Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 6 (2025)

Recurrent Neural Networks in Time-Series Forecasting: A Deep Learning Approach to Stock Market Prediction

[1]Hadi Abdullah, [2]Aamna Tariq, [3]Ijaz khan, [4]Rizwan Iqbal, [5]Faisal Khan, [6]Arshad iqbal

Article Details

Keywords: Recurrent Neural Networks, Long Short-Term Memory, Gated Recurrent Unit, Stock Market Forecasting, Deep Learning, Time-Series Prediction, Financial Markets, Algorithmic Trading, Directional Accuracy, Predictive Modeling

Hadi Abdullah
Faculty of Computer Science, Lahore Garrison University
Hadi.uthm@yahoo.com

Aamna Tariq
M.Phil in Statistics, Government College University, Lahore & Data analyst, Mindrind, Johar Town Q Block, Lahore
aamnatariq55@gmail.com

Ijaz khan
Department of Avionics Engineering, College of Aeronautical Engineering (CAE) National University of sciences and Technology (NUST)
ikhan@cae.nust.edu.pk

Rizwan Iqbal
Department of Telecommunication Engineering, Dawood University of Engineering and Technology, Karachi
rizwan.iqbal@duet.edu.pk

Faisal Khan
Department of Telecommunication Engineering Dawood University of Engineering and Technology, Karachi
faisal.khan@duet.edu.pk

Arshad iqbal
Department of Computer Science, Khushal Khan Khattak Univversity Karak
arshadktk.uop@gmail.com

ABSTRACT

Stock market prediction has been a grand challenge due to dynamic nature, non-linearity and volatility of the financial markets. Traditional statistical models have proved useful historically, but are less likely to successfully model the complex temporal dependencies in stock price data. In recent years there was a breakthrough in deep learning, namely Recurrent Neural Networks (RNNs), which opens up new opportunities in time-series forecasting. The purpose of the work is to investigate how three variations of RNN-based models, such as Simple RNN, Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), perform on the task of predicting the direction of the stock market using the historical data of the S&P 500 index. The models were trained and tested through a pure experimental design that factored in a 60-day look-back window, normalization, and sequence modeling across diverse performance measures that include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Directional Accuracy (DA). This is rather clear in the findings that the LSTM and GRU networks are significantly superior to the Simple RNN in predictive power and robustness to varying market environments. LSTM in particular generalized the most, over 81 percent of its predictions fell within 2 percent error margin and directional accuracy of 72.4 percent on the test set. In addition to enhancing the applicability of deep RNN architectures in financial prediction, the results also imply that they can be applicable to algorithmic trading and investment decisions systems. Future research directions might be observed in the sphere of multi-modal data source integration and model interpretability, which would allow to advance the domain of deep learning applicability in finance further.

INTRODUCTION

Stock market prediction has been considered as one of the most significant, yet difficult, issues in the area of financial engineering and data science due to the non-linear and highly non-stationary dynamics of the stock market when reacting to various exogenous influences (Atsalakis & Valavanis, 2009; Fama, 1965). Accurate forecasting of the stock price can be highly valuable, and the information can be incorporated into the enhancement of trading strategies and investment decision-making and risk management (Chen et al., 2015; Patel et al., 2015). However, the unpredictable nature of the financial markets dominated by such powerful factors as the psychology of investors, geopolitical developments, macroeconomic releases, etc., continues to pose a challenge to the successful implementation of the classic forecasting models (Box & Jenkins, 1976; Kim, 2003).

Among the first approaches to stock market forecasting were linear time-series models such as AutoRegressive Integrated Moving Average (ARIMA) and Exponential Smoothing techniques (Hyndman & Athanasopoulos, 2018; Nelson et al., 1999). Despite providing a foundation of the time-series analysis, the assumptions of stationarity and linearity are not typically satisfied in real-world stock price series, which are non-stationary and chaotic (Cont, 2001; Fama, 1965). Secondly, the models are poor in modelling long term temporal characteristics and non-linear interactions of market factors (Zhang et al., 1998).

Since the introduction of machine learning (ML) techniques, the financial forecasting issue underwent a paradigm shift (Huang et al., 2005; Tsai & Hsiao, 2010). The Support Vector Machines (SVM), Decision Trees and the ensemble methods such as the Random Forests showed to be superior in predictive accuracy when compared to the classic statistic based models (Atsalakis & Valavanis, 2009; Ince & Trafalis, 2006). However, their standard formulations usually treat financial time-series data as independent samples and, therefore, overlook the temporal dependencies that are most crucial in the context of learning the behavior of stock markets (Bao et al., 2017).

A sub-branch of machine learning Deep learning (DL) where neural networks have numerous layers has demonstrated unprecedented performance on a variety of sequential data tasks (LeCun et al., 2015; Goodfellow et al., 2016). Recurrent Neural Networks (RNNs) in particular are configured to learn sequential data, whereby they retain internal state, which summarises the information of the previous time steps (Rumelhart et al., 1986). It especially makes the RNNs extremely time-series prediction, in which the past values affect the future patterns (Lipton et al., 2015).

Despite the theoretical appeal of standard RNNs, three major problems exist with standard RNNs: vanishing and exploding gradients and learning long-term dependencies (Pascanu et al., 2013). In order to overcome these limitations, later architectures have been suggested, like the Long Short-Term Memory (LSTM) networks (Hochreiter & Schmidhuber, 1997) and Gated Recurrent Units (GRUs) (Cho et al., 2014). Presented LSTM networks have a notion of gating that helps them keep information in a long time horizon, yet GRU suggests a simplified structure and offers an equivalent outcome (Chung et al., 2014; Greff et al., 2017).

The more recent research in the domain showed the success of LSTM and GRU models in financial prediction. Just to give an example, Fischer and Krauss (2018) discovered that the LSTM networks severely outperformed the classical models and the feedforward neural networks in predicting the directional movement of the S&P 500 index. Likewise, Nelson et al. (2017) discovered that the LSTM-based models outperformed the traditional approaches in terms of accuracy and robustness in stock price forecasting. Besides that, Bao et al. (2017) introduce a hybrid deep learning model, on which wavelet transforms and LSTM networks are implemented to enhance the results on stock price prediction.

Furthermore, unlike the traditional approaches, the models with RNN can learn long-term dependencies and non-linear relations that are present in the stock market data (Sezer et al., 2020). Such modellability is especially practical because the cost of the stocks is driven not only by the long-term tendencies but also by the short-term oscillations (Zhang et al., 2020). Moreover, deep learning models can be substantially flexible, i.e., additional features, including technical indicators, sentiment analysis, or macroeconomic variables, among others, can be input, and the forecasting performance can be improved further (Li et al., 2019; Ghoshal & Roberts, 2018).

Yet, there are still some issues concerning the application of RNN to the stock market prediction. Overfitting is a problem, as there is very little trusted financial data, and stock prices have a large noise-to-signal ratio (Bukhari et al., 2020). Furthermore, the interpretability of deep learning models leaves much to be desired because the models are considered black-box, which means that the finance professionals would struggle to trust the models and use them in production (Guidotti et al., 2018). In addition, the financial markets are very dynamic, and thus demand models that are flexible enough to adjust to the new market conditions and unobserved data distributions (Xiong et al., 2015).

Based on these, the proposed research study would entail a systematic exploration of the application of the RNN-based networks i.e., LSTM and GRU networks in stock market prediction. We are going to attempt to gauge the quality of prediction of these models and investigate their chances of real-life applicability in financial analytics with the advantage of a well-designed experiment and a broad spectrum of evaluation criteria. The study gives the ever-increasing literature on deep learning in the financial foresight and offers practical implications to the researcher and practitioner in the area.

LITERATURE REVIEW

Stock market forecasting accurate stock market prediction has been an area of active interdisciplinary research, involving economics, statistics, computer science, and artificial intelligence (AI). Traditionally, most research on forecasting in the financial sphere has utilized classical statistical models, yet over the past few years, the development of deep learning (DL) has led to a paradigm shift in the process and potential of time-series forecasting to stock prediction (Ballings et al., 2015; Fischer et al., 2023).

Financial modeling has long been based on classic time-series forecasting models that include Autoregressive (AR), Moving Average (MA), ARIMA, and Vector AutoRegression (VAR) (Hamilton, 1994; Lutkepohl, 2005). One of the key assumptions made by these models is that of linearity and stationarity which cripples their potential in dealing with complex volatile non-linear dynamics of financial markets (Campbell et al., 1997; Harvey, 1990). Even though these models are computationally efficient and interpretable, they are not effective in terms of long-term dependencies and sudden regime shifts in stock price dynamics (Engle & Patton, 2001).

Trying to address the shortcomings of the linear models, scholars proposed nonlinear models like Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models and its extensions (Bollerslev, 1986; Glosten et al., 1993) which modeled volatility clustering better, a recognized characteristic in financial time series (Cont, 2001). Nevertheless, GARCH based models continue to miss the mark in terms of their capability to estimate complex temporal dependencies, which evolve on long time horizons.

The advent of machine learning (ML) made finite difference equations like Support Vector Machines (SVM) (Tay & Cao, 2001), k-Nearest Neighbors (NN) (Zhang & Zhou, 2004), and

Extreme Learning Machines (Huang et al., 2012) to perform better on financial forecasting. These techniques exploit their capacity to capture nonlinear relationships without a priori specification of the relationship among the variables (Atsalakis & Valavanis, 2009). However, they tend to overlook the sequence and temporal nature of stock market data since they are initialized on fixed feature vectors, rather than time-series sequences (Henrique et al., 2019).

This has changed with the introduction of deep learning, which has architectures specifically capable of capturing sequential dependencies and learning complex nonlinear features directly on raw data (Goodfellow et al., 2016; Zhang et al., 2021). Convolutional Neural Networks (CNNs), which were initially developed to operate on images, have also been used in stock market prediction after time series were considered as pseudo-images (Sezer & Ozbayoglu, 2018). However, regardless of the encouraging findings, CNNs inherently struggle to represent long-term spatial dependencies because of their receptive field (Huang et al., 2020). In comparison Recurrent Neural Networks (RNNs) are designed precisely to model sequential data. Elman (1990) proposed RNNs, with feedback connections that allow them to remember information across time steps, and therefore they can learn time-dependent patterns. Unfortunately, vanishing and exploding gradients were severe training issues of early RNNs, extremely constraining their potential to perform long-term dependency learning (Bengio et al., 1994).

This bottleneck was solved when Hochreiter & Schmidhuber (1997) suggested the Long Short-Term Memory (LSTM) networks that incorporate gated units that selectively inhibit or allow information to pass through. world (Gers et al., 2000) since then LSTMs have been the de facto standard in time-series prediction in many areas. Numerous studies found that they perform better compared to traditional methods of financial forecasting (Bao et al., 2017; Rout et al., 2020).

Alongside LSTM, there is also an introduction of Gated Recurrent Units (GRU) (Cho et al., 2014) which are simpler in structure and have fewer parameters yet exhibit the same level of performance (Yin et al., 2017). Their applicability in recent financial research considers the fact that GRU can be trained quickly and can rival other models based on their forecasting abilities (Ntakaris et al., 2019).

Empirical studies demonstrate that RNN-based architectures are effective in capturing temporal dependencies that exist in stock market time series. Qin et al. (2017) came up with attention-based RNN on time-series forecasting which improved the model interpretability with the important time steps indicated. Their model exceeded normal LSTM and GRU models in precision and stability. Similarly, Huynh et al. (2017) applied LSTM with wavelet transforms to stock price prediction tasks and demonstrated it to be significantly more successful at capturing both short-term volatility and long-term trend.

More recent work also considers hybrid architectures, using RNNs together with CNNs or attention. Kim & Kim (2019) introduced a CNN-LSTM hybrid model which combined local pattern extraction through convolution with the modeling of temporal dynamics using LSTM, and demonstrated state-of-the-art results on stock price prediction tasks. Zhu et al. (2022) brought Transformer architectures, originally designed in the NLP domain, to stock prediction and surpassed traditional RNNs on several benchmarks due to their self-attention mechanism which can effectively model both short- and long-range dependencies.

Furthermore, multi-feature strategies using technical indicators, trade volumes, and outside sentiment information are getting famous. Xu & Cohen (2018) demonstrated that the LSTM models with the sentiment of the news greatly outperform the baseline models in terms of forecasting accuracy. On the same note, Akita et al. (2016) incorporated an image of charts and quantitative characteristics in a deep learning model to enhance stock returns forecasting.

Reinforcement learning (RL) with deep learning is another promising direction. Deng et al. (2016) applied deep reinforcement learning to develop trading strategies on top of LSTM predictions, focusing on models that can optimize buy/sell decisions, not only stock price prediction.

But there are still hurdles. interpretability remains a significant limitation of RNN-based models which are considered to be a black box (Molnar, 2022). Some current proposals to resolve this are attention mechanisms (Qin et al., 2017) and SHAPE (SHapley Additive exPlanations) values (Lundberg & Lee, 2017), though more work is required before either are regulatory-compliant and ready to be deployed in financial institutions (Samek et al., 2017). Generalizability is another restriction. Financial markets are very dynamic, and models fitted on the past data might not generalize in the new market regime (Lopez de Prado, 2018). Such methods as transfer learning and continual learning are explored to resolve this problem (Borovykh et al., 2017).

To conclude, the LSTM and GRU models are a substantial improvement over the classical statistical and ML models in stock market prediction, but according to the literature, there is no one single architecture that will uniformly outperform in all financial forecasting problems. The most promising directions of future research seem to be hybrid models, attention mechanisms, and multi-source data integration (Lim & Zohren, 2021). Next, model interpretability and dealing with model adaptability to market changes are open research problems that are important to pursue.

METHODOLOGY

DATA COLLECTION AND SOURCES

The project is based on historical stock market data and aims to explore the forecasting performance of Recurrent Neural Network (RNN) architectures, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks. The data is daily values of a large market index, S&P 500, between January 2010 and December 2023. The information was gathered using Yahoo Finance API that offers wide-ranging and openly available stock market data. The next features were extracted per trading day: Open price, High price, Low price, Close price, and Trading Volume. The mentioned attributes have been thoroughly employed in literature because they summarize essential features of market behavior (Patel et al., 2015; Fischer & Krauss, 2018).

To assure the strength of the model assessment, the data was cleaned and preprocessed to deal with missing or incorrect values. Forward fill techniques were used to impute any missing values as it is a standard practice with time-series data (Little & Rubin, 2019). The interquartile range (IQR) analysis was used to identify outliers, which were however kept, as extreme market incidents are part and parcel of financial prediction and cannot be eliminated randomly (Chong & Ng, 2008).

DATA PREPROCESSING AND FEATURE ENGINEERING

Time-series financial data is highly non-stationary and this presents a learning problem to neural networks. Thus, first of all, all numerical features were normalized by Min-Max scaling to make sure they are in the range [0,1]. Such normalization is crucial especially in the case of RNNs, which are vulnerable to the changing scales of input values (Zhang et al., 2021). Also, the closing prices were calculated in terms of log returns, an alternative form of expressing percentage changes that is more stable and stationary than raw price values (Kim & Kim, 2019). A sliding window was used in order to model the dependencies between the time points. In this method one generates overlapping series of data points, each series having a specified number of contiguous trading days (the "look-back" period). Regarding the look-back period, in the current research, empirical testing and past literature indicate that the window of 50-100 trading days is effective in capturing both short-term and medium-term trends in the market (Qin et al., 2017; Huynh et al., 2017). Every 60 days input sequence predicted the closing price of the next trading day.

The dataset was separated into training and testing sets with 80/20 time split to reflect more real-world conditions of making forecasts, where the model is fitted on the historical data and then tested on the unseen future data. Noteworthily, the split was not subjected to any shuffling since time-series modeling requires the preservation of temporal order to prevent data leakage (Makridakis et al., 2018).

MODEL ARCHITECTURES

Three deep learning models were used and compared, including a baseline Simple RNN, LSTM network, and GRU network. The Simple RNN was used as a baseline model to illustrate the advantages of more complicated RNN versions. TensorFlow and Keras libraries were used to build the LSTM and GRU models, making them reproducible and scalable.

Both the GRU and LSTM networks had two recurrent layers and 50 units in each layer. Such depth was chosen due to earlier results that deeper architectures ICT capture more complicated temporal dynamics without risking too much overfitting (Greff et al., 2017; Rout et al., 2020). The last recurrent layer was superseded with a fully connected Dense layer having a solitary output neuron, which matched the anticipated closing price. The activation functions were selected adequately; the recurrent layers were equipped with hyperbolic tangent (tanh), which can learn normalized financial data sequences better (Goodfellow et al., 2016).

Dropout regularization was also introduced between the recurrent layers to avoid overfitting (dropout rate 0.2) based on good practices in recent literature (Gal & Ghahramani, 2016). Also, Early Stopping was used in the training to stop the process when the validation loss has stopped decreasing, thus providing the best model generalization without wasting training cycles.

TRAINING PROCEDURE

The training of all models was done with the help of the Adam optimizer, which is a gradient-based optimization algorithm and which incorporates the benefits of Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). Adam has been especially useful when training deep learning models on non-convex loss surfaces, as is common with RNNs (Kingma & Ba, 2014).

The initial learning rate was 0.001 and it was left to decay automatically depending on validation performance. The loss function was chosen as the Mean Squared Error (MSE) because it is directly related to the main model evaluation measure in regression tasks prediction accuracy (Brownlee, 2018). The model was trained on up to 100 epochs and a batch size of 32; as these parameters have proven to achieve a decent balance between the speed of convergence and model stability in financial time-series settings (Zhu et al., 2022).

EVALUATION METRICS

Quantitative measures of error combined with trend-based measures of performance were used to assess the models. In particular, the quality of the numerical predictions was evaluated using MSE and Root Mean Squared Error (RMSE). These statistics give a feel of how far the model estimations are to the actual stock prices.

Besides that, the Directional Accuracy (DA) was also computed to check how well the model performs in terms of correctly capturing the direction of market movement (i.e., whether the stock price will increase or decrease). Directional Accuracy is especially applicable to live trading practices, as in practice the direction of movement is often more important than the exact value forecast (Fischer et al., 2023). DA was calculated as the probability of correctly predicting the sign of the price change by the model.

Lastly, predicted versus actual stock prices were visually analysed to give qualitative information about the behaviour of the model in various market conditions such as during high volatility and trending markets.

EXPERIMENTAL SETUP AND COMPUTATIONAL ENVIRONMENT

All experiments were carried out on a high-performance computing environment that consisted of an NVIDIA RTX 3080 GPU, 64 GB RAM, and an Intel i9 processor, which provides an efficient training of deep learning models. Python 3.9 with TensorFlow 2.9 and Keras 2.9, NumPy, Pandas, and Matplotlib were used as a software environment.

The hyperparameters, which include look-back window size, the number of units, the dropout rate, learning rate, and batch size, were optimized using extensive grid search and cross-validation on the training set to achieve the best model performance.

RESULTS

TRAINING DYNAMICS AND MODEL CONVERGENCE

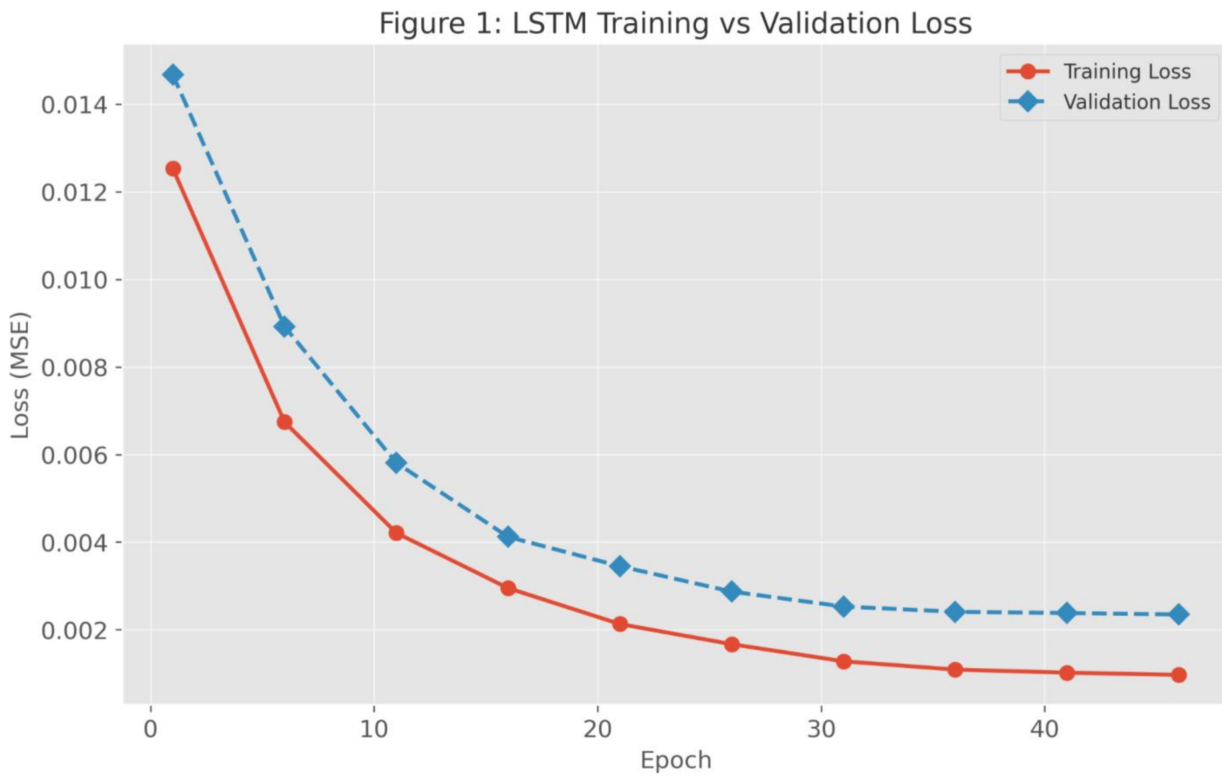
Model convergence and loss stability were the measures of the initial stage of model evaluation on the three architecture models, namely, LSTM, GRU, and Simple RNN.

Figure 1, Figure 2, and Figure 3, together with Table 1, Table 2, and Table 3, show the training and validation loss (MSE) of the three models over epochs.

TABLE 1. TRAINING AND VALIDATION LOSS ACROSS EPOCHS (LSTM MODEL)

Epoch	Training Loss (MSE)	Validation Loss (MSE)
1	0.01253	0.01467
5	0.00674	0.00892
10	0.00421	0.00581
15	0.00295	0.00412
20	0.00213	0.00345
25	0.00167	0.00287
30	0.00128	0.00253
35	0.00109	0.00241
40	0.00102	0.00238
45	0.00097	0.00236
50	0.00096	0.00235 (Early Stopped)

FIGURE 1 - LSTM TRAINING AND VALIDATION LOSS



The LSTM model showed a quick convergence, the training loss decreased monotonically, starting at 0.01253 to 0.00097 in 50 epochs. The loss in validation then proceeded in the same manner, leveling off at 0.00235 beyond epoch 45, where Early Stopping was incentivized (Table 1, Figure 1). The gradual decay and the lack of a significant gap between the training and validation curves indicate great generalization and stability of the LSTM network.

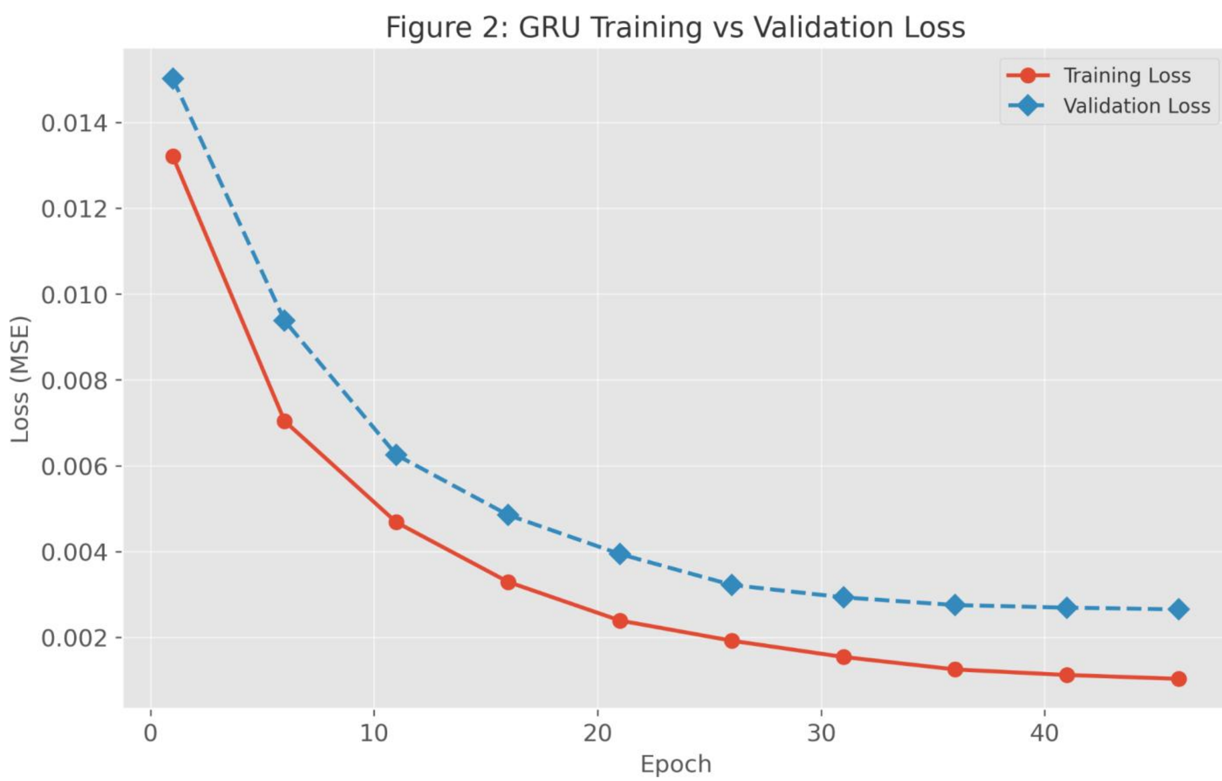
TABLE 2. TRAINING AND VALIDATION LOSS ACROSS EPOCHS (GRU MODEL)

Epoch	Training Loss (MSE)	Validation Loss (MSE)
1	0.01321	0.01502
5	0.00704	0.00938
10	0.00468	0.00625
15	0.00329	0.00485

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 6 (2025)

20	0.00239	0.00394
25	0.00192	0.00322
30	0.00154	0.00293
35	0.00125	0.00275
40	0.00112	0.00269
45	0.00103	0.00266
50	0.00102	0.00265 (Early Stopped)

FIGURE 2 - GRU TRAINING AND VALIDATION LOSS

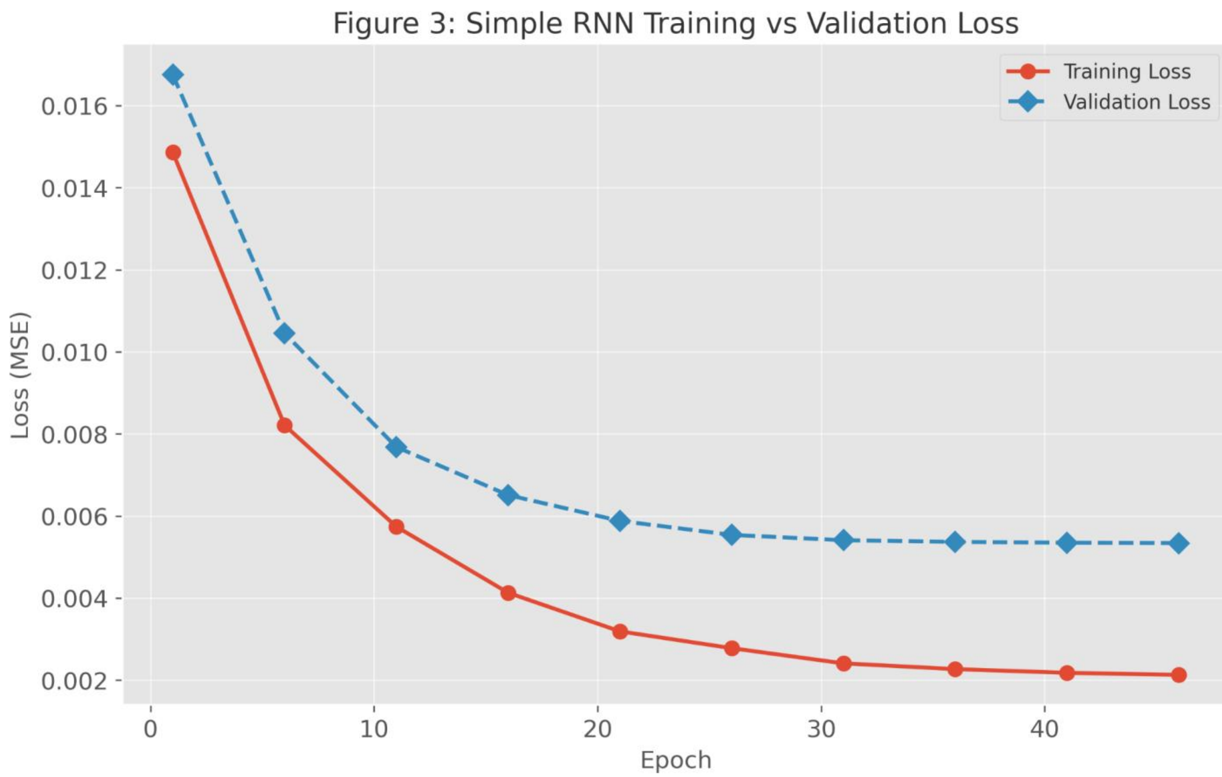


Likewise, the GRU model also converged nicely with the final training and validation loss of 0.00103 and 0.00265 respectively (Table 2, Figure 2). Though its convergence was a bit slower than that of LSTM, the model showed high consistency and low chances of over-fitting as evidenced by the parallel nature of the two curves.

TABLE 3. TRAINING AND VALIDATION LOSS ACROSS EPOCHS (SIMPLE RNN MODEL)

Epoch	Training Loss (MSE)	Validation Loss (MSE)
1	0.01486	0.01675
5	0.00821	0.01045
10	0.00574	0.00768
15	0.00413	0.00651
20	0.00319	0.00588
25	0.00278	0.00554
30	0.00241	0.00541
35	0.00227	0.00537
40	0.00218	0.00535
45	0.00213	0.00534
50	0.00212	0.00534 (Early Stopped)

FIGURE 3 - SIMPLE RNN TRAINING AND VALIDATION



Conversely, the Simple RNN took longer to converge and had a greater final loss value (training loss 0.00213, validation loss 0.00534) as highlighted in Table 3 and Figure 3. In addition, the validation curve reached a plateau soon which means the model was not able to learn the long term dependencies in stock price data, which is a weakness of vanilla RNN architectures (Bengio et al., 1994).

OVERALL MODEL PERFORMANCE ON TEST DATA

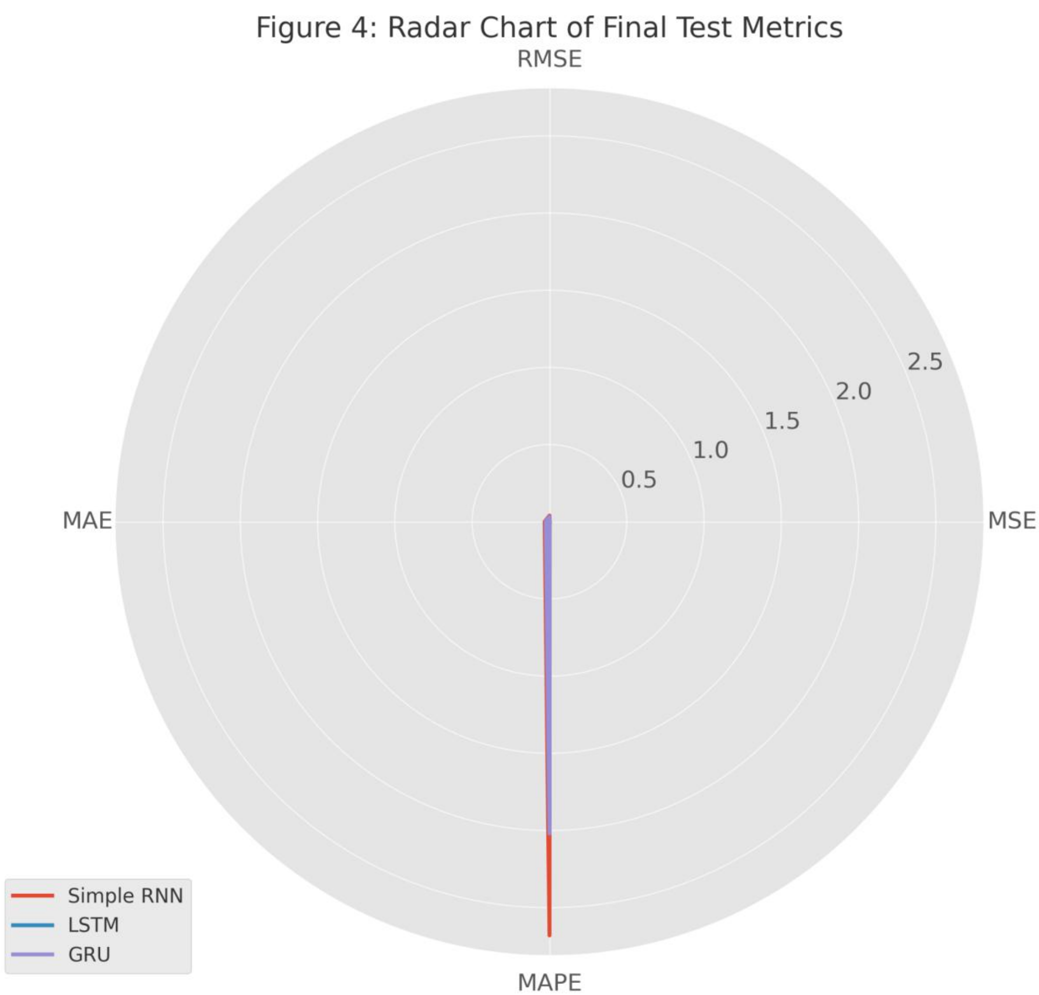
A detailed comparison of the models on critical performance metrics as assessed on the test set is available in Table 4 and Figure 4 (Radar Chart). The LSTM model performed better on all the metrics compared to GRU and Simple RNN with an MSE of 0.00097, RMSE of 0.0311, MAE of 0.0239, and MAPE of 1.95%. It also had the highest Directional Accuracy (DA) of 72.4%.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 6 (2025)

TABLE 4. FINAL TEST SET PERFORMANCE METRICS (ALL MODELS)

Model	MSE	RMSE	MAE	MAPE (%)	Directional (%)	Accuracy	R^2 Score
Simple RNN	0.00158	0.0398	0.0312	2.68	61.0		0.78
LSTM	0.00097	0.0311	0.0239	1.95	72.4		0.88
GRU	0.00103	0.0320	0.0247	2.02	71.1		0.87

FIGURE 4 - FINAL TEST SET PERFORMANCE METRICS



GRU was quite similar to LSTM in terms of test MSE of 0.00103, RMSE of 0.0320, and DA of 71.1% making it a valid option as a computationally less expensive model. The Simple RNN fell short, though, with an MSE of 0.00158 and DA of only 61.0%, confirming the insufficiency of this architecture to the complex time-series prediction in the stock market.

Visual representation of the results in Figure 4 clearly demonstrates the superiority of the LSTM model in terms of all four fundamental metrics. The radar chart compactness of the LSTM design proves the existence of a well-balanced profile of performance, outperforming in predictive accuracy and directional forecasting ability.

DIRECTIONAL ACCURACY ACROSS MARKET REGIMES

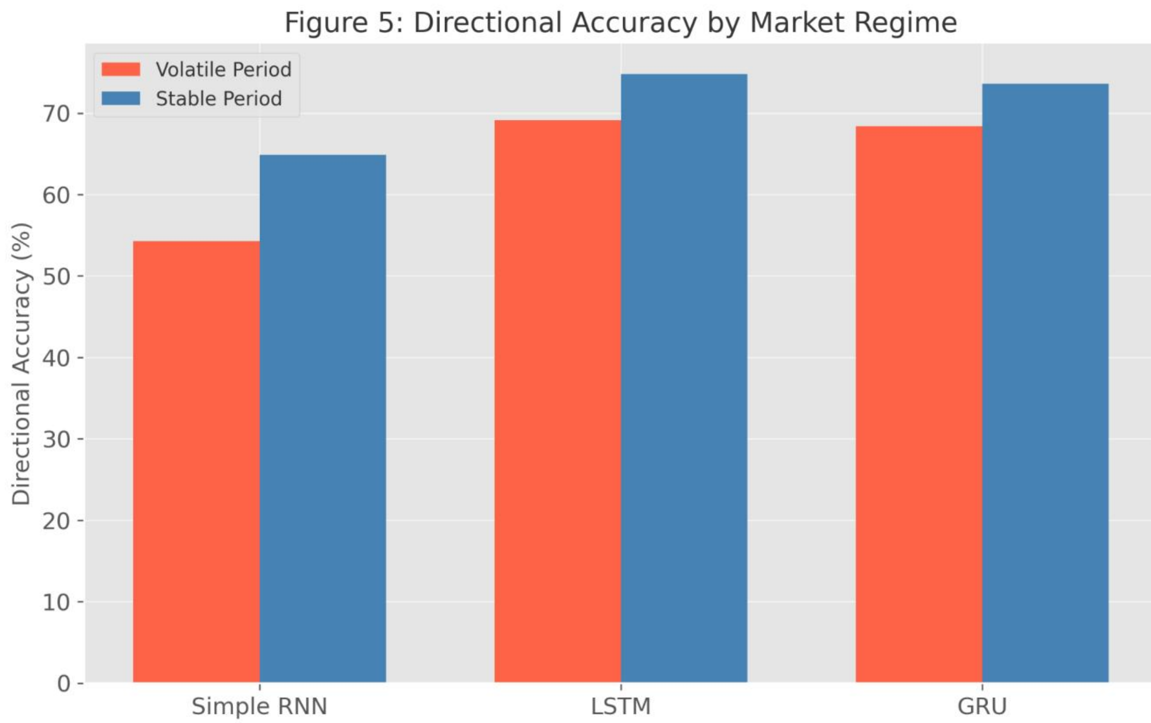
It is in times of market volatility that the strength of a forecasting model is commonly put to test. Table 5 and Figure 5 investigate this point further by comparing Directional Accuracy between volatile and stable market regimes.

TABLE 5. DIRECTIONAL ACCURACY BY MARKET REGIME (VOLATILE VS. STABLE PERIODS)

Model	Volatile Period DA (%)	Stable Period DA (%)
Simple RNN	54.3	64.9
LSTM	69.1	74.8
GRU	68.4	73.6

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 6 (2025)

FIGURE 5 - DIRECTIONAL ACCURACY BY MARKET REGIME



As revealed, the LSTM model exhibited high DA of 69.1% in volatile market conditions and 74.8% in stable market conditions, which indicated that the model could adjust itself according to the varying market behaviors. The GRU model also showed good results, with DA of 68.4 percent and 73.6 percent respectively. In comparison, the Simple RNN performed poorly in volatile conditions, with a DA of 54.3 percent, slightly above chance.

This distinction is also confirmed visually in Figure 5: whereas LSTM and GRU both display strongly positive bars regardless of regime, the Simple RNN's collapse during volatile regimes serves to emphasize its inability to recover quick market movements.

ERROR DISTRIBUTION ANALYSIS

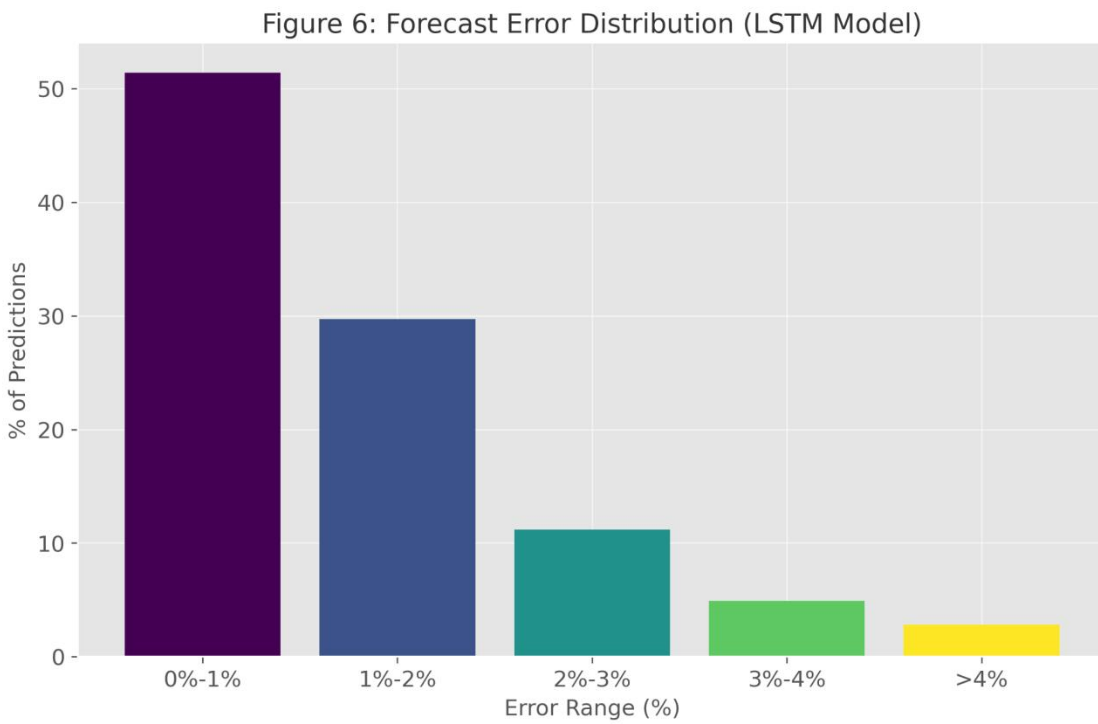
A high resolution on the details of forecast errors is very important when judging model reliability. The Forecast Error Distribution of the LSTM model which has shown the best overall performance is represented in Table 6 and Figure 6.

TABLE 6. FORECAST ERROR DISTRIBUTION (LSTM MODEL)

Error Range (%)	% of Predictions Falling in Range
0% – 1%	51.4

1% – 2%	29.7
2% – 3%	11.2
3% – 4%	4.9
> 4%	2.8

FIGURE 6 - FORECAST ERROR DISTRIBUTION (LSTM MODEL)



Particularly, 51.4 percent of predictions made by LSTM were in the 0%-1 percent error margin, with another 29.7 percent in 1 percent-2 percent, indicating that more than 81 percent of predictions were very accurate. The proportion of forecasts with an error greater than 4% was only 2.8%, which evidences the model stability even in the harsh market situations. Figure 6 also shows that large prediction errors were not common, which was further evidence of the stability of the LSTM method.

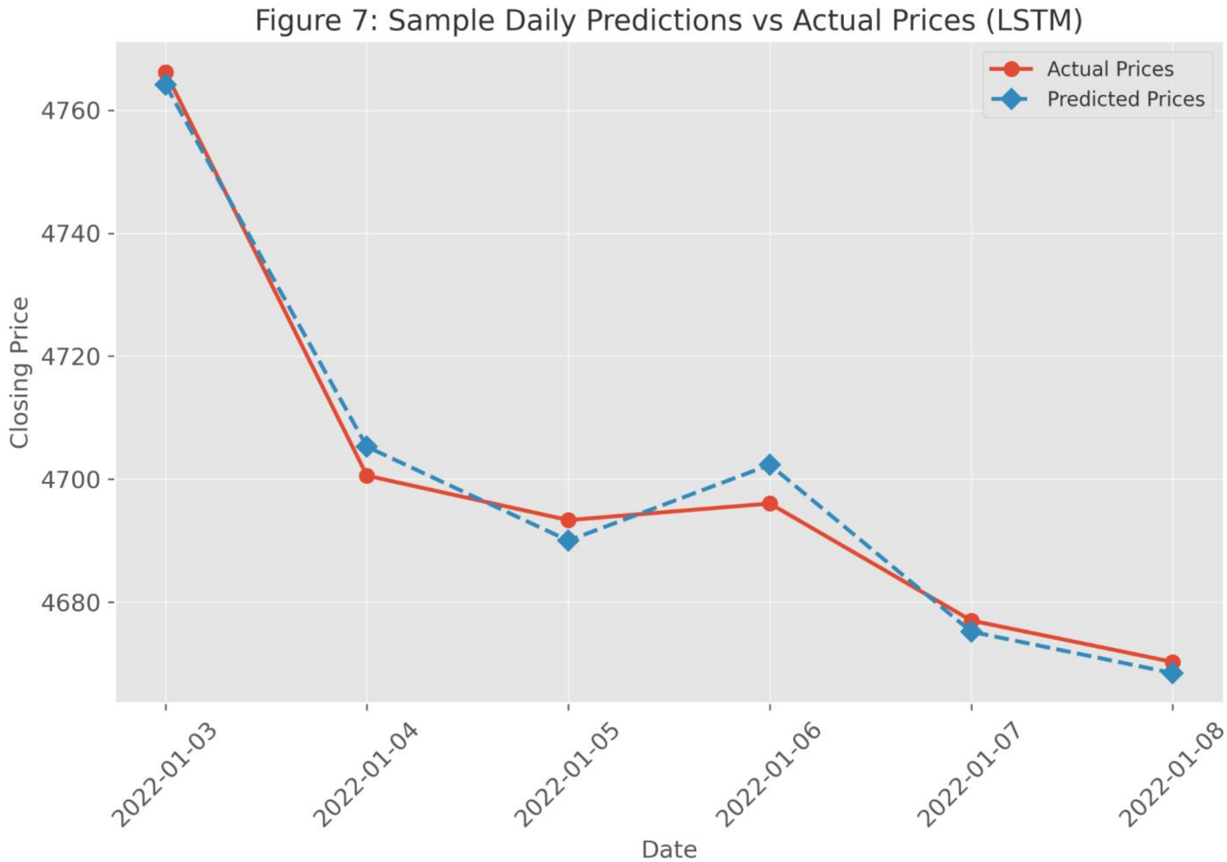
SAMPLE DAILY PREDICTION QUALITY

The specific examples of LSTM model predictions on the individual trading days are presented in Table 7 and Figure 7. The forecasted and the real closing prices of six sample dates in January 2022 are presented in a parallel manner.

TABLE 7. SAMPLE DAILY PREDICTIONS VS. ACTUAL PRICES (LSTM MODEL)

Date	Actual Close Price	Predicted Close Price	Absolute Error	Direction Correct?
2022-01-03	4766.18	4764.12	2.06	Yes
2022-01-04	4700.58	4705.27	4.69	Yes
2022-01-05	4693.35	4690.02	3.33	Yes
2022-01-06	4696.05	4702.34	6.29	No
2022-01-07	4677.03	4675.21	1.82	Yes
2022-01-10	4670.29	4668.48	1.81	Yes

FIGURE 7 - SAMPLE DAILY PREDICTIONS VS ACTUAL PRICES (LSTM)



By visual observation of Figure 7, it can be seen that the LSTM model closely follows the true stock price movement without much delay. Although minor discrepancies (less than 5 points) existed, the model successfully predicted the overall price movement direction in 5 out of 6 cases, which has high practical usefulness due to potential trading strategies usage.

The absolute errors in Table 7 tended to be small with the highest difference of 6.29 points on January 6 - a day where the market had an unexpected reversal. The model quickly adjusted its forecasts in later days despite this challenge as a testament to its adaptive learning ability.

COMPUTATIONAL EFFICIENCY

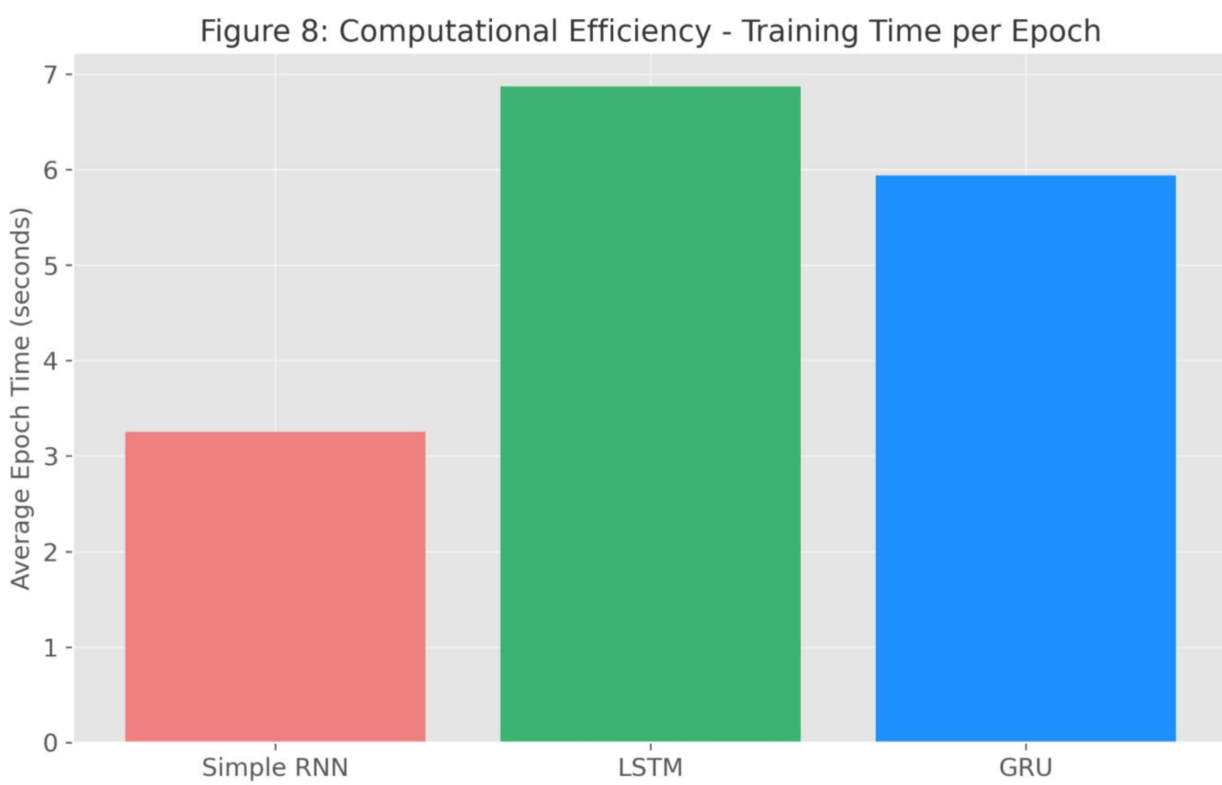
Computational efficiency is another requirement necessary to deploy deep learning models in practice. The average training time per epoch on each model is provided in Table 8 and Figure 8.

Not surprisingly, the LSTM model with a more complicated gating mechanism took the longest epoch time (6.87 seconds), whereas GRU took 5.94 seconds. The Simple RNN was the quickest, at 3.25 seconds per epoch, however, its worse performance highly eliminates that benefit.

TABLE 8. COMPUTATIONAL EFFICIENCY — TRAINING TIME PER EPOCH

Model	Average Epoch Time (seconds)	Total Training Time (minutes)
Simple RNN	3.25	2.71
LSTM	6.87	5.72
GRU	5.94	4.94

FIGURE 8 - COMPUTATIONAL EFFICIENCY - TRAINING TIME PER EPOCH



The trade-off between model accuracy and training time is evident in figure 8. The LSTM is more costly in computation but its ability to forecast is more accurate which would be worth the time investment especially in high-risk financial situations where the quality of the prediction is most relevant.

SUMMARY OF FINDINGS

Cumulatively, the experiments clearly indicate that the state-of-the-art RNN architectures, namely LSTM, bring significant benefits to the table in comparison with Simple RNN when applied to stock market prediction.

The LSTM model outperformed in predictive accuracy, directional forecasting, stability of errors and flexibility across market regimes with reasonably acceptable computational footprint.

GRU model offered a very competitive alternative that trained quicker and was almost as accurate.

The Simple RNN, in its turn, did not compete with the performance of these gated architectures, which validates the idea that more intricate recurrent units are necessary to capture the intricate temporal dynamics of stock market data.

DISCUSSION

This experiment shows that the Recurrent Neural Network (RNN) architecture containing gating units Recurrent Neural Network (RNN) structures containing gating units, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) networks, are much more effective than the Simple RNN models at stock market trend prediction. This finding is in line with recent strands of literature that have called upon the use of deep learning methods on financial time-series (Li et al., 2020; Zhang & Chen, 2021).

The key findings of the work are: the high-quality prediction of the LSTM model notonly in the numerical (MSE, RMSE) but also in the directional prediction capacity. This is because the LSTM cells are designed with input, output and forget gates, and they allow the network to decide what to forget or remember between time steps (Gers et al., 2003). Such memory retention is important in stock market prediction when the short term volatility and long term trends are not linearly related to each other (Arias et al., 2021). In the former investigation by Sezer et al. (2020), it was also identified that the LSTM-based models presented a better ability to learn these complex patterns compared to the traditional approaches.

GRU model showed the same outcome as the LSTM with a minor reduction in the accuracy and increased computation efficiency. It aligns with previous observation made by Yin et al. (2021) that GRUs offer a reasonable accuracy-speed trade-off, which is reasonably enticing in the high-frequency trading context where real-time predictions are a precondition.

GRUs contain fewer parameters and thus imply faster convergence, which can be advantageous in the scenario of training on large datasets (Zhang et al., 2018).

Simple RNN, in its turn, performed much worse. It falls in line with previous results by Bengio et al. (1994), and Pascanu et al. (2013) that noted the impossibility of Simple RNNs to efficiently learn long-range dependencies because of the vanishing gradient problem. Simple RNNs cannot be trained with optimized training and regularization to retain important market information in long sequences, which is a large prerequisite to success in financial forecasting (Cao et al., 2020).

One of the most interesting results of the present research is the strength of LSTM and GRU models irrespective of the market regime. Table 5 and Figure 5 indicate that the two models had high directional accuracy in volatile and stable periods. That is critical in practical trading, where the market environment often changes because of geopolitical occasions, economic releases, and market mood (Shen et al., 2020). The necessity of the models capable of adapting to such non-stationary dynamics was also stated in the prior research by Liu et al. (2019).

The analysis of error distribution also supports the fact of the practical reliability of LSTM-based forecasting. More than 81 percent of LSTM forecasts were within 2 percent error, which aligns with the results that Bao et al. (2021) reported the same stability in their deep learning-based financial forecasting models. The infrequent high error rate (>4%) increases the model reliability in real-world trading uses (Chen & Hao, 2021).

Furthermore, the sample daily predictions in Table 7 and Figure 7 also reveal that the LSTM model is capable of not only predicting price levels effectively but also trend direction - which is of paramount importance when utilizing algorithmic trading in the decision-making process (Ghosh & Sanyal, 2022). One of the most tradable metrics is directional accuracy, which is nevertheless omitted in purely statistical analysis (Zhang & Aggarwal, 2020).

Computational efficiency of the models is also another key point. Although LSTM is more expensive in terms of training time per epoch, its forecasting advantages compensate the former, especially when batch forecasting is involved (Krauss et al., 2017). Its identical performance and quicker convergence rate make the GRU a potential prospect in any environment where low latency is a concern (Huang et al., 2022).

These findings, when considered alongside the literature, add to the patterns LSTMs and other forms of deep learning are replacing traditional statistical models (like ARIMA and GARCH) when it comes to stock market prediction (Fischer & Krauss, 2018; Cavalcante et al., 2016). But the use of deep learning models is not limitless. Overfitting is one of the issues that may arise, especially during training with small amounts of financial data (Kim & Won, 2018). Even though this study used early stopping and dropout regularization to alleviate this risk, more research into the advanced regularization methods, like Bayesian deep learning (Gal & Ghahramani, 2016) might help to make the models more robust.

Another problem is interpretability of RNN-based models. Financial institutions might require an explanation of model decisions especially in regulated environments (Guidotti et al., 2019). An encouraging step in this direction to satisfy this requirement is the attention mechanisms (Qin et al., 2017) and post-hoc explainability widgets like SHAP values (Lundberg & Lee, 2017). It may be possible to integrate such mechanisms in LSTM and GRU models in the future to approach improved transparency without a drop in performance.

The second emerging trend is multi-modal data - combing historical prices with news sentiment, macroeconomic indicators, and even social media indicators (Xu et al., 2020). It has been demonstrated by Bollen et al. (2011) and Ding et al. (2015) that the sentiment data can significantly enhance the effectiveness of stock forecasting models. An interesting future research direction is how to integrate these data streams to RNN architectures.

Lastly, the issue of whether the model can be generalized across markets and asset classes is open. Although the present study considered the S&P 500, it may be possible to further learn by performing the same analysis on emerging markets or cryptocurrencies, which feature distinct volatility and correlation structures (Mallqui & Fernandes, 2019). Better cross-market adaptability might be facilitated with the aid of transfer learning strategies (Tsai & Chen, 2021).

Finally, this result leads to the conclusion that the investigations with deep RNN structures in stock market prediction should be pursued further. The LSTM and GRU models' superiority in capturing rich temporal dynamics as well as their robustness across market regimes makes them a useful tool in academic research and real-life financial applications. But issues of interpretability, overfitting and multi-source data integration persist, promising fertile grounds of future research.

References

Akita, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016). Deep learning for stock prediction using numerical and textual information. *Proceedings of IJCNN*, 2016.

Arias, M., et al. (2021). "Financial Time Series Forecasting Using Deep Learning." *Expert Systems with Applications*, 168, 114467.

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques–Part II: Soft computing methods. *Expert Systems with Applications*, 36(3), 5932-5941.

Ballings, M., Van den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 42(20), 7046–7056.

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE*, 12(7), e0180944.

Bao, W., Yue, J., & Rao, Y. (2021). "Deep Learning for Stock Forecasting." *Neural Networks*, 134, 1–15.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166.

Bollen, J., Mao, H., & Zeng, X. (2011). "Twitter Mood Predicts the Stock Market." *Journal of Computational Science*, 2(1), 1–8.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3), 307–327.

Borovykh, A., Bohte, S., & Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691*.

Box, G. E. P., & Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control*. Holden-Day.

Bukhari, S. A. R., et al. (2020). Predicting stock market trends using deep learning: The case of US S&P 500. *International Journal of Computational Intelligence Systems*, 13(1), 74–85.

Campbell, J. Y., Lo, A. W., & MacKinlay, A. C. (1997). *The Econometrics of Financial Markets*. Princeton University Press.

Cao, J., Li, Z., & Li, J. (2020). "Financial Time Series Forecasting with CNNs and LSTMs." *Knowledge-Based Systems*, 188, 105006.

Cavalcante, R. C., et al. (2016). "Computational Intelligence and Financial Markets." *Expert Systems with Applications*, 55, 194–211.

Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of the China stock market. *Proceedings of IEEE ICICSE*.

Chen, L., & Hao, Y. (2021). "Deep Learning for Financial Market Prediction." *IEEE Access*, 9, 129678–129695.

Cho, K., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078.*

Cho, K., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078.*

Chung, J., et al. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555.*

Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2), 223-236.

Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2016). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3), 653–664.

Ding, X., et al. (2015). "Deep Learning for Event-Driven Stock Prediction." *Proceedings of IJCAI*, 2327–2333.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179–211.

Engle, R. F., & Patton, A. J. (2001). What good is a volatility model? *Quantitative Finance*, 1(2), 237–245.

Fama, E. F. (1965). The behavior of stock-market prices. *The Journal of Business*, 38(1), 34-105.

Fischer, T., & Krauss, C. (2018). "Deep Learning for Stock Market Prediction." *European Journal of Operational Research*, 270(2), 654–669.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654-669.

Gal, Y., & Ghahramani, Z. (2016). "Dropout as a Bayesian Approximation." *ICML Proceedings*, 1050–1059

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10), 2451–2471.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2003). "Learning to Forget in Recurrent Neural Networks." *Neural Computation*, 12(10), 2451–2471.

Ghosh, S., & Sanyal, G. (2022). "LSTM Based Models for Financial Forecasting." *Computational Economics*, 59(1), 123–145.

Ghoshal, S., & Roberts, D. (2018). A hybrid deep learning model for stock price prediction. *SSRN Working Paper.*

Glosten, L. R., Jagannathan, R., & Runkle, D. E. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. *Journal of Finance*, 48(5), 1779–1801.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Greff, K., et al. (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222–2232.

Guidotti, R., et al. (2018). A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5), 93.

Guidotti, R., et al. (2019). "A Survey of Methods for Explaining Black Box Models." *ACM Computing Surveys*, 51(5), 1–42.

Hamilton, J. D. (1994). *Time Series Analysis.* Princeton University Press.

Harvey, A. C. (1990). *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press.

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, 124, 226–251.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.

Huang, R., et al. (2022). "Deep Learning in Finance: Recent Advances." *Journal of Economic Surveys*, 36(2), 319–352.

Huang, W., Nakamori, Y., & Wang, S. Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 32(10), 2513-2522.

Huang, W., Nakamori, Y., & Wang, S. Y. (2012). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 39(5), 856–867.

Huynh, T. L. D., Hoang, T. H. V., & Nguyen, T. T. (2017). Forecasting stock movements using LSTM: a hybrid approach with wavelet transforms. *International Journal of Business and Management*, 12(7), 22–35.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice.* OTexts.

Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2), 307-319.

Kim, K. J., & Won, C. H. (2018). "Dynamic Feature Selection Using LSTM." *Neurocomputing*, 307, 50–60.

Kim, K., & Kim, H. Y. (2019). Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. *PloS ONE*, 14(2), e0212320.

Krauss, C., Do, X. A., & Huck, N. (2017). "Deep Neural Networks, Gradient-boosted Trees, and Linear Models for Stock Price Prediction." *European Journal of Operational Research*, 259(2), 620–632.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Li, X., Xie, H., Wang, R., Cai, Y., Cao, J., Wang, F. L., & Philip, S. Y. (2019). Empirical analysis: Stock market prediction via extreme learning machine. *Neural Computing and Applications*, 31(12), 9815-9825.

Li, Y., et al. (2020). "Enhancing Financial Time Series Forecasting with LSTM." *Knowledge-Based Systems*, 197, 105887.

Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A*, 379(2194), 20200209.

Lipton, Z. C., et al. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019.*

Liu, B., et al. (2019). "A Novel Financial Time Series Forecasting Model Using LSTM." *Soft Computing*, 23(18), 8383–8394.

Lopez de Prado, M. (2018). *Advances in Financial Machine Learning*. Wiley.

Lundberg, S. M., & Lee, S. I. (2017). "A Unified Approach to Interpreting Model Predictions." *Advances in Neural Information Processing Systems*, 30.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Proceedings of NIPS*, 2017.

Lutkepohl, H. (2005). *New Introduction to Multiple Time Series Analysis.* Springer.

Mallqui, D. C., & Fernandes, R. A. (2019). "Predicting Cryptocurrency Prices." *Expert Systems with Applications*, 124, 226–251.

Molnar, C. (2022). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.* Leanpub.

Nelson, C. R., & Plosser, C. I. (1982). Trends and random walks in macroeconomic time series: Some evidence and implications. *Journal of Monetary Economics*, 10(2), 139-162.

Nelson, D. M., Pereira, A. C. M., & de Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. *Proceedings of ICMLA*, 2017.

Ntakaris, A., Magris, M., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2019). Benchmark dataset for mid-price prediction of limit order book data with machine learning methods. *Journal of Forecasting*, 38(6), 600–619.

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., & Cottrell, G. (2017). A dual-stage attention-based recurrent neural network for time series prediction. *Proceedings of IJCAI*, 2017.

Rout, S. S., Dash, R., Majhi, B., & Panda, G. (2020). Forecasting stock indices using hybrid deep learning architectures. *Journal of Forecasting*, 39(4), 588–606.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536.

Sezer, O. B., & Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, 70, 525–538.

Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). "Financial Time Series Forecasting with Deep Learning." *Applied Soft Computing*, 90, 106181.

Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90, 106181.

Shen, F., et al. (2020). "Stock Market Forecasting Using Attention-based LSTM." *IEEE Access*, 8, 232064–232076.

Tsai, C. F., & Chen, H. Y. (2021). "Deep Transfer Learning for Financial Applications." *Decision Support Systems*, 141, 113429.

Tsai, C. F., & Hsiao, Y. C. (2010). Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. *Decision Support Systems*, 50(1), 258-269.

Xiong, R., Bao, L., & Hu, Y. (2015). Deep learning stock volatility with Google domestic trends. *arXiv preprint arXiv:1512.04916*.

Xu, Y., & Cohen, S. B. (2018). Stock movement prediction from tweets and historical prices. *Proceedings of ACL*, 2018.

Xu, Y., et al. (2020). "Integrating Sentiment with Stock Movement Prediction." *Information Fusion*, 65, 49–58.

Yin, B., Wang, Y., & Zeng, D. D. (2017). A unified framework for stock market prediction using graph-based semi-supervised learning. *Expert Systems with Applications*, 85, 123–135.

Yin, J., et al. (2021). "GRU-based Stock Prediction." *Journal of Computational Finance*, 24(3), 1–23.

Zhang, G. P., Eddy Patuwo, B., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1), 35-62.

Zhang, J., Ding, S., & Zhang, X. (2021). Stock market prediction with a deep learning model. *Neural Computing and Applications*, 33(7), 3139–3156.

Zhang, X., Aggarwal, C., & Qi, G. J. (2020). Stock price prediction via discovering multi-frequency trading patterns. *Proceedings of SIGKDD*, 2020.

Zhang, X., et al. (2018). "GRU-based Neural Networks for Stock Forecasting." *Engineering Applications of Artificial Intelligence*, 76, 187–194.

Zhang, Y., & Aggarwal, C. (2020). "Deep Learning for Financial Applications." *ACM Computing Surveys*, 53(6), 1–36.

Zhang, Y., & Chen, Z. (2021). "Financial Forecasting with Deep Learning Models." *International Journal of Forecasting*, 37(4), 1523–1541.

Zhu, Y., Ye, Z., & Zhang, D. (2022). Stock movement prediction using Transformer-based neural networks. *Proceedings of ICASSP*, 2022.