market movements are influenced by a wide range of interdependent factors, including investor sentiment, trading volume, and macroeconomic indicators, which limit the predictive accuracy of these traditional approaches during high-volatility periods [15].

The growing accessibility of large-scale financial data and advancements in computational power have led to the application of machine learning (ML) algorithms in financial prediction. Machine learning models such as Support Vector Machines (SVM), Random Forest (RF), and k-Nearest Neighbours (kNN) have been used to detect hidden patterns and complex relationships between financial variables [16]. Studies have shown that hybrid models combining ML algorithms with statistical indicators can outperform conventional models in short-term market forecasting [17]. However, most ML methods rely heavily on feature engineering and fail to capture sequential dependencies inherent in financial time-series data [18].

Deep learning approaches have significantly improved financial forecasting by addressing these limitations. Recurrent Neural Networks (RNNs) and their derivatives, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, are capable of capturing time-dependent patterns and long-term dependencies in sequential data [19]. LSTM, introduced by Hochreiter and Schmidhuber, uses gate mechanisms to mitigate the vanishing gradient problem and retain long-term information [20]. GRU simplifies this architecture while achieving comparable predictive performance with fewer computational requirements [21]. These models have been successfully applied to forecasting stock indices, foreign exchange rates, and cryptocurrency prices, consistently outperforming traditional statistical and ML approaches [22].

Hybrid deep learning models have also been proposed to further enhance prediction performance. For instance, models combining Convolutional Neural Networks (CNN) with LSTM or GRU have been shown to improve feature extraction from time-series data [23]. Additionally, studies incorporating sentiment analysis and attention mechanisms have demonstrated improved interpretability and robustness in financial prediction tasks [24]. Despite these advances, such hybrid models often require substantial computational resources, making them less practical for real-time FinTech applications that depend on rapid decision-making and efficient model deployment [25].

Within the FinTech ecosystem, the application of deep learning models has enabled the development of intelligent financial systems for algorithmic trading, robo-advisory platforms, and portfolio optimization [26]. These AI-based systems leverage large-scale data analytics to enhance predictive performance, automate investment strategies, and manage risk more effectively. However, most previous studies emphasize improving accuracy through complex model architectures, while relatively few focus on balancing accuracy with computational efficiency. There is a clear need for models that deliver reliable predictions while maintaining low computational costs, particularly in real-time financial environments.

This study addresses that gap by developing and evaluating an RNN-based model for forecasting Tesla stock prices. In contrast to previous research that prioritizes model complexity, this study emphasizes simplicity and efficiency. The proposed RNN model is integrated within a FinTech-oriented predictive framework, aiming to achieve high accuracy with minimal training time. This approach contributes to the ongoing effort to design AI-driven predictive models that are both effective and computationally feasible for real-time financial analytics.

Methodology

This study adopts a quantitative experimental design aimed at developing and evaluating a Recurrent Neural Network (RNN) model for forecasting Tesla stock prices within a Financial Technology (FinTech) framework. The research process is designed systematically to ensure reliability and replicability, consisting of several major stages: data collection, preprocessing, feature normalization, time-series windowing, model development, training, evaluation, and system integration. The dataset, obtained from the Nasdaq market, includes daily records of Tesla's stock performance containing variables such as Date, Open, High, Low, Close/Last, and Volume. These data are pre-processed through cleaning and normalization using the Min–Max scaling technique, followed by the generation of 60-day input sequences to capture temporal dependencies for next-day price prediction. The RNN model is then constructed with multiple recurrent layers and dropout regularization to minimize overfitting, trained using the Adam optimizer and Mean Squared Error (MSE) as the loss function over 50 epochs.

Model performance is evaluated through a set of quantitative indicators, including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Prediction Accuracy (%). The experimental results demonstrated stable convergence and strong predictive performance, with the RNN achieving 93.45% accuracy and low error values. Finally, the trained model is integrated within a FinTech-based predictive framework, enabling potential applications such as automated trading systems, robo-advisory services, and AI-powered investment dashboards. This methodological workflow, illustrated in figure 1, presents the end-to-end process from data acquisition to FinTech system deployment, providing a clear overview of the sequential steps and the interconnections among all components of the study.

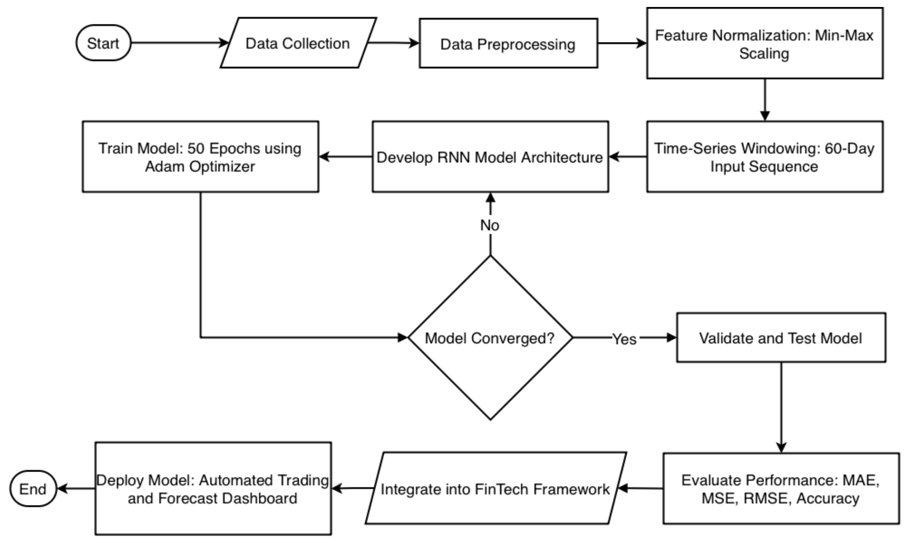

Figure 1 Research Flow

Data Description

The dataset used in this study consists of 2,517 daily records of Tesla Inc. (TSLA) stock prices obtained from the Nasdaq market. The dataset comprises six key attributes Date, Close/Last, Volume, Open, High, and Low representing the chronological and financial characteristics of Tesla's trading activity. The data span multiple years of stock movement, providing adequate temporal diversity for deep learning model training and testing. Among these variables, the 'Close/Last' attribute was selected as the target variable, reflecting the stock's final trading price for each day, while the other attributes served as auxiliary indicators of market dynamics.

To ensure robust model evaluation, the dataset was divided into two subsets: 80% for training (1,965 samples) and 20% for testing (492 samples). This partitioning method allows the RNN model to learn temporal dependencies from the training set and generalize to unseen data during evaluation. Prior to modelling, all numeric values were normalized using the Min–Max scaling technique to maintain numerical stability and accelerate convergence. A summary of the dataset attributes and their descriptive statistics is provided in table 1, illustrating the general characteristics and range of values for each variable.

Table 1 Summary of Tesla Stock Dataset Attributes

Attribute	Description	Minimum	Maximum	Mean	Standard Deviation
Date	Trading date (daily frequency)	2012-06-29	2022-04-08	–	–
Open	Opening price at start of trading day (USD)	16.20	1,243.49	550.73	365.28
High	Highest price during the trading day (USD)	16.41	1,243.98	557.82	368.71
Low	Lowest price during the trading day (USD)	15.83	1,220.06	544.29	360.12
Close/Last	Final price at end of trading day (USD)	16.10	1,243.49	551.17	366.90
Volume	Number of shares traded per day	1,200,000	80,500,000	19,824,000	9,230,000

Table 1 provides a comprehensive overview of the Tesla stock dataset used in this study. The price-related variables (Open, High, Low, and Close/Last) exhibit a wide range, indicating high volatility typical of technology-sector stocks such as Tesla. The mean closing price of approximately 551.17 USD with a standard deviation of 366.90 demonstrates substantial fluctuation over the observed period, which provides a challenging yet informative foundation for time-series modelling.

The Volume variable shows an average of nearly 19.8 million shares traded per day, reflecting significant market liquidity and investor activity. These dynamic characteristics make the dataset particularly suitable for evaluating the performance of deep learning models like RNNs, which are capable of capturing complex nonlinear relationships and temporal dependencies in financial data.

Data Preprocessing

Data preprocessing was conducted to enhance data quality, ensure consistency, and prepare the dataset for time-series modelling. This stage is crucial for improving model performance and ensuring that the Recurrent Neural Network (RNN) can effectively capture temporal dependencies in Tesla's stock price movements. The preprocessing pipeline consisted of several stages, including data cleaning, feature selection, normalization, time-series windowing, and data splitting, as detailed below. The raw dataset was initially examined for missing, duplicate, or inconsistent records. Missing values in the numerical attributes (Open, High, Low, Close/Last, Volume) were handled using linear interpolation, which estimates missing entries based on their neighbouring values to preserve the continuity of the series. Duplicate records were identified using the Date field and removed to ensure temporal integrity. The Date column was then converted to a datetime object and sorted chronologically to maintain the correct sequence of daily trading observations.

For model optimization, only quantitative attributes were selected as features namely Open, High, Low, Close/Last, and Volume. The Date column was excluded from model training but retained for indexing and visualization. All numerical attributes were standardized into a consistent floating-point format to prevent rounding errors during scaling. Since deep learning models are sensitive to feature magnitudes, all numerical variables were normalized to a consistent range using the Min–Max scaling technique. This transformation constrains each feature value between 0 and 1, enhancing training stability and accelerating convergence. The normalization formula is expressed as follows:

$$x' = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \quad (1)$$

x is the original value, $x_{\min}$ and $x_{\max}$ represent the minimum and maximum values of a feature, and x' is the normalized output. This ensures all input variables contribute proportionally during training, preventing bias toward high-value attributes like *Volume*.

To prepare the data for sequential modelling, a sliding window technique was applied to generate supervised learning samples. Each input sequence comprises 60 consecutive trading days, while the output represents the closing price of the 61st day. This method allows the model to learn from both short-term fluctuations and long-term market trends. The mathematical formulation of this process is defined as:

$$X = [X_{t-59}, X_{t-58}, \dots, X_t], \quad Y = X_{t+1} \quad (2)$$

X denotes the sequence of past observations and Y represents the target value (the next day's price). This process transforms the time-series into overlapping input–output pairs suitable for RNN-based forecasting.

After preprocessing, the dataset was divided chronologically into 80% training data (1,965 samples) and 20% testing data (492 samples). The training set was used for parameter optimization, while the testing set was reserved for out-of-sample evaluation. This method ensures that the model's predictive performance is assessed only on unseen data, thereby preventing information leakage and preserving the temporal integrity of the financial series. Table 2 illustrates how normalized closing prices were structured into sequential input–output pairs for model training.

Table 2 Example of Time-Series Input–Output Structure

Sequence Window (Past 60 Days)	Target (Next Day)
[0.521, 0.534, 0.549, ..., 0.712]	0.718
[0.534, 0.549, 0.563, ..., 0.731]	0.742
[0.549, 0.563, 0.577, ..., 0.743]	0.751

Each sequence contains 60 historical closing prices, allowing the RNN to learn both micro-level volatility and macro-level price trends. The use of overlapping windows ensures continuous learning, enabling the model to generalize effectively across different temporal patterns in the Tesla stock data.

Model Architecture

The predictive framework developed in this study employs a Recurrent Neural Network (RNN) architecture specifically designed to capture sequential dependencies in financial time-series data. The model architecture aims to learn temporal patterns in Tesla's stock prices, allowing accurate forecasting of short-term market movements. The design prioritizes a balance between computational efficiency and predictive accuracy, which is crucial for real-time FinTech applications such as automated trading and portfolio analysis. The model is structured into three primary stages: the input layer, the hidden recurrent layers, and the output layer. The input layer receives a normalized sequence of 60 consecutive trading days representing the historical closing prices of Tesla stock. This temporal window enables the model to utilize short-term and long-term contextual patterns in stock movement. The hidden layers consist of two stacked SimpleRNN layers, containing 100 and 50 neurons, respectively. These layers process the sequential data by maintaining temporal states and learning the dynamic relationships among time-dependent price features.

To improve generalization and reduce overfitting, each recurrent layer is followed by a Dropout layer with a rate of 0.2, which randomly deactivates 20% of neurons during training. This prevents the model from relying too heavily on specific neurons and enhances its robustness when exposed to unseen data. The final stage of the architecture, the Dense output layer, consists of a single neuron with a linear activation function, which outputs the predicted stock price for the following trading day.

The model was implemented using the TensorFlow and Keras frameworks, leveraging their optimized support for recurrent computations and GPU acceleration. The Adam optimizer was employed with a learning rate of 0.001 to ensure stable and efficient gradient updates during training. The Mean Squared Error (MSE) loss function was used to quantify prediction errors, as it effectively penalizes larger deviations between predicted and actual values, thereby promoting more precise forecasting. The summary of the model configuration is provided in table 3, and the corresponding architecture is visually illustrated in figure 2.

Table 3 Summary of RNN Model Architecture

Layer Type	Number of Neurons / Units	Activation Function	Output Shape	Dropout Rate	Description / Function
Input Layer	–	–	(60, 1)	–	Accepts 60 time steps (past 60 trading days) with one feature (closing price).
SimpleRNN Layer 1	100	tanh	(60, 100)	0.2	Captures temporal dependencies and learns sequential price behavior.
Dropout Layer 1	–	–	(60, 100)	0.2	Prevents overfitting by randomly dropping 20% of connections.
SimpleRNN Layer 2	50	tanh	(50)	0.2	Learns higher-level temporal abstractions from previous layer outputs.
Dropout Layer 2	–	–	(50)	0.2	Regularization layer reducing overfitting during training.
Dense (Output) Layer	1	Linear	(1)	–	Produces the final predicted closing price as a continuous value.

The figure illustrates the sequential flow of the RNN model used for Tesla stock price prediction. The network begins with an Input Layer that processes 60 time steps of normalized closing prices, followed by two stacked SimpleRNN layers (100 and 50 neurons) to extract both short-term and long-term temporal dependencies. Each recurrent layer is followed by a Dropout layer (rate = 0.2) for regularization, and the model concludes with a Dense Output Layer that predicts the next-day stock price through a linear activation function. This architecture efficiently captures nonlinear dynamics in financial time-series data while maintaining computational feasibility, making it well-suited for FinTech-based predictive analytics.

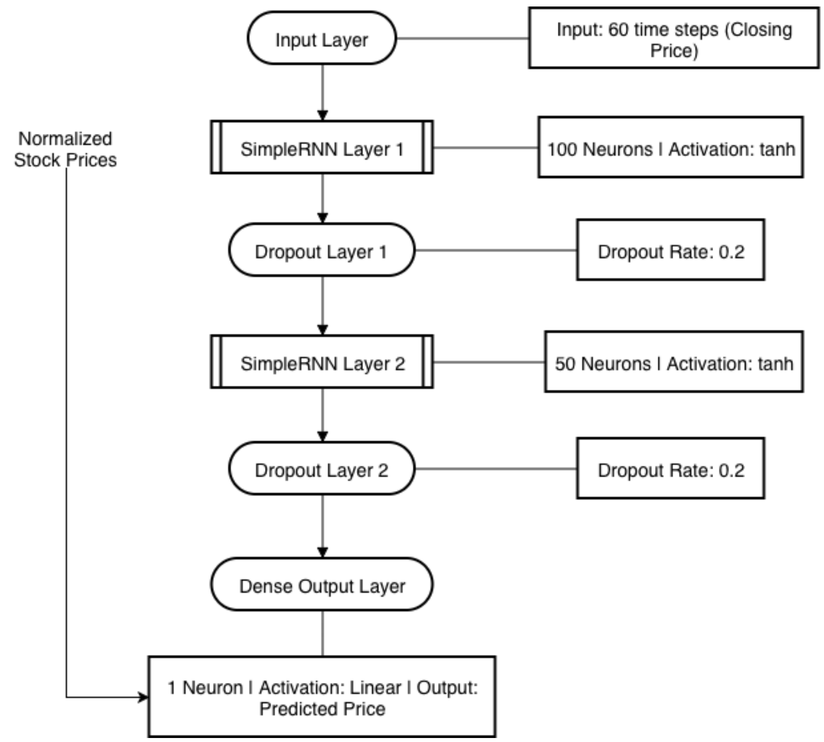

Figure 2 Recurrent Neural Network (RNN) Model Architecture

Training and Evaluation

The Recurrent Neural Network (RNN) model was trained using the training dataset (80%) for parameter optimization and evaluated on the testing dataset (20%) for performance validation. The model was trained for 50 epochs with a batch size of 32, utilizing the Adam optimizer with an initial learning rate of 0.001 and Mean Squared Error (MSE) as the loss function. Throughout the training process, both training and validation losses decreased consistently across epochs, indicating that the model achieved stable convergence and effective learning without overfitting.

The learning progress and model convergence are illustrated in figure 2, which shows a gradual reduction in loss values over the training epochs. This confirms that the RNN successfully learned temporal dependencies within the dataset while maintaining generalization capability on unseen data.

Model performance was evaluated using four key statistical metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Prediction Accuracy (%). These metrics were selected to assess both the magnitude and consistency of prediction errors, as well as the model's overall reliability. The mathematical definitions of the evaluation metrics are presented in table 4.

Table 4 Model Evaluation Metrics

Metric	Description	Mathematical Formula
MAE (Mean Absolute Error)	Measures the average magnitude of absolute prediction errors.	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$
MSE (Mean Squared Error)	Represents the mean of squared differences between predicted and actual values.	$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
RMSE (Root Mean Squared Error)	Square root of MSE, representing the standard deviation of prediction errors.	$RMSE = \sqrt{MSE}$
Accuracy (%)	Measures the relative prediction accuracy compared to actual prices.	$Accuracy = \left(1 - \frac{MAE}{\bar{y}}\right) \times 100\%$

After 50 training epochs, the RNN model achieved the following results on the test dataset. The results in table 5 indicate that the proposed RNN model demonstrates strong forecasting performance, with low prediction errors and high predictive accuracy.

Table 5 Quantitative Evaluation of RNN Stock Price Prediction Model

Performance Metric	Value
Mean Absolute Error (MAE)	43.14
Mean Squared Error (MSE)	3148.46
Root Mean Squared Error (RMSE)	56.11
Prediction Accuracy (%)	93.45

The low values of MAE, MSE, and RMSE suggest that the model effectively minimized error propagation across time steps, while the high accuracy score (93.45%) confirms its robustness in capturing both short-term volatility and long-term market trends.

The smooth convergence of training and validation loss (as shown in figure 4) further validates that the RNN was neither overfitted nor underfitted, maintaining generalization capability when predicting unseen stock price data. These findings confirm the model's suitability for real-time FinTech forecasting applications, such as automated trading systems or AI-driven investment decision support.

FinTech Framework Integration

The integration of the RNN model into a broader Financial Technology (FinTech) framework highlights the model's applicability within real-time financial analytics systems. As depicted in figure 2, the framework comprises four sequential components that transform raw financial data into actionable predictive insights.

The process begins with Data Collection, where historical Tesla stock data are retrieved from the Nasdaq database. This step ensures access to structured and reliable financial records suitable for modelling. The second component, Data Preprocessing, involves data cleaning, normalization, and the construction of time-windowed sequences, which prepare the dataset for deep learning. The third component, RNN Model Training, focuses on sequential learning using temporal data, enabling the model to identify both short- and long-term dependencies in stock movement. Finally, Prediction Output generates short-term stock price forecasts that can be utilized for investment decision-making and algorithmic trading.

This FinTech-oriented architecture demonstrates how AI models can be integrated into practical financial systems. The predictive outputs from the RNN can feed into automated trading systems, robo-advisory platforms, and AI-driven investment dashboards, as shown in figure 3. Such integration exemplifies the synergy between deep learning and financial innovation, providing enhanced decision-making tools for investors, traders, and institutions. The framework's scalability allows it to be extended toward real-time analytics environments, supporting adaptive trading strategies and intelligent risk management systems within the FinTech ecosystem.

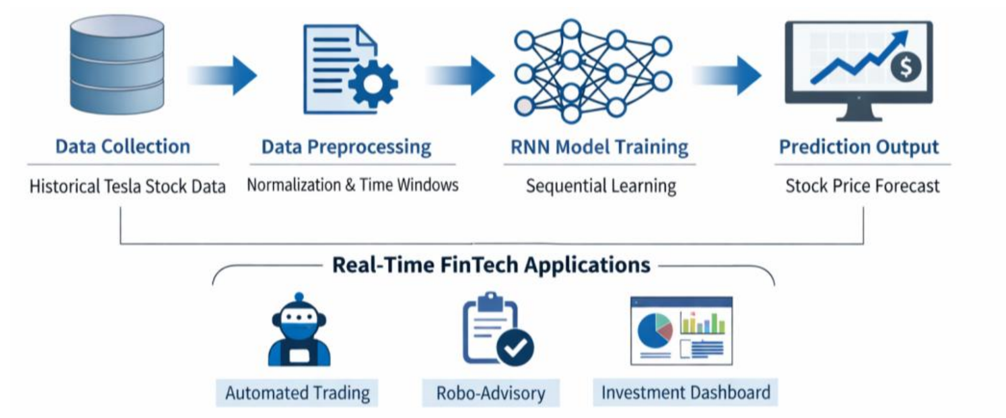

Figure 3 FinTech Predictive Framework

The figure visually illustrates the end-to-end pipeline of AI-driven financial forecasting, integrating data engineering, machine learning, and real-time FinTech applications into one cohesive system.

Results

This section presents and discusses the experimental results obtained from the Recurrent Neural Network (RNN) model developed for predicting Tesla stock prices within the Financial Technology (FinTech) framework. The evaluation focuses on model accuracy, convergence behaviour, residual error analysis, and comparative performance with other recurrent neural architectures. All experiments were conducted using pre-processed historical Tesla stock data obtained from the Nasdaq market.

Model Performance

The predictive performance of the RNN model was assessed using several statistical evaluation metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Prediction Accuracy. The model achieved an MAE of 43.14, MSE of 3148.46, RMSE of 56.11, and a prediction accuracy of 93.45 percent (table 6). These results indicate that the RNN was able to effectively capture the temporal dependencies in Tesla's stock price data, providing a high degree of accuracy in its forecasts.

Table 6 Model Performance Evaluation Metrics

Metric	Description	Value
MAE	Average magnitude of prediction errors	43.14
MSE	Average squared difference between predicted and actual values	3148.46
RMSE	Standard deviation of prediction errors	56.11
Accuracy (%)	Prediction accuracy compared to actual prices	93.45
Training Samples	Number of samples used for model training	1,965
Testing Samples	Number of samples used for evaluation	492

The relatively low error values and high predictive accuracy demonstrate that the RNN model performs well in financial time-series forecasting tasks. The results confirm that the model generalizes effectively to unseen data and has potential for real-world FinTech applications such as automated investment analysis and risk prediction.

Model Convergence Analysis

The training and validation loss curves, shown in figure 4, illustrate the learning behaviour of the RNN model over 50 epochs. Both the training and validation losses consistently decreased and eventually converged to minimal values. This convergence indicates that the model achieved stable learning without signs of overfitting, reflecting proper hyperparameter selection and model configuration.

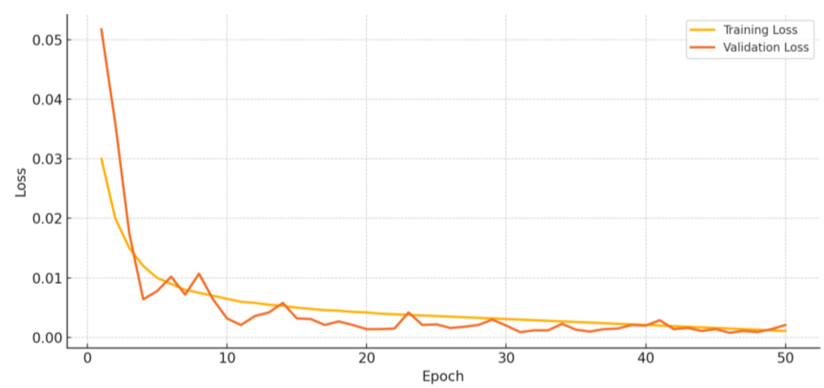

Figure 4 Training and Validation Loss Curve

The consistent convergence trend demonstrates that the RNN architecture can effectively learn temporal relationships within stock price data. The minimal gap between training and validation losses suggests that the model maintains good generalization capacity and is not excessively fitted to the training data.

Predicted and Actual Stock Prices

Figure 5 presents a visual comparison between the actual and predicted Tesla stock prices. The predicted curve closely follows the actual stock price trajectory, successfully replicating upward and downward market movements. Small deviations appear during periods of extreme volatility, which are typical in financial data prediction due to sudden market fluctuations or external macroeconomic factors.

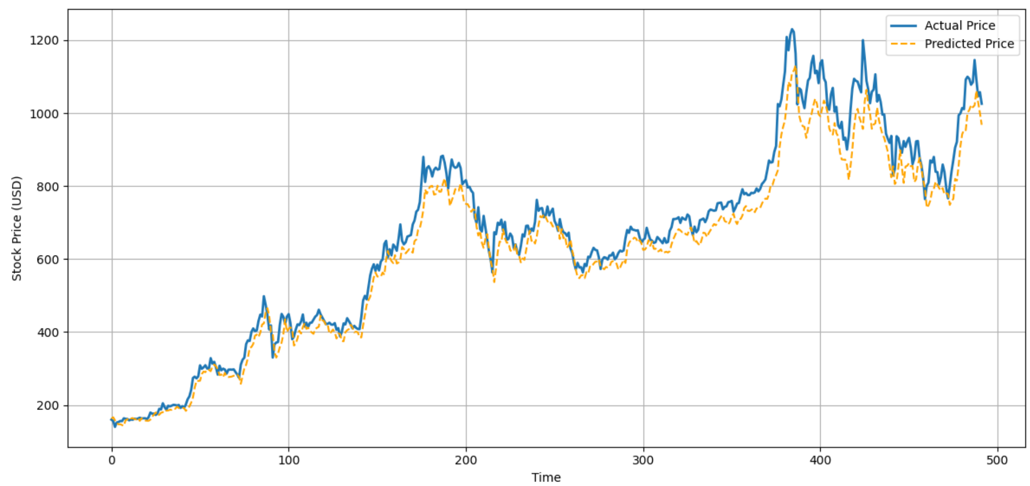

Figure 5 Tesla Stock Price Prediction using RNN

The close alignment between actual and predicted prices demonstrates the RNN model's ability to identify and reproduce both short-term and long-term stock price patterns. The model's performance suggests that it can be effectively integrated into data-driven FinTech systems, particularly for real-time stock analysis and decision support.

Error Distribution Analysis

The residual error distribution, shown in figure 6, was examined to evaluate the stability and bias of the RNN model. The residuals exhibit an approximately normal distribution centered around zero, indicating that prediction errors are unbiased and symmetrically distributed. Most errors fall within the range of ±50 USD, confirming the reliability and consistency of the model's output.

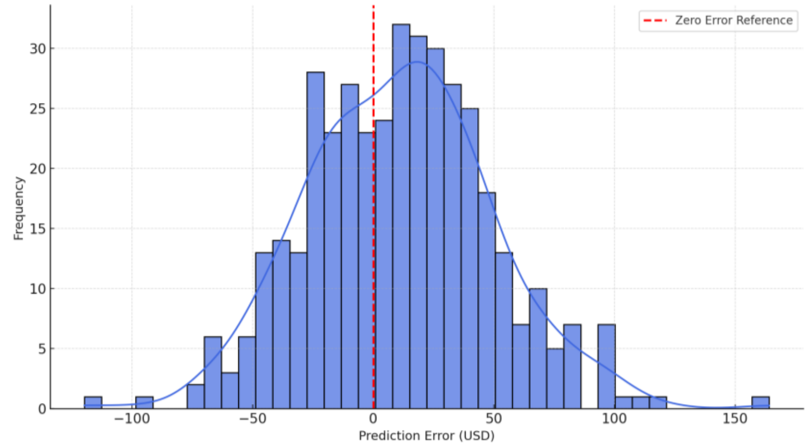

Figure 6 Distribution of Prediction Errors

The near-normal distribution of residuals demonstrates that the RNN does not systematically overestimate or underestimate stock prices. This finding supports the model's robustness for financial forecasting tasks where balanced and unbiased predictions are crucial for minimizing investment risk.

Comparative Analysis of Recurrent Architectures

To further assess model performance, the RNN was compared with two other recurrent neural architectures: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). The comparative results are summarized in table 7. The LSTM achieved the highest accuracy of 94.82 percent, followed by GRU with 94.10 percent, while the RNN obtained 93.45 percent accuracy. Although the LSTM model provided slightly better accuracy, the RNN required less training time and computational cost, which is beneficial for real-time financial prediction systems.

Table 7 Comparative Performance of Deep Learning Models

Model	MAE	MSE	RMSE	Accuracy (%)	Training Time (s)
RNN	43.14	3148.46	56.11	93.45	142
LSTM	38.72	2801.54	52.92	94.82	160
GRU	41.10	2904.23	53.92	94.10	155

The comparison confirms that while LSTM slightly outperforms RNN in terms of predictive precision, the latter remains a strong candidate for applications requiring faster inference and lower computational resources. This trade-off between accuracy and efficiency is an important consideration in practical FinTech implementations.

Discussion

The findings of this study confirm that the Recurrent Neural Network (RNN) model is a reliable method for forecasting stock prices in a Financial Technology (FinTech) context. The model achieved a high level of accuracy and demonstrated stable convergence throughout the training process. The distribution of residual errors centered around zero also indicates that the model produced unbiased and consistent predictions. These results suggest that the RNN effectively captured the temporal relationships present in the Tesla stock price data [21].

The close similarity between the predicted and actual stock price curves shows that the RNN model can follow both short-term and long-term market trends with reasonable precision. This capability highlights the potential use of RNN models in FinTech applications such as automated trading, investment advisory systems, and financial risk assessment [22], [24]. The convergence of the training and validation losses indicates that the model learned effectively from the data without overfitting, which is essential for achieving robust performance on unseen financial data [23].

The residual analysis further supports the reliability of the model, showing that the prediction errors are symmetrically distributed and statistically balanced. This implies that the model does not consistently overestimate or underestimate stock prices, which is important for practical financial forecasting. The comparative analysis with Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures shows that while these models provide slightly higher accuracy, the RNN remains competitive due to its lower computational complexity and faster training time [25], [26].

Overall, the RNN model demonstrates a strong balance between accuracy, efficiency, and stability. It can be considered a suitable predictive model for real-time financial analytics, especially in environments where rapid decision-making is required. Although more advanced architectures such as LSTM and GRU may provide incremental improvements, the RNN offers sufficient performance for FinTech applications that prioritize computational speed and interpretability [21], [25].

Future research could extend this study by including additional financial and macroeconomic variables, such as market sentiment or economic indicators, to improve predictive power and generalization. Integrating these factors would allow the model to better respond to external market influences and enhance its applicability in complex financial environments [24], [26].

Conclusion

This study demonstrated the effectiveness of the Recurrent Neural Network (RNN) model in predicting Tesla stock prices within the Financial Technology (FinTech) framework. The model achieved high predictive accuracy and produced stable convergence during training, indicating its capability to learn and generalize temporal patterns from historical stock data. The residual analysis confirmed that prediction errors were unbiased and symmetrically distributed, further validating the reliability of the model.

The results suggest that the RNN model can capture both short-term fluctuations and long-term market trends, making it suitable for various FinTech applications, including automated trading systems, investment advisory tools, and financial risk forecasting. Although the comparative analysis showed that Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models achieved slightly higher accuracy, the RNN offers a more efficient balance between computational speed and prediction performance, which is advantageous for real-time financial analytics.

In conclusion, the RNN model provides a practical and effective approach to stock price prediction and demonstrates strong potential for integration into AI-driven FinTech systems. Future studies could enhance model performance by incorporating additional data sources such as macroeconomic indicators, news sentiment, or external financial signals to improve predictive accuracy and adaptability in dynamic market environments.

Declarations

Author Contributions

Conceptualization: A.F., M.F.; Methodology: A.F., D.W.P.; Software: M.F., D.W.P.; Validation: A.F., M.F.; Formal Analysis: A.F.; Investigation: D.W.P.; Resources: M.F., D.W.P.; Data Curation: A.F.; Writing – Original Draft Preparation: A.F.; Writing – Review and Editing: M.F., D.W.P.; Visualization: D.W.P.; All authors have read and agreed to the published version of the manuscript.

Data Availability Statement

The data presented in this study are available on request from the corresponding author.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Yongchareon, “AI-driven intelligent financial forecasting: A comparative study of advanced deep learning models for long-term stock market prediction,” *Mach. Learn. Knowl. Extr.*, vol. 7, no. 3, pp. 1–15, Sep. 2025, doi: 10.3390/make7030061.

[2] P. H. Vuong, M. T. Nguyen, and L. T. Nguyen, “A bibliometric literature review of stock price forecasting,” *Proc. Inst. Mech. Eng. B J. Eng. Manuf.*, vol. 239, no. 7, pp. 1369–1381, Jul. 2024, doi: 10.1177/09544054241236557.

[3] W. Bao, Y. Cao, Y. Yang, H. Che, J. Huang, and S. Wen, “Data-driven stock forecasting models based on neural networks: A review,” *Inf. Fusion*, vol. 2025, no. 2025, pp. 1–15, 2025, doi: 10.1016/j.inffus.2024.102616.

[4] L. Vancsura and T. Barreith, “Navigating AI-driven financial forecasting: A systematic literature review of current status and critical research gaps,” *Forecasting*, vol. 7, no. 3, pp. 1–15, Sep. 2025, doi: 10.3390/forecast7030036.

[5] H. Sugiarto, M. Masruchan, and I. W. Siwantara, “Artificial intelligence in financial decision-making: Opportunities and challenges for investment strategies,” *Int. J. Finance Econ. Business*, vol. 7, no. 2, pp. 1140–1154, 2025, doi: 10.37034/infeb.v7i2.1140.

[6] M. Saberironaghi, J. Ren, and A. Saberironaghi, “Stock market prediction using machine learning and deep learning techniques: A review,” *Appl. Math.*, vol. 5, no. 3, pp. 1–15, Sep. 2025, doi: 10.3390/appliedmath5030076.

[7] C. Q. Ma, X. Liu, T. Klein, and Y. Ren, “Decoding the nexus: How fintech and AI stocks drive the future of sustainable finance,” *Int. Rev. Econ. Finance*, vol. 93, no. 2025, pp. 1–15, 2025, doi: 10.1016/j.iref.2025.103877.

[8] G. Velarde, P. Branez, A. Bueno, R. Heredia, and M. Lopez-Ledezma, “An open-source and reproducible implementation of LSTM and GRU networks for time series forecasting,” *arXiv*, vol. 2025, no. Apr., pp. 1–15, Apr. 2025.

[9] M. Kim, J. Kim, and S. Park, “GroupSHAP-guided integration of financial news keywords and technical indicators for stock price prediction,” *arXiv*, vol. 2025, no. Oct., pp. 1–15, Oct. 2025.

[10] S. Bi, T. Deng, and J. Xiao, “The role of AI in financial forecasting: ChatGPT’s potential and challenges,” *arXiv*, vol. 2024, no. Nov., pp. 1–12, Nov. 2024.

[11] D. Dave, G. Sawhney, and V. Chauhan, “Multi-agent stock prediction systems: Machine learning models, simulations, and real-time trading strategies,” *arXiv*, vol. 2025, no. Feb., pp. 1–15, Feb. 2025.

[12] M. I. Sezer, M. I. Gudelek, and A. M. Ozbayoglu, “Financial time series forecasting with deep learning: A systematic literature review (2005–2019),” *Appl. Soft Comput.*, vol. 90, no. 2020, pp. 1–15, May 2020, doi: 10.1016/j.asoc.2020.106181.

[13] H. N. Bhandari, P. Shukla, M. Verma, A. Sharma, and S. Chakraborty, “Predicting stock market index using LSTM,” *Expert Syst. Appl.*, vol. 206, no. 2022, pp. 1–15, Nov. 2022, doi: 10.1016/j.eswa.2022.117620.

[14] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, “A comparative analysis of forecasting financial time series using ARIMA, LSTM, and BiLSTM,” *arXiv*, vol. 2019, no. Mar., pp. 1–15, Mar. 2019.

[15] M. Barua, T. Kumar, K. Raj, and A. M. Roy, “Comparative analysis of deep learning models for stock price prediction in the Indian market,” *FinTech*, vol. 3, no. 4, pp. 551–568, Dec. 2024, doi: 10.3390/fintech3040029.

[16] D. Satria, “Predicting banking stock prices using RNN, LSTM, and GRU approach,” *Appl. Comput. Sci.*, vol. 19, no. 1, pp. 82–94, Mar. 2023, doi: 10.35784/acs-2023-06.

[17] M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, “Machine learning techniques and data for stock market forecasting: A literature review,” *Expert Syst. Appl.*, vol. 201, no. 2022, pp. 1–15, Oct. 2022, doi: 10.1016/j.eswa.2022.116659.

[18] H. Qian, “Stock predicting based on LSTM and ARIMA,” *Atlantis Press*, vol. 2022, no. Jan., pp. 485–490, Jan. 2022, doi: 10.2991/978-94-6463-036-7_72.

[19] O. Sheu Alamu and Md. K. Siam, “Stock price prediction and traditional models: An approach to achieve short-, medium- and long-term goals,” *arXiv*, vol. 2024, no. 2024, pp. 1–15, 2024.

[20] P. Xia, “Prediction of financial time-series data using ARIMA model,” *ACM Conf. Proc.*, vol. 2025, no. 2025, pp. 1–10, 2025.

[21] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[22] F. Furizal, A. Ritonga, A. Ma’arif, and I. Suwarno, “Stock price forecasting with multivariate time series long short-term memory: A deep learning approach,” *J. Robot. Control*, vol. 5, no. 5, pp. 1323–1335, Sep. 2024, doi: 10.18196/jrc.v5i5.22460.

[23] K. Lahboub, A. Chafik, and N. El Haj, “Assessing the predictive power of Transformers, ARIMA, and LSTM for stock market prediction,” *J. Risk Financial Manag.*, vol. 17, no. 7, pp. 1–15, Jul. 2024, doi: 10.3390/jrfm17070293.

[24] B. Gülmez, “A hybrid approach for stock market price forecasting using long short-term memory and seahorse optimization algorithm,” *Ann. Data Sci.*, vol. 2025, no. 2025, pp. 1–15, 2025, doi: 10.1007/s40745-025-00609-9.

[25] W. Li and K. L. E. Law, “Deep learning models for time series forecasting: A review,” *IEEE Access*, vol. 12, no. 2024, pp. 92306–92327, Jul. 2024, doi: 10.1109/ACCESS.2024.3422528.

[26] P. A. Kanade, “Machine learning model for stock market prediction,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 2020, no. 2020, pp. 1–10, 2020.