


## Article

# Predicting Economic Trends and Stock Market Prices with Deep Learning and Advanced Machine Learning Techniques

Victor Chang <sup>1,\*</sup> , Qianwen Ariel Xu <sup>1</sup>, Anyamele Chidozie <sup>2</sup> and Hai Wang <sup>3</sup>

<sup>1</sup> Department of Operations and Information Management, Aston Business School, Aston University, Birmingham B4 7ET, UK; qianwen.ariel.xu@gmail.com

<sup>2</sup> School of Computing, Engineering and Digital Technologies, Teesside University, Middlesbrough TS1 3BX, UK

<sup>3</sup> School of Computer Science and Digital Technologies, Aston University, Birmingham B4 7ET, UK; h.wang10@aston.ac.uk

\* Correspondence: v.chang1@aston.ac.uk or victorchang.research@gmail.com

**Abstract:** The volatile and non-linear nature of stock market data, particularly in the post-pandemic era, poses significant challenges for accurate financial forecasting. To address these challenges, this research develops advanced deep learning and machine learning algorithms to predict financial trends, quantify risks, and forecast stock prices, focusing on the technology sector. Our study seeks to answer the following question: “Which deep learning and supervised machine learning algorithms are the most accurate and efficient in predicting economic trends and stock market prices, and under what conditions do they perform best?” We focus on two advanced recurrent neural network (RNN) models, long short-term memory (LSTM) and Gated Recurrent Unit (GRU), to evaluate their efficiency in predicting technology industry stock prices. Additionally, we integrate statistical methods such as autoregressive integrated moving average (ARIMA) and Facebook Prophet and machine learning algorithms like Extreme Gradient Boosting (XGBoost) to enhance the robustness of our predictions. Unlike classical statistical algorithms, LSTM and GRU models can identify and retain important data sequences, enabling more accurate predictions. Our experimental results show that the GRU model outperforms the LSTM model in terms of prediction accuracy and training time across multiple metrics such as RMSE and MAE. This study offers crucial insights into the predictive capabilities of deep learning models and advanced machine learning techniques for financial forecasting, highlighting the potential of GRU and XGBoost for more accurate and efficient stock price prediction in the technology sector.

**Keywords:** stock prices; deep learning; artificial neural networks; recurrent neural networks; long short-term memory (LSTM); gated recurrent unit (GRU)



**Citation:** Chang, V.; Xu, Q.A.; Chidozie, A.; Wang, H. Predicting Economic Trends and Stock Market Prices with Deep Learning and Advanced Machine Learning Techniques. *Electronics* **2024**, *13*, 3396. <https://doi.org/10.3390/electronics13173396>

Academic Editor: Simeone Marino

Received: 17 June 2024

Revised: 15 August 2024

Accepted: 22 August 2024

Published: 26 August 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The finance sector is a crucial domain for applying advanced deep learning (DL) and machine learning (ML) models due to its dynamic nature and the significant stakes involved in financial decision-making. Accurate financial forecasting in this sector can lead to substantial economic benefits, reduced risks, and more informed decisions. In the complex and constantly evolving world of finance, forecasting has been a key focus for many researchers over the years. The volatility and unpredictability of the stock market present significant challenges for investors. Predicting the future performance of companies through financial forecasting is one of the most extensively studied applications in the finance industry. Accurate stock price predictions play a critical role in making profitable investment decisions, although the inherent complexities of the financial market make it a formidable task.

Companies raise capital by dividing their ownership and selling shares, making stock price prediction a significant financial application. Stock prices fluctuate based on factors

such as company performance, brand value, market activity, inflation, trends, and investor sentiment. While some aspects, like sales and purchases, can be estimated, the complexities add a layer of difficulty to developing accurate models that capture trends and forecast future prices. Predicting future trends can be the difference between investment success and failure for investors. Traditional methods, such as technical and fundamental analyses, have been used to study patterns and predict future stock prices, but they often fall short when dealing with the dynamic, non-stationary nature of stock markets influenced by factors like announcement headlines, social media tweets, corporate news, and other mood indicators [1,2].

Over the years, numerous statistical methods like regressions and time series models (ARIMA, SARIMA [3], GARCH) have been employed to predict future stock prices. While beneficial in some respects, these methods struggle with handling stock price data. For example, the autoregressive integrated moving average (ARIMA) model has been applied to predict the stock market using historical financial data; however, these statistical models often fall short due to the non-linear structure of time series data [4].

To overcome the inefficiencies of statistical methods, various Artificial Intelligence (AI) models have been developed and integrated into statistical analysis to predict future stock market trends. These include classical machine learning (ML) algorithms such as Support Vector Machines (SVMs) [5] and Random Forest (RF) [6], as well as deep learning (DL) algorithms such as recurrent neural networks (RNNs) [7], Convolutional Neural Networks (CNNs) [8], and other deep learning methods for multivariate time series data analysis. Rao and Reimherr introduce a novel class of non-linear function-on-function regression models specifically designed for functional data using neural networks. The authors propose two model-fitting strategies: Function-on-Function Direct Neural Networks (FFDNNs) and Function-on-Function Basis Neural Networks (FFBNNs). These strategies are tailored to leverage the inherent structure of functional data and capture complex relationships between functional predictors and responses [9]. These AI models, with their capacity to learn from extensive datasets and continuously improve, offer promising potential for automated and more accurate future stock price predictions.

Deep learning methods have been extensively used in the existing literature to predict future stock prices, significantly contributing to improved model accuracy [10]. White was a pioneer in implementing an artificial neural network (ANN) for financial market forecasting, using the daily prices of IBM as a database [11]. Although this initial study did not achieve the expected results, it highlighted several difficulties, such as the overfitting problem and the low complexity of the neural network, which used only a few entries and one hidden layer. This study highlighted possible future improvements, including adding more features to the ANN, working with different forecasting horizons, and evaluating model profitability. Over the years, deep learning capabilities have greatly improved, and various parameter tuning methods have been developed to address the issues mentioned by White [11]. A family of recurrent neural network (RNN) architectures, including variations of gated recursive units (GRUs) and long and short-term memory (LSTM), have become popular methods for predicting stock market patterns. Recent studies highlight the effectiveness of combining sentiment analysis with deep learning models. For instance, Sonkiya et al. used BERT for sentiment analysis and GANs for stock price prediction, showing improved performance over traditional methods like ARIMA and neural networks such as LSTM and GRU [12]. Similarly, Maqsood et al. demonstrated that incorporating sentiment from local and global events into deep learning models enhances prediction accuracy, as evidenced by improved RMSE and MAE metrics [13]. Another innovative approach by Patil et al. utilized graph theory to model the stock market as a complex network. Their hybrid models, which combined graph-based structural information with deep learning and traditional machine learning techniques, outperformed standard models by leveraging the spatio-temporal relationships between stocks [14].

Despite these advancements, there is a notable gap in current research. Comparative analyses of LSTM and GRU for predicting stock prices of technology companies are

insufficient. Existing studies often lack the necessary industry specificity, resulting in unsatisfactory predictions when models trained on general stock market data are applied to specific industries or companies. This study aims to address this gap by focusing on the technology industry and applying LSTM and GRU models to enhance the precision of technology stock forecasts. By comparing these models, we aim to determine the more effective method for predicting technology sector stock prices, ultimately aiding investors in making data-driven decisions.

This study uniquely applies LSTM and GRU deep learning models, along with various machine learning algorithms, to predict stock prices in the technology sector. We aim to identify the more effective model among them, offering a crucial contribution to financial forecasting. The objective is to better understand the patterns, trends, and volatility of the tech stock market and develop an efficient model to bolster the accuracy of tech stock forecasts, enabling data-driven decision-making for investors.

The remainder of this paper is structured as follows: Section 2 provides a brief introduction to the various computational methods and data analysis techniques utilized in our study. Section 3 covers the preliminaries, our approach, and illustrative examples. Section 4 presents the numerical results, while Section 5 details additional experiments and validations. Finally, Section 6 concludes the paper and highlights directions for future research.

## 2. Theoretical Background

This section outlines the various computational methods and data analysis techniques employed in our study to predict stock price movements. The theoretical foundation of our approach relies on both deep learning and traditional machine learning frameworks. Deep learning is particularly adept at processing and learning from large datasets, making it ideal for the complex patterns observed in stock market data. Machine learning algorithms like XGBoost complement deep learning by providing efficient, scalable methods for regression and classification.

### 2.1. Review of the LSTM and GRU Architecture

In traditional neural networks, the output of a neuron is rarely used as an input for the next step. When we focus on a proven oddity, though, we observe that our ultimate output is often influenced by both external inputs and prior produce. For example, while reading a book, comprehension of each sentence is based on both the current flow of words and the context set by previous sentences. Traditional neural networks do not have the idea of ‘context’ or ‘constancy’.

A simple RNN with an input circle produces a result  $h_t$  for some information  $x_t$  at time step  $t$ . It uses two bits of information,  $x_{t+1}$  and  $h_t$ , to obtain the output  $h_{t+1}$  at the next time step  $t + 1$ . Data may be transmitted from one network step to the next using a circle. An RNN, on the other hand, is not without impediments. When the setting is from a long time ago, it helps tremendously to get the intended result. However, RNNs face challenges when required to rely on distant past information to produce the desired output. This RNN stumbling block was extensively studied by Hochreiter [15] and Bengio et al. [16], who also identified the underlying theories to determine why RNNs may not work in the long run. Fortunately, LSTM models and GRUs are built to address these challenges.

The standard neural network is severely constrained without context-based reasoning. To overcome this restriction, the concept of recurrent neural networks (RNNs) has been developed. Figure 1 illustrates a simple RNN with a feedback loop on the left.  $X$  denotes the input layer, and  $A$  is the middle layer consisting of multiple hidden layers that receive  $X$ . The figure compares the simple RNN with a feedback loop to its equivalent unrolled form on the right side. In a time series data sequence, if  $X_0$  is the input at the start time, and  $h_0$  is the output, then  $h_0$  together with  $x_1$  will be the input for the next step, and this process is repeated for all inputs from different time periods, allowing the network to remember the context during training. In the next section, we summarize LSTM and GRU networks.

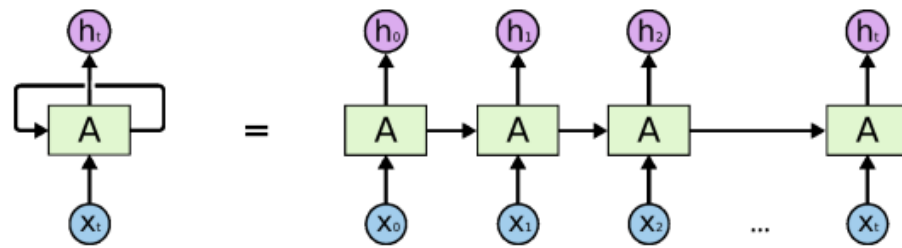


Figure 1. Simple RNN with an input circle and its equivalent unrolled presentation.

LSTM and GRU Networks

Hochreiter and Schmidhuber developed an exceptional type of RNN that can learn over long distances. Various other researchers later improved this leading effort [17–19]. LSTM and GRUs were developed to solve the protracted dependency problem. Sutton and Barto discussed the evolution and refinement of LSTM and GRUs from RNNs [20]. RNNs are made up of a series of repeating neural network modules. In a standard RNN, repeating modules contain a simple computational node, represented by a single tanh activation function, as shown in Figure 2.

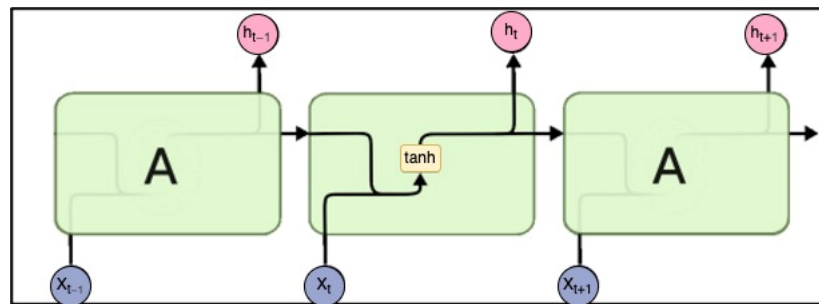


Figure 2. The repeating module in a standard RNN contains a single layer.

LSTM cells can track information over multiple time steps. Information is added or eliminated through structures called gates. Gates naturally allow information through via a sigmoid neural net layer and a pointwise multiplication. The repeating module in an LSTM is shown in Figure 3. LSTM models process the information by first forgetting irrelevant parts of the previous state, then storing the most relevant parts of the new information to the state of the cell, thirdly updating their internal status, and then finally producing the output.

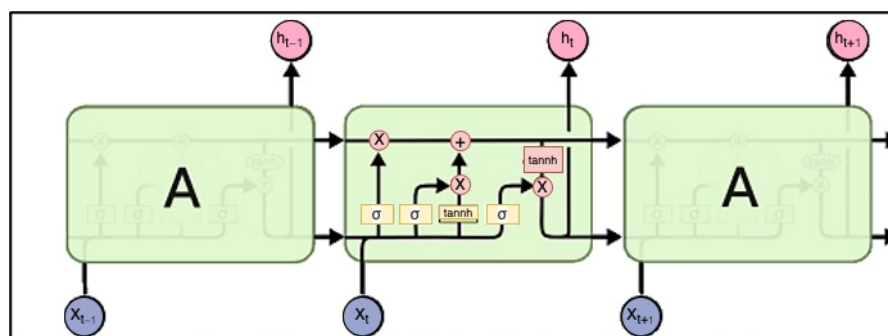


Figure 3. The repeating module in an LSTM.

The forget gate in an LSTM unit determines which cell state information to exclude from the model. The memory cell takes the previous instant  $h_{t-1}$  and the current input information  $x_t$  and transforms them into a long vector  $(h_{t-1}, x_t)$  to become

$$f_t = \sigma \left( W_f \cdot [h_{t-1}, x_t] \right) + b_f \tag{1}$$

where  $W_f$  is the weight matrix associated with the forget gate,  $b_f$  is the bias term, and  $\sigma$  is the sigmoid activation function. To determine how much of the current inputs  $x_t$  should be allocated to the cell  $C_t$ , an input gate is utilized, preventing non-essential information from accessing the memory cells:

$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$C_t = \tan h(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

where  $W_i$  and  $W_c$  are the weight matrices for the input gate and candidate cell state, respectively, and  $b_i$  and  $b_c$  are their respective bias terms. The function  $\tan h$  is the hyperbolic tangent activation function.

$$C_t = f_t * C_{t-1} + i_t * C_t \tag{4}$$

The output gate determines how much of the current cell state is included in the output. The sigmoid layer processes the output information first, followed by the tanh function, and then multiplies it by the sigmoid layer output to get the final output component:

$$o_t = \sigma (W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

where  $W_o$  is the weight matrix for the output gate and  $b_o$  is the bias term.

The final output value of the cell is defined as

$$h_t = o_t * \tanh(C_t) \tag{6}$$

Cho created the Gated Recurrent Unit (GRU), a kind of RNN, in 2014 with the purpose of fixing the vanishing gradient issue of RNNs [21]. The GRU's key benefit over other structures is that it requires fewer parameters, trains quicker, and requires less data to generalize. The structure of the GRU model is shown in Figure 4.

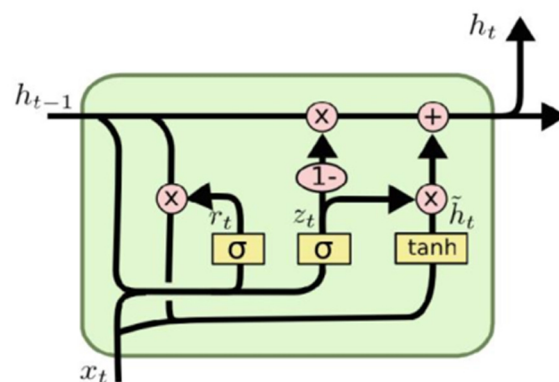


Figure 4. The internal structure of the GRU model.

The update and reset gates produce intermediate values  $z_t$  and  $r_t$ , respectively, while the final memory of the general-purpose unit stores the result  $h_t$  [21]. The update specifies the amount of prior input  $x_t$  and output  $h_{t-1}$  that should be conveyed to the next cell, governed by the weight  $W_t$ . The reset gate determines how much data should be erased from memory.

The following are the most essential equations that characterize the operation of the GRU:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{7}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{8}$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t * h_{t-1}, x_t]) \tag{9}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{10}$$

where  $W_z$ ,  $W_r$ , and  $W_h$  are the weight matrices for the update gate, reset gate, and candidate activation, respectively. The operator  $\cdot$  denotes matrix multiplication, while  $*$  denotes element-wise multiplication. The functions  $\sigma$  and  $\tanh$  are the sigmoid and hyperbolic tangent activation functions, respectively.

In this paper, we will use deep learning (DL) models to analyze selected technological stock patterns as one-dimensional time series and attempt to forecast future stock prices by examining past historical prices and the most critical technical indicators. This research will compare the performance of the LSTM and the GRU ensemble models on selected technology stock data to investigate stock price patterns.

### 2.2. Attention Mechanism

The attention mechanism has recently gained much traction in the context of time series data. Self-attention, global attention, and local attention are examples of attention approaches. In general, applications such as voice recognition, machine translation, and part of speech tagging benefit greatly from the attention mechanism.

The concentrated attention is focused on a single element in the input, which is picked information by maximal or random sampling, and it requires further training to get exceptional outcomes. On the other hand, soft attention is a process that assigns weights to all of the information to allow more effective information utilization. In the soft attention mechanism, the attention score at time  $t$  ( $e_t$ ) is computed using a weight matrix  $W_a$  and a bias term  $b$ , acting on the input elements  $x_1, x_2, \dots, x_T$ :

$$e_t = \tanh(W_a \{x_1, x_2, \dots, x_T\} + b) \tag{11}$$

These scores are then normalized using the softmax function to produce the attention weights ( $a_t$ ):

$$a_t = \frac{\exp(e_t)}{\sum_{k=1}^T \exp(e_k)} \tag{12}$$

The attention mechanism generally involves two steps: the first phase involves calculating the attention distribution, and the second step involves computing the weighted average of the incoming information using the attention distribution as a guide. The process is initiated with the attention scoring function  $S$ , passing the result to the softmax layer, generating the attention weights  $(1, 2, \dots, n)$ . Following that, the softmax layer is handed the attention weights  $1, 2, \dots$ . Finally, the attention weight vector is weighted and averaged against the input data to arrive at the final result. The attention process is shown in Figure 5.

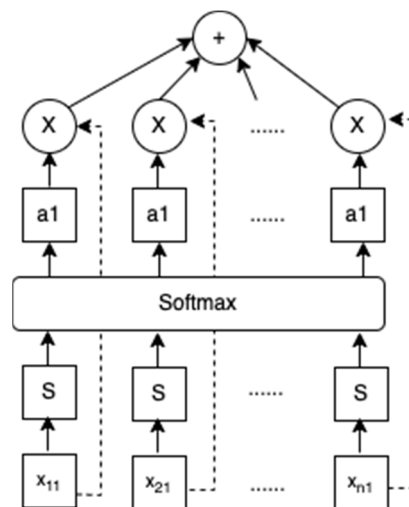


Figure 5. The basic structure of the attention model.

### 2.3. Time Series Forecasting Methods

Time series forecasting is a critical aspect of data analysis and prediction, particularly when dealing with sequential data points recorded at regular intervals. Three methods are introduced as follows.

#### 2.3.1. Autoregressive Integrated Moving Averages (ARIMAs)

Autoregressive integrated moving average (ARIMA) models are a popular choice for stock price prediction due to their ability to handle the complex and dynamic nature of financial time series data [22]. Stock prices often exhibit non-stationary behavior, and ARIMA models excel at differencing the data to achieve stationarity, making them suitable for modeling. Moreover, these models incorporate autoregressive and moving average components, allowing them to capture dependencies on past stock prices and the impact of past shocks, both of which are crucial factors in stock price movements. ARIMA models also offer parameter tuning flexibility, making them adaptable to specific stock price datasets. Their interpretability further aids in understanding the driving factors behind stock price predictions. As a baseline model, ARIMA provides a solid foundation for assessing the performance of more advanced forecasting techniques in our project, making it a valuable choice for stock price prediction tasks [1,4].

The ARIMA model works by first differencing the time series data to achieve stationarity, removing trends and seasonality. Then, it utilizes autoregressive (AR) terms to model the relationship between current and past values and moving average (MA) terms to account for the impact of past shocks or white noise. The model's order, represented as  $(p, d, q)$ , determines the number of AR and MA terms and the degree of differencing needed. The ARIMA model estimates these parameters and fits the model to the data. During forecasting, it uses past observations and model parameters to make predictions for future data points. We employed the auto-ARIMA module from the 'pmdarima' package for our analysis, leveraging its automatic selection of the optimal  $p$ ,  $d$ , and  $q$  terms for our time series model. This approach ensured that we obtained the best possible results, streamlining the modeling process and enhancing forecast accuracy.

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q} \quad (13)$$

where  $\hat{y}_t$  represents the forecasted value,  $\mu$  is the mean term,  $\phi_1, \dots, \phi_p$  are the autoregressive coefficients,  $y_{t-1}, \dots, y_{t-p}$  are the lagged values of the series,  $\theta_1, \dots, \theta_q$  are the moving average coefficients, and  $e_{t-1}, \dots, e_{t-q}$  are the lagged forecast errors.

#### 2.3.2. XGBoost (Extreme Gradient Boost)

Extreme Gradient Boosting (XGBoost) is a powerful machine learning algorithm renowned for its accuracy and robustness in predictive modeling [23]. In our stock price prediction, XGBoost is a compelling choice for several reasons. First, it can handle complex, non-linear relationships in financial time series data, making it well suited for capturing intricate patterns in stock prices. Second, XGBoost can handle missing data, an occasional issue in financial datasets, through its built-in handling mechanisms. Finally, XGBoost offers flexibility in parameter tuning, enabling us to fine-tune the model's performance for our specific dataset. The XGBoost model works by building an ensemble of decision trees, where each tree corrects the errors of the previous one. These trees are combined into a strong predictive model. The algorithm assigns a weight to each tree and uses a gradient descent optimization process to minimize the prediction errors. The final prediction is a sum of predictions from all the trees [4,23]. Through this ensemble approach, XGBoost leverages the strengths of multiple decision trees to provide accurate and reliable predictions, making it a valuable asset in our stock price prediction project.

### 2.3.3. Facebook Prophet

The Facebook Prophet algorithm is an open-source time series data prediction tool developed by Facebook using the additive regression model. It is robust in identifying the components of time series data like trend and seasonality and forecasting values by combining them. It accepts only two columns ('ds' for date and 'y' for values) as the input dataset. Implementing Facebook Prophet does not require an in-depth prerequisite knowledge of time series data. It provides generalized parameters and automatically uncovers seasonal movements. The performance of FB Prophet may vary based on the dataset as it depends on seasonality and trends. Ref. [24] highlights the importance of timing in enhancing forecasting accuracy, which is accomplished with the use of the prophet algorithm. This journal uses the Facebook Prophet library to define three different hyperparameters, namely seasonality, trend, and holidays.

Having established the theoretical foundations and the rationale behind the selection of our computational models, the next section delves into the preliminary considerations and the detailed development of our methodological approach. This includes data collection, preprocessing techniques, model training, and evaluation metrics, providing a comprehensive overview of how we operationalize the theoretical insights discussed here.

### 2.4. Other Recent Advancements in the Area

Yun et al. [25] improve stock price prediction by using genetic algorithms to optimize feature subset selection. The authors maximize feature subset selection by combining genetic algorithms with machine learning regressions, improving the interpretability and precision of stock price predictions. This method stands out in particular for how well it strikes a balance between interpretability and model complexity. There are a few restrictions on the study, though. The arbitrary selection of external factors and technological indicators may have impacted the accuracy of the prediction. Furthermore, the study does not completely account for the social environment of stock market dynamics, which includes market news and public opinion, and it lacks clear criteria for feature selection.

The application of Bi-Directional Long Short-Term Memory (Bi-LSTM) networks for stock price prediction is examined by the authors in [26]. This method has the advantage of analyzing data sequences both forward and backward, which may highlight patterns and trends that conventional models would miss. The outcomes demonstrate that Bi-LSTM models have the potential to outperform conventional LSTM models, particularly when managing the volatility of stock market data. The authors do, however, also note that it is possible that the Bi-LSTM model will not be able to adequately capture the intricate and erratic character of market movements. They contend that in order to make the model more reliable and strong for everyday application, more testing and modification are required.

In their publication, Zhao and Yang [27] provide a thorough method for predicting changes in stock prices through the integration of many deep learning models. In order to take advantage of the temporal and geographical characteristics of financial data, the authors suggest a framework that blends many neural network designs, including CNNs and LSTM models. The goal of this integrated strategy is to increase prediction accuracy by identifying the intricate relationships that influence changes in stock prices. The study shows that when it comes to stock price direction prediction, the integrated framework performs better than conventional machine learning models. Though the study's findings are encouraging, it also draws attention to issues with computational complexity and the requirement for huge datasets in order to properly train these deep learning models. The practical use of the framework may be limited, particularly for smaller enterprises or individual investors, due to its heavy reliance on large amounts of data and computer resources.

In the study [28], a multi-layered long short-term memory (LSTM) network is used to present a novel technique for enhancing stock price prediction. In order to improve the LSTM architecture's capacity to identify long-term dependencies in financial time series data, the authors concentrate on optimizing it. The study successfully tackles the difficulties of predicting stock prices, which are essentially volatile and non-linear, by



employing a sequential LSTM approach. The findings show that our improved LSTM model outperforms more conventional approaches in terms of prediction, especially when it comes to identifying the complex patterns of stock price fluctuations. The model's reliance on substantial computational resources and extensive, high-quality datasets is one of the study's acknowledged potential shortcomings.

A dedicated recurrent neural network (RNN) architecture designed for time series data is introduced by Lu and Xu [29] with a focus on stock price prediction. The temporal relationships and non-linear patterns of financial data, among other difficulties, are well handled by the TRNN model. The TRNN outperforms conventional RNN architectures in terms of predicted accuracy and processing overhead by incorporating particular approaches. This paper makes a strong argument for the use of sophisticated RNN models in stock price prediction by highlighting the significance of customized neural network designs in financial forecasting.

### 3. Preliminary Considerations and Development of the Approach

In this section, we lay the groundwork for our stock price prediction models, detailing the data collection, preprocessing, and model evaluation methodologies.

#### 3.1. Historical Context and Progression

Stock price prediction is one of the most challenging applications in financial studies due to the complex nature of stock price time series. Numerous factors, including historical time series records, key technical indicators, macroeconomic variables, and investor sentiment, influence stock prices, leading to non-stationarity and non-linearity in the data. Artificial neural networks (ANNs) and, particularly, deep learning (DL) methods can be advantageous in predicting future stock prices and aid investors in reducing investment risk.

The pioneering study in applying ANNs for forecasting stock prices dates back to White's work in 1988, where he developed a standard feedforward single hidden layer architecture to predict IBM's stock prices [11]. Although this study had drawbacks, such as the overfitting problem, it opened avenues for more advanced models, such as recurrent neural networks (RNNs).

In machine learning models, input data points are transformed into outputs through a learning process derived from exposure to existing input–output pairs. The main step in ML or DL is to transform the data meaningfully. In ANNs, the learning process is done by building a set of layers where information is fed to the first layer (input layer) and passes through subsequent layers until purified information is produced. The depth of the network refers to the number of layers contributing to the structure of the model. Deep learning occurs when the number of layers is substantial.

Feedforward Neural Networks and recurrent neural networks are two main types of neural networks. While the former involves information flowing from the input layer to the output layer, the latter includes at least one cyclic path of synaptic connections. The neurons in RNNs not only use the inputs to the neuron, but also use the outputs from the previous time steps. Hence, RNNs are suitable for sequential data such as time series. Long short-term memory (LSTM) and Gated Recurrent Units (GRUs) are two types of RNNs designed to address the vanishing gradient problem during network training by the backpropagation algorithm through time.

#### 3.2. Data Collection, Exploration, and Preparation

Stock market data can be fascinating to study, and excellent predictive models can result in significant financial gains. Finding a large, well-structured dataset on a diverse set of companies can be challenging despite the seemingly limitless availability of financial data on the internet. The dataset for this study is accessed from the API of Yahoo Finance, which is often used as a reliable source of financial data. Yahoo Finance provides a comprehensive collection of financial data that includes stock prices, indices, ETFs, mutual funds, bonds,

and options worldwide. In addition, it offers extensive historical data, some going back many decades. This is especially useful for long-term financial analysis and historical research. We utilized stock data from Apple (AAPL), Amazon (AMZN), Google (GOOG), and Microsoft (MSFT) from the past ten years, sourced from the Yahoo Finance database. These stocks were chosen to leverage the findings of this study in building effective price forecasting algorithms to aid investment decisions. Exploratory data analysis (EDA) will be employed to gain a better understanding of the basic characteristics and nature of the collected dataset, including data visualization.

### 3.2.1. Train and Test Split

The dataset will be split into a training set and a test set in an 80:20 ratio; the training set will be used to train the models, while the test set will be used to evaluate their performance. The 80/20 split is commonly used because it often provides a good balance, allowing the model to learn from a large portion of the data while reserving enough data for testing. The data were split into training and testing sets using k-fold cross-validation to guarantee the models' robustness and generalizability. This method offers a more accurate approximation of the model's performance on unknown data while also assisting in the reduction of overfitting.

### 3.2.2. Data Shaping for LSTM and GRU Models

LSTM and GRU models require data structured into time steps or look-back periods. For this study, both the training and testing datasets are structured with a 60-day look-back period (60 time steps). Consequently, the models will use the last 60 days of data to predict current or future stock prices.

### 3.3. Preprocessing and Normalization

Normalization is a common approach for preparing data for machine learning, often included as part of data cleansing. The primary goal of normalization is to scale all attributes consistently. This makes it easier to discuss the performance and training stability of the model. When the ranges of the features differ, normalization is necessary. There are several approaches to normalization, also known as rescaling, including the following:

- (i) Min-Max normalization: this technique scales a feature to fit within a specific range, usually 0 to 1, according to the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (14)$$

where  $x$  is the original value, and  $\min(x)$  and  $\max(x)$  are the minimum and maximum values in the dataset, respectively.

- (ii) Mean normalization: this method adjusts the data based on the mean and can be computed as the following:

$$x' = \frac{x - \text{average}(x)}{\max(x) - \min(x)} \quad (15)$$

where the mean of the dataset ( $\text{average}(x)$ ) is used.

- (iii) Z-score normalization: also known as standardization, this approach uses the Z-score or standard score and is often utilized in machine learning algorithms like Support Vector Machines (SVMs) and logistic regression. It can be calculated using the following formula:

$$z = \frac{x - \mu}{\sigma} \quad (16)$$

where  $\mu$  is the mean and  $\sigma$  is the standard deviation of the dataset.

Given the wide range and high volatility of the volume and turnover elements in this study, we employ *Min-Max* normalization to scale all attributes between 0 and 1.

### 3.4. Model Evaluation

The model's performance will be assessed using the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Directional Accuracy (MDA), and the coefficient of determination ( $R^2$ ) [30]. As the MAE and RMSE values decrease, it becomes easier to predict how close the predicted value will be to the actual value. The model's fit is expected to be better as the coefficient of determination ( $R^2$ ) approaches one. Mean Directional Accuracy (MDA) is generally used to evaluate the model's ability to predict the direction of change rather than the magnitude of the forecasting error. The formula for RMSE, MAE,  $R^2$ , and MDA are shown below.

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

where  $y_i$  and  $\hat{y}_i$  are the actual and forecasted values, respectively, and  $N$  is the total number of observations.

$$MAE = \frac{1}{N} \sum_{j=1}^N |x_j - \hat{x}_j| \quad (18)$$

where  $x_j$  and  $\hat{x}_j$  are the actual and forecasted values at time  $j$ .

$$R^2 = \frac{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y}_i)^2} \quad (19)$$

where  $\bar{y}$  is the mean of the actual values.

$$MAD = \frac{1}{N} \sum_{j=1}^N 1_{\text{sign}}(x_j - x_{j-1}) \quad (20)$$

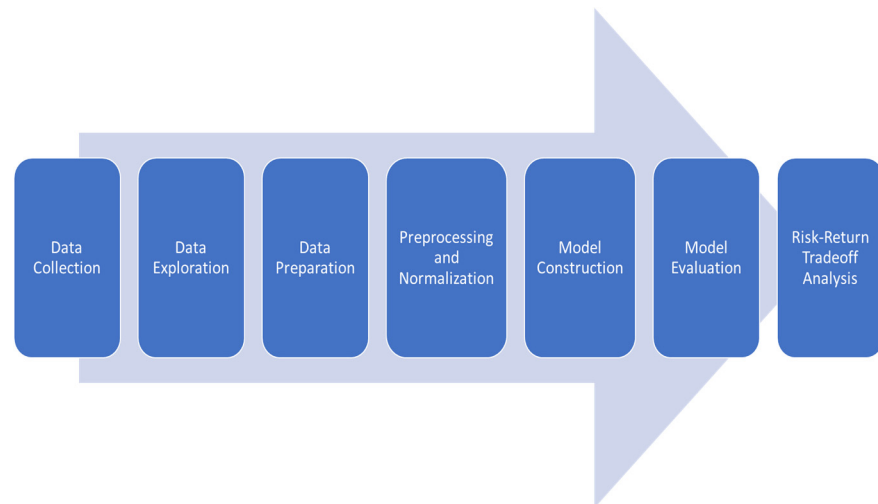
where  $N$  is the total number of observations (trading days), and  $x_j$  and  $x_{j-1}$  are the actual and forecast values, respectively.

The methodology framework is developed as follows (Figure 6). In summary, we start with data collection, gathering historical stock price data and financial indicators from Yahoo Finance for companies such as Apple, Amazon, Google, and Microsoft. This is followed by data exploration, where we perform exploratory data analysis (EDA) to understand the dataset's characteristics and trends. Next, we prepare the data by splitting it into training (80%) and testing (20%) sets, employing k-fold cross-validation to ensure robustness. Pre-processing and normalization are then applied, using techniques like Min-Max, Mean, and Z-score normalization to make the data suitable for model training. For model construction, we develop long short-term memory (LSTM) and Gated Recurrent Unit (GRU) models, as well as XGBoost and Facebook Prophet, for machine learning approaches to predict future stock prices. The models' performance is evaluated using metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Directional Accuracy (MDA), and the coefficient of determination ( $R^2$ ) to assess accuracy and effectiveness. Finally, we conduct a risk–return tradeoff analysis to examine the predicted stock prices in terms of risk and return, aiding investment decisions. This comprehensive and systematic approach ensures the development and evaluation of effective stock price prediction models, enhancing the accuracy of financial forecasts and supporting informed investment choices.

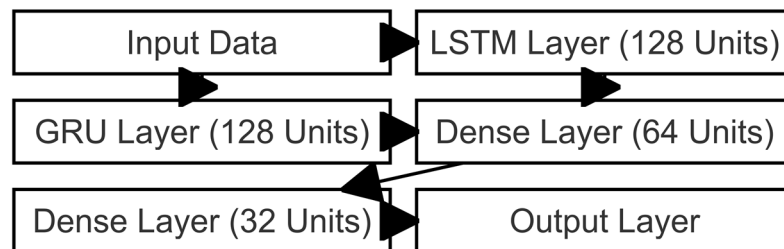
We now proceed to evaluate its performance through comprehensive numerical results and analyses in the next sections.

### 3.5. The Architectural Diagram

The architectural diagram for processing and analyzing data is presented in Figure 7 with the explanations as follows.



**Figure 6.** The methodology framework for this research.



**Figure 7.** The architecture diagram for processing and analyzing data.

**Input Layer:** Time series data, such as stock prices over a given look-back period (e.g., 60 time steps), make up the input data. A minimum of 4 GB memory is expected.

**Layer of LSTM (128 Units):** With 128 units, the LSTM layer is the first hidden layer. The LSTM layer keeps a memory of prior inputs over numerous time steps, which allows it to identify long-term dependencies in the time series data. This aids in seeing patterns that might not be obvious at first but are essential for precise forecasting.

**GRU Layer (128 Units):** After that, the output of the LSTM layer is sent to a GRU layer, which has 128 units as well. Similar to the LSTM, the GRU layer is more computationally efficient overall yet may still capture dependencies across time. By combining the benefits of both recurrent unit types, LSTM and GRU layers improve the model's capacity to identify intricate temporal patterns in the data.

**Dense Layer (64 Units):** The following layer consists of 64 units and is dense (completely connected). In order to create a more condensed representation that will be utilized to produce the final prediction, this layer processes the output from the GRU layer.

**Dense Layer (32 Units):** The features retrieved by the earlier layers are further refined by a second dense layer with 32 units, which reduces the amount of data that will go into the final forecast.

**Layer of Output:** The prediction, usually the stock price for the following time step or day, is provided by the output layer, which is the last layer. This layer has a linear activation function, which is common for regression tasks like stock price prediction and includes a single unit for predicting a single value.

## 4. Numerical Results

### 4.1. Data Preprocessing and Exploratory Analysis

This study collected daily historical stock datasets for Apple, Google, Microsoft, and Amazon stocks using the API of Yahoo Finance. The selected stocks are from international public companies traded at both NASDAQ and the NYSE. The time series data range from

1 January 2013 to 30 March 2022, encompassing 3775 trading days. The daily time series data was downloaded automatically using Python's connection to the Yahoo Finance API. Daily open price, daily highest price, daily lowest price, daily close price, daily adjusted closing price, and daily trading volume are all included in the dataset. Table 1 below presents the description of the features provided in the datasets downloaded from Yahoo Finance.

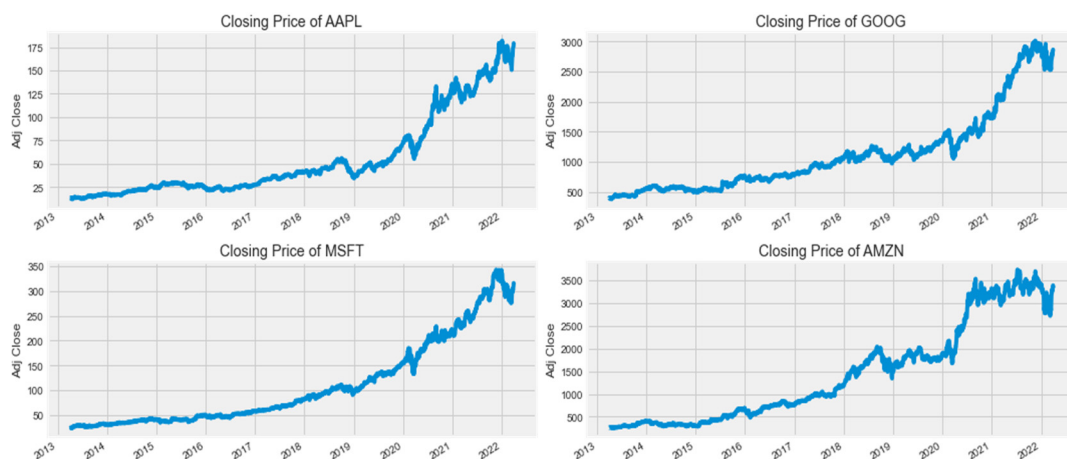
**Table 1.** Description of the features provided in the datasets.

FEATURE	DESCRIPTION
OPEN PRICE	The price at which a stock was initially traded at the start of a trading day.
CLOSE PRICE	The last price of a stock in the last transaction on a given trading day.
HIGH PRICE	The highest price at which a stock traded on a specified trading day.
LOW PRICE	The lowest price at which a stock traded on a specified trading day.
ADJUSTED CLOSE PRICE	Adjusted close price based on the reflection of dividends and splits.
TRADING VOLUME	A total number of shares/contracts traded on a given trading day.

The closing data were used to compute the daily returns for each technological stock used to train the models. The most straightforward and obvious way to understand the stock trend is through the characteristics of the stock price. Compared to the absolute value of stock prices, price trend returns are more effective in stock forecasting. Different stocks have different base prices, leading to large variations in absolute stock price values. Using daily returns reduces the prediction's sensitivity to the price base.

For the training and testing of the model, the data were split into training and test sets, with 80% of the total data used for training the model and the remaining 20% used for testing.

Figure 8 below shows the closing price line chart for the selected technological stocks, providing a quick overview of the collected data.



**Figure 8.** The closing price of the technological stock prices selected.

In order to develop a better understanding of the technological stock price data used in this study, summary statistics for all the features were computed and are presented in Appendix A.

Based on the summary statistics provided in the tables above, for all four companies, all features appear to be right-skewed, as the means are consistently higher than the medians, suggesting an upward trend in stock prices over time. In addition, the high standard deviations of the features, especially for the adjusted closing price and trading volume, indicate that the stock prices and trading volumes of these companies were more volatile during the reporting period.

#### 4.2. Hyperparameter Selection Process

The process of selecting the best collection of hyperparameters for a model is known as hyperparameter tuning. Variables that can be adjusted during this optimization process include the number of units, batch size, learning rate, and dropout rate.

1. Units: The optimization strategy sets the number of units in each LSTM and GRU model to 128 and 64 in the first and second layers, respectively.
2. Batch size: For tuning the model, the batch size is set to 1.
3. Learning rate: The learning rate of the Adam optimizer is set at 0.1.
4. Dropout layer: During model training, it is common to observe a pattern where the model performs well on the training data but fails to replicate this success on the testing and validation data. This discrepancy, often due to overfitting, is a major concern, especially in deep learning models that require a substantial amount of data for training. Dropout is a simple but effective regularization strategy used in neural networks to mitigate this overfitting problem. The cells of the recurrent neural network are dropped at random. The dropout rate is around 0.2.

The model's training may be excessive or insufficient. Early stopping criteria are often used to prevent complications caused by having too many or too few epochs. These criteria allow for the creation of a large number of training epochs and then stopping the training when the model's parameters no longer improve on the validation set.

The full specification of the parameters used to train the model is provided in Table 2.

**Table 2.** Model Training Parameter Specifications.

PARAMETER	VALUES
NODES WITHIN INPUT LAYER	look-back period $\times$ input features
STEPS	2026 with early stopping criteria of the patience of 1 epoch
BATCH SIZE	1
HIDDEN LAYER	1 LSTM/GRU layer with 128 units
DROP OUT LAYER	0.2 dropout rate
LOOK-BACK PERIOD	60
OUTPUT LAYER	1

#### 4.3. Results of the Models

This section contains the performance of the deep learning models (LSTM and GRU) for each of the stock prices considered, as well as the technical indicators (Table 3). The Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), coefficient of determination ( $R^2$ ), and Mean Absolute Difference (MAD) are used to evaluate the performance of the models. In addition, the performance of the models is compared to the other deep learning and traditional forecasting methods reported in the literature (Table 4). Our study showed that the GRU model generally outperformed the LSTM model across multiple metrics such as RMSE and MAE, with the GRU achieving an RMSE of 3.43 and an MAE of 6.53 for Apple stock, which is notably lower than the LSTM's RMSE of 9.15 and MAE of 7.81. When compared to other models from existing studies, such as the S-GAN model, which reported an RMSE of 1.83 on Apple stock, our models still indicate room for improvement. The consideration of investigator sentiment enhanced the prediction capability of S-GAN [12]. Additionally, the ARIMA model from the literature showed an RMSE of 18.25, indicating that our GRU model offers significant advancements over traditional methods [12]. However, the performance of the LSTM model is less competitive than other LSTM models in the literature [31,32]. These comparisons highlight that while our GRU model is competitive, especially with respect to more traditional approaches, there is potential for further performance enhancement by integrating more advanced techniques and conducting extensive hyperparameter optimization.

**Table 3.** Model performance for stocks.

MODELS	RMSE	MAE	R <sup>2</sup>	MAD	TRAINING TIME (SECS)	STOCK
LSTM	9.1463	7.8058	0.7609	6.0689	54	Apple
GRU	3.4273	6.5298	0.8229	6.7607	51	Apple
LSTM	103.5552	61.7088	0.4679	107.5323	57	Google
GRU	67.4582	35.1966	0.7742	115.7959	52	Google
LSTM	32.3734	31.045	0.6998	13.8247	63	Microsoft
GRU	8.0805	5.2005	0.8319	6.6803	61	Microsoft
LSTM	116.948	5.8673	0.7479	191.7697	64	Amazon
GRU	82.9599	3.8673	0.8731	182.9617	61	Amazon

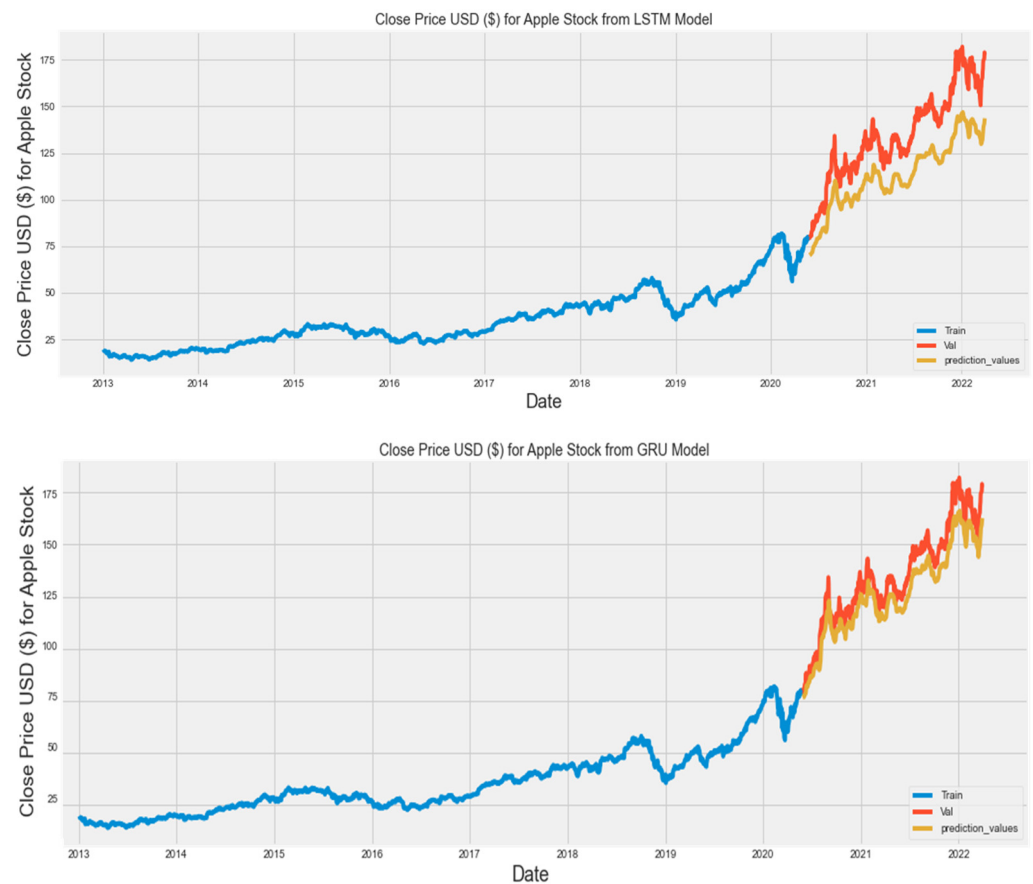
**Table 4.** Comparison of Model Performance Against State-of-the-Art Methods Reported in the Literature.

MODELS	RMSE	STOCK
LSTM	9.1463	Apple
GRU	3.4273	Apple
LSTM	103.5552	Google
GRU	67.4582	Google
LSTM	32.3734	Microsoft
GRU	8.0805	Microsoft
LSTM	116.948	Amazon
GRU	82.9599	Amazon
LSTM [25]	18.89.	Apple (AAPL), Google (GOOG), Microsoft (MSFT), and Amazon (AMZN)
S-GAN [26]	1.827	APPLE
ARMIA [26]	18.2469	APPLE
LSTM [27]	6.59	Apple (AAPL), Google (GOOG), Microsoft (MSFT), and Amazon (AMZN)
Ridge [27]	8.72	Apple (AAPL), Google (GOOG), Microsoft (MSFT), and Amazon (AMZN)
Neural Network [27]	7.91	Apple (AAPL), Google (GOOG), Microsoft (MSFT), and Amazon (AMZN)

#### 4.3.1. Apple Stock Prediction

Using the daily historical stock datasets for Apple Inc. (Cupertino, CA, USA), along with the technical indicators, the performance of LSTM and GRU models was assessed with the Mean Absolute Error (MAE), the Root Mean Square Error (RMSE), the Mean Absolute Deviation (MAD), and the coefficient of determination (R<sup>2</sup>). The results are presented in Table 3.

From Table 3, it can be observed that the GRU model forecasts the Apple stock price more accurately, as the RMSE and MAE values are considerably lower for the GRU model (3.4273 and 6.5298, respectively) than LSTM model (9.1463 and 7.8058, respectively). Additionally, the R<sup>2</sup> value is higher for the GRU model (0.8229) than for the LSTM model (0.7609), suggesting a better fit. It should also be observed that the GRU model has a shorter training time than the LSTM model. Figure 9 below depicts the pattern of the actual closing prices and predicted closing prices for the LSTM and GRU models.



**Figure 9.** Apple stock: actual and predicted close price, LSTM and GRU models.

#### 4.3.2. Google Stock Prediction

Using the daily historical stock datasets for Google Inc. (Mountain View, CA, USA), along with technical indicators, the performance of the LSTM and GRU models was assessed using MAE, RMSE, MAD, and  $R^2$ .

From Table 3, it can be observed that the GRU model makes more accurate stock price predictions for Google, with lower RMSE and MAE values (67.4582 and 35.1966, respectively), compared to the LSTM model (103.5552 and 61.7088, respectively). The  $R^2$  value is also higher for the GRU model (0.7742) than for the LSTM model (0.4679). The GRU model also has a shorter training time. Figure 10 below depicts the pattern of the actual closing prices and predicted closing prices for the LSTM and GRU models.

#### 4.3.3. Microsoft Stock Prediction

The performance of the LSTM and GRU models for Microsoft stock, as measured by MAE, RMSE, MAD, and  $R^2$ , is summarized in Table 3.

Table 3 indicates that the GRU model forecasts Microsoft stock prices more accurately, with significantly lower RMSE and MAE values (8.0805 and 5.2005, respectively) compared to the LSTM model (32.3734 and 31.0450, respectively). The GRU model also shows a better fit ( $R^2 = 0.8319$ ) and has a shorter training time than the LSTM model. Figure 11 shows the evolution of actual and expected closing prices for the LSTM and GRU models. Furthermore, the GRU model needs less training time than the LSTM model.

#### 4.3.4. Amazon Stock Prediction

The performance of the LSTM and GRU models for Amazon stock, as measured by MAE, RMSE, MAD, and  $R^2$ , is summarized in Table 3.

From Table 3, it is evident that the GRU model predicts Amazon stock prices more accurately, with lower RMSE and MAE values (82.9599 and 3.8673, respectively) compared



to the LSTM model (116.9480 and 5.8673, respectively). The  $R^2$  value is higher for the GRU model (0.8731) than for the LSTM model (0.7479). The GRU model also has a shorter training time. Figure 12 below depicts the actual closing prices and predicted closing prices for the LSTM and GRU models.

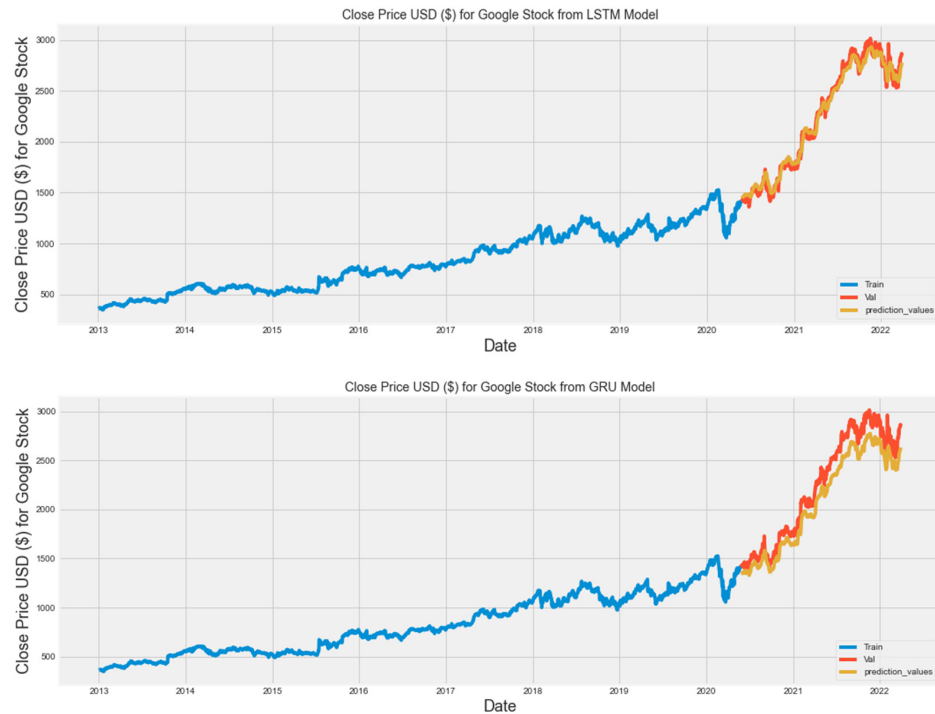


Figure 10. Google stock: actual and predicted close, LSTM and GRU models.

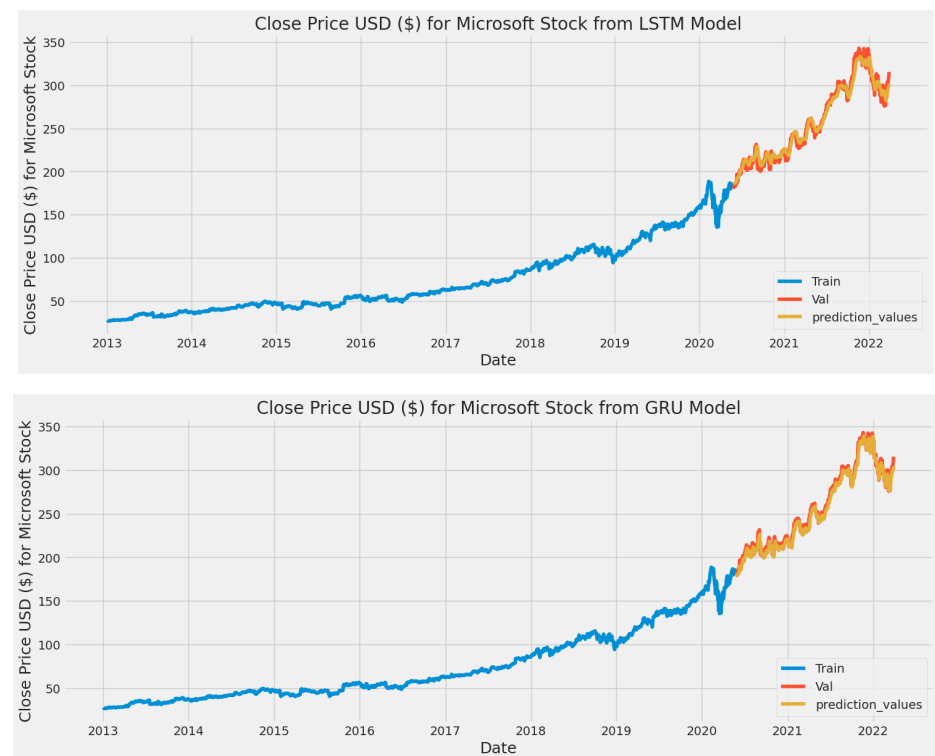
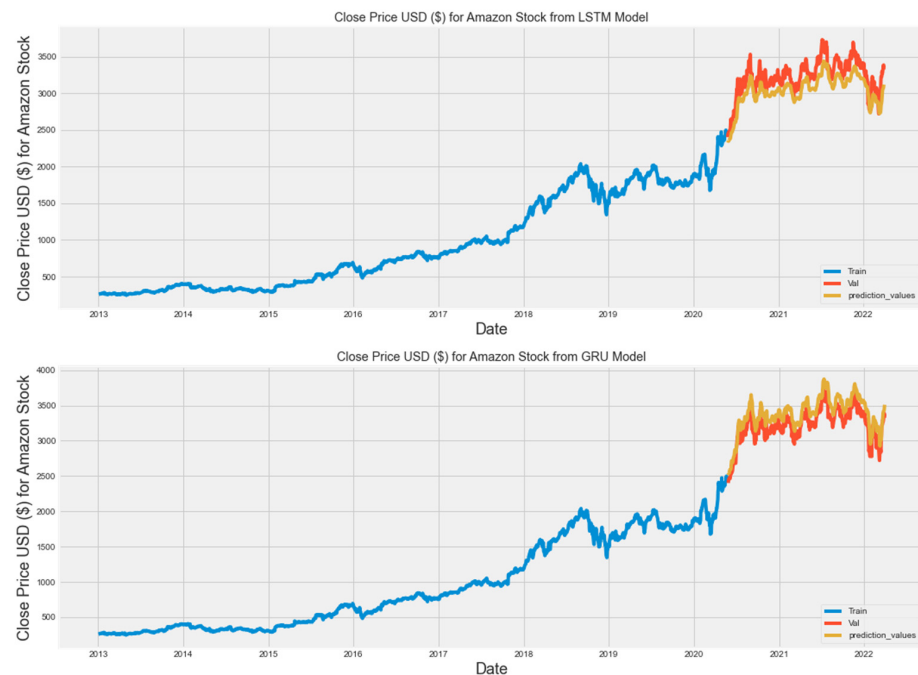


Figure 11. Microsoft stock: actual and predicted close price, LSTM and GRU models.



**Figure 12.** Amazon stock: actual and predicted close price, LSTM and GRU models.

In all the examined cases, the Gated Recurrent Unit (GRU) model not only demonstrated superior forecasting accuracy but also trained faster. This dual advantage illustrates the GRU model's efficiency and effectiveness in predicting stock prices, a crucial aspect of market investments. However, there is an interesting nuance worth considering. For Apple and Google stock, the GRU model has a slightly higher Mean Absolute Deviation (MAD), which indicates the average distance between each data point and the mean. While the GRU model's predicted averages are closer to the actual values, some individual predictions deviate more from the actual values compared to the long short-term memory (LSTM) model. The slight increase in MAD may be due to the inherent variability of stock prices. Different stocks have different characteristics resulting from a company's market behaviors, such as trading volume, market sentiment, market or company events, and financial performance, resulting in higher volatility for certain stocks. Therefore, the slightly higher MAD of the GRU model does not necessarily imply a lack of predictive power but may reflect the nature of the data it is dealing with. To conclude, the GRU model appears to be more suitable for firms to use as a stock price forecasting tool, given its overall advantages in terms of forecasting accuracy and time efficiency. However, it is recommended to consider individual forecasting biases when dealing with stocks with high price volatility.

#### 4.4. Predicted Risk–Return Tradeoff

A risk–return tradeoff plot was created to link the predicted stock prices from the GRU model with effective decision-making. This plot visually represents the model's performance by connecting the risks from predicted returns among the stock prices. It visualizes these tradeoffs for the four technology stocks considered in this study: Apple, Google, Microsoft, and Amazon. The risk–return tradeoff plot is presented in Figure 13 below.

As observed from the predicted risk–return tradeoff plot presented above, there is a positive relationship between risk and expected returns for each of the four technology stocks considered in this study, aligning with the foundational principle of finance that higher returns usually come at the cost of higher risk. However, there are disparities in the results of tradeoff analyses of these technology giants. Investing in Google stock is the most conservative investment, with the lowest risk and lowest expected returns. On the other hand, the risk associated with Amazon stock is higher than that of Apple stock. However, Apple stock predicts more expected returns despite the lower risk than

Amazon stock. This could be due to the high price of Amazon’s stock, which might result in more price volatility. Apple stock’s counter-intuitive scenario might result from market sentiment, Apple’s strong financial performance, or the company’s potential for future growth. It suggests that Apple could provide an attractive risk–return tradeoff for investors. Microsoft stock is also associated with lower risk but considerably higher expected returns when compared with Google stocks. This might reflect investor confidence in Microsoft’s business model, its diverse range of offerings, and its solid financial performance. The risk–return tradeoff chart shows that the risk and return profiles of different stocks vary even within the same industry. The analysis provides decision-makers with an effective tool to align their investment decisions with their risk appetite and return expectations.

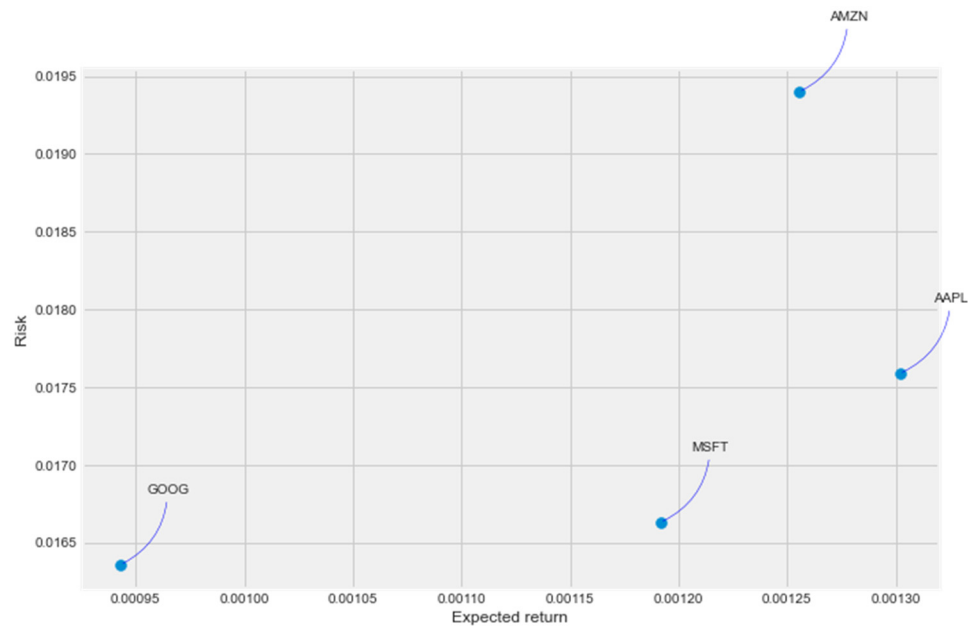


Figure 13. Predicted risk–return tradeoff plot.

### 5. Additional Experiments and Validations

Our objective is to identify the most accurate model for each of the four stock prices. The primary goal of this analysis is to construct reliable and precise forecasting models specifically tailored for short- to medium-term predictions. To ensure the consistency and reliability of these models, we conducted a validation exercise over a 30-day time horizon, starting from 1 January 2023. This timeframe simulates the intended use of these models in real-world scenarios, allowing us to assess their practical effectiveness and suitability for forecasting stock prices in a reasonable timeframe. The results are summarized in Table 5.

Table 5. Performance of four selected models on stocks.

STOCK	MODEL	MSE	MAE	RMSE
Apple	ARIMA	256.58	13.15	16.01
Apple	XGBOOST	254.24	14.61	15.94
Apple	LSTM	113.32	8.04	10.64
Apple	FB PROPHET	1355.6	31.03	36.81
Apple	GRU	95.23	7.41	9.76
Amazon	ARIMA	1194.5	28.56	34.56
Amazon	XGBOOST	426.92	17.69	20.66
Amazon	LSTM	240.69	8.6	15.51

**Table 5.** *Cont.*

STOCK	MODEL	MSE	MAE	RMSE
Amazon	FB PROPHET	5819.41	70.9	76.28
Amazon	GRU	210.47	7.96	14.51
Google	ARIMA	813.05	24.17	28.51
Google	XGBOOST	56.22	5.94	7.49
Google	LSTM	82.59	4.84	9.08
Google	FB PROPHET	4172.36	59.34	64.59
Google	GRU	73.21	4.53	8.56
Microsoft	ARIMA	1940.95	38.09	44.05
Microsoft	XGBOOST	239.23	13	15.46
Microsoft	LSTM	278.49	10.39	16.68
Microsoft	FB PROPHET	9446.06	88.19	97.19
Microsoft	GRU	249.92	9.87	15.81

### 5.1. Performance of Four Selected Models on Apple Stock

The Root Mean Square Error (RMSE) is our primary metric for assessing the accuracy of forecasting models. As shown in Table 5, the RMSE scores were 10.64 for LSTM, 15.94 for XGBoost, 16.01 for ARIMA, and 36.81 for Facebook Prophet. A lower RMSE signifies better predictive performance, indicating that LSTM and XGBoost outperformed ARIMA and Facebook Prophet. LSTM achieved the lowest RMSE, suggesting it was the most accurate in capturing Apple's stock price trends, followed closely by XGBoost. ARIMA and Facebook Prophet had higher RMSE scores, implying they struggled to capture stock price fluctuations effectively. Nevertheless, model selection should consider other factors like computational complexity and suitability for the specific forecasting task.

### 5.2. Performance of Four Selected Models on Amazon Stock

In the case of Amazon stock predictions, the RMSE scores were 15.51 for LSTM, 20.66 for XGBoost, 34.56 for ARIMA, and a notably higher 76.28 for Facebook Prophet. As shown in Table 5, a lower RMSE signifies better predictive accuracy, and here, LSTM exhibited the lowest RMSE, indicating its superior ability to capture Amazon's stock price trends. XGBoost also performed well, with a relatively low RMSE. In contrast, ARIMA had a higher RMSE, suggesting it struggled to effectively capture stock price movements. Facebook Prophet, with the highest RMSE, appears to have had the most difficulty in accurately forecasting Amazon's stock prices.

### 5.3. Performance of Four Selected Models on Google Stock

In the context of Google stock predictions, the RMSE scores were 9.08 for LSTM, 7.49 for XGBoost, 28.51 for ARIMA, and a substantially higher 64.59 for Facebook Prophet. As shown in Table 5, a lower RMSE score indicates better predictive accuracy, and in this case, both LSTM and XGBoost delivered commendable results with low RMSE values, suggesting their effectiveness in capturing Google's stock price trends. In contrast, ARIMA exhibited a higher RMSE, indicating it struggled to predict the stock price movements accurately. Facebook Prophet, with the highest RMSE, seems to have faced significant challenges in providing accurate forecasts for Google stock.

### 5.4. Performance of Four Selected Models on Microsoft Stock

The RMSE scores obtained were 16.68 for LSTM, 15.46 for XG Boost, 44.05 for ARIMA, and the score was notably higher at 97.19 for Facebook Prophet. As shown in Table 5, lower RMSE values indicate better predictive accuracy, and in this case, both LSTM and

XGBoost demonstrated a relatively strong performance, with low RMSE scores, suggesting their effectiveness in capturing Microsoft's stock price trends. ARIMA, on the other hand, exhibited a higher RMSE, indicating some difficulty in accurately predicting the stock price movements. With the highest RMSE, Facebook Prophet has faced substantial challenges in providing accurate forecasts for Microsoft stock.

### *5.5. Discussion: Forecasting Accuracy*

Advanced machine learning techniques have yielded a collection of top-performing models, each with unique strengths in providing predictive insights. These models consistently demonstrate their expertise in delivering directional accuracy, enabling us to grasp the general trends in stock price movements. Furthermore, they excel in generating predicted values that closely mirror actual stock prices, highlighting their proficiency in capturing the intricate patterns hidden in the financial markets.

Among these models, the XGBoost model stands out as the best model, with the highest accuracy, and it is a symbol of predictive power. Its exceptional precision in forecasting Google's stock price, in particular, underscores the robustness and finesse of the XGBoost algorithm in decoding the complexities inherent in the stock market. It has earned its place as the most accurate model in our extensive analysis, leaving an indelible mark on our quest for precision in stock price forecasting.

Our models can predict the general direction of stock movements, but they have trouble capturing unexpected, occasionally unanticipated market changes brought on by uncertainties and speculations, which motivate us to investigate further under what conditions we can rely on these models with certainty and be solely satisfied with their accuracy. Interestingly, our expedition has revealed some issues due to the inconsistency in various accuracy metrics. This variability serves as a reminder of the multifaceted nature of stock market predictions and emphasizes the importance of evaluating results from multiple angles. The absence of a uniform standard for assessment underscores the necessity of a comprehensive evaluation approach.

Furthermore, it is crucial to note that achieving forecasting excellence yields a variety of solutions. Each model is unique across all scenarios and within every scoring metric. Choosing the most suitable model becomes a subjective decision, dependent on the user's specific objectives and preferences. For example, investors focusing solely on directional accuracy may favor one model, while those concerned about short-term fluctuations may lean toward a different one. Thus, the adaptability of model selection to align with predefined goals becomes important in our quest for precision.

To sum it up, our exploration into forecasting accuracy has revealed valuable insights. Our models are effective in indicating stock price trends and are generally accurate. However, they struggle with the unpredictability of the market. Hence, we suggest that users take a balanced approach, consider models from different perspectives, and exercise caution when using them in the ever-changing world of stock market predictions.

### *5.6. Implications of This Research*

This work advances the field of deep learning-based stock price prediction in several significant ways. First, it fills a vacuum in the existing literature by conducting a targeted comparison analysis of LSTM and GRU models, specifically for the technology industry. Second, the models built here offer more precision than those trained on general stock market data, thanks to the incorporation of industry-specific knowledge. Finally, the study offers investors useful advice by determining the best model for technology stock predictions. These contributions build on earlier research by highlighting the significance of industry-specific modeling techniques and their potential to enhance investment decision-making.

The implications of this study are not just theoretical but have significant economic consequences. Robust stock price forecasting models, such as the GRU model discussed, can empower investors to make more informed decisions, potentially improving portfolio performance. By focusing on the technology sector, this study provides insights into a

field that has been a key driver of economic growth. The potential for improved technology stock price forecasting to enhance market efficiency and capital allocation is a compelling prospect.

## 6. Discussions

### 6.1. Contributions

In this paper, we clearly demonstrate that we have made three major contributions to this research.

First, we developed machine learning (ML) frameworks for social, economic, and demographic prediction, as we have developed ML models to perform accurate analysis and predictions for selected stock prices and the risks associated with them. Modeling stock prices provides crucial insights into the dynamics of financial markets, with profound implications for the economy and society at large. Stock markets essentially represent public expectations of corporate growth and economic health. Advanced forecasting fuels data-driven decision-making, risk assessment, and policy actions that shape social outcomes [33].

A second contribution is our use of big data and data sources for digital and computational analysis, since we have used a big data approach to analyze stock market prices and predictions and investigate their relations to the US market and its economy. This research implemented machine learning on an extensive dataset of 3775 daily observations across four major technology stocks over ten years. The data-intensive modeling approaches demonstrate the power of modern computational statistics to uncover complex patterns in economic time series data [34].

Third, we used deep learning for stock prediction, as the primary goal of this study was to employ deep learning, AI, and machine learning methods like the recurrent neural network (RNN) to accurately anticipate the pattern of future stock prices in the technology sector. We used daily technology stock data and basic technical indicators and compared LSTM and GRU models, which belong to the RNN family, to ascertain which of them is more efficient in predicting stock prices of technology industries. To achieve this aim, this study collected daily historical stock datasets for Apple, Google, Microsoft, and Amazon stocks from the API of Yahoo Finance through the Python 'pandas\_datareader.data' and Yahoo Finance library. The stocks selected are for international public companies traded at both NASDAQ and the NYSE. The time series data range from 1 January 2013 to 30 March 2022. Together, the series contains 3775 trading days. The daily time series data were automatically downloaded because Python is connected to the Yahoo Finance API. The dataset includes daily prices: open, highest, lowest, close, adjusted close, and daily trading volume.

The study applied deep learning models to analyze selected technological stock patterns as a one-dimensional time series and forecast future stock prices by examining past historical prices and the most critical technical indicators. The analysis built a comparison system to examine the performance of the LSTM and the GRU ensemble models on the selected technology stock data to identify a parsimonious model for the real-world representation of the technology stock markets.

The performances of the LSTM and GRU models were assessed using the Mean Absolute Error (MAE), the Root Mean Square Error (RMSE), the Mean Absolute Deviation (MAD), and the coefficient of determination ( $R^2$ ). From the results, it is observed that the GRU model makes it simpler to predict how close the predicted value of the Apple, Google, Amazon, and Microsoft stocks are to the actual value, as the RMSE and MAE values are considerably lower for the GRU model for all of the technology stocks than for the LSTM model. Moreover, the model's fit ( $R^2$ ) is observed to be better for the GRU model than for the LSTM model. It was also observed from our analysis that the GRU model has a shorter training time than the LSTM model. Therefore, the GRU model produced a better forecasting system for predicting daily technology stock data and fundamental technical indicators. It can be used to efficiently estimate the pattern of future stock prices within the technology industry.

Lastly, this study linked the predicted stock from the GRU model with effective decision-making. The risk–return tradeoff plot was computed as a visual depiction of the model performance to connect risks from predicted returns among the technological stock prices, and it can be observed that there is a positive relationship between risk and expected returns for each of the four technology stocks considered in this study. Investing in Google stock is associated with the lowest risk and lowest expected returns. However, the risk associated with Amazon stock is higher than that of Apple stock. However, Apple stock predicted more expected returns despite the lower risk compared with Amazon stock. Microsoft stock is also associated with lower risk, but considerably higher expected returns compared with Google stocks.

The present study has several contributions. Firstly, our study focuses on the technology sector, comparing LSTM and GRU models specifically for technology stocks like Apple, Google, Microsoft, and Amazon. This sector-specific analysis reveals unique patterns that are not seen in broader market studies, providing more useful insights for technology investors. Additionally, we evaluate not only the accuracy but also the training efficiency of the models, offering practical insights into their computational performance. Our study also includes a risk–return analysis based on predicted stock prices, giving practical insights into investment strategies. These points highlight the unique aspects of our approach and the significant contributions of our work.

### 6.2. Limitations of the Study

There are two main limitations of this study. Firstly, the primary setback experienced in the process of this study was the inadequacy of stock price data. As stated earlier in this study, a number of factors influence stock market volatility, and building an efficient machine learning model that predicts these stock prices with minimum error requires a sizeable number of attributes that are not available for many of the stocks considered. The Bureau of Labor Statistics reports that there are around 260 trading days every year, which is considered insufficient if there is the need to go far in time for more examples, such as by examining data from the last two to three years.

The other limitation is that building an effective system for stock market prediction requires a denoising process that involves adding more technical indicators, such as the daily sentiment polarity score, which will help remove human feelings for the proper estimation of future stock prices. However, this process requires complicated computation methods that could not be considered in this study due to time constraints.

### 6.3. Future Research

The study results show that the GRU model is an effective model for predicting technology stock prices among the recurrent neural network models. However, this result cannot be generalized for all other stock market data due to the lack of a sizeable amount of stock data. Therefore, it can only be concluded arbitrarily that the GRU model outperforms the LSTM model in stock forecasting. It is recommended that future studies compare these two models on a larger quantity of datasets and extend the estimation to stock price data in other industries.

Future studies should also consider focusing on building the stock price prediction system through a deep neural network that considers historical financial data, technical indicators, and financial news, and use a large volume of the training dataset to yield less prediction error. The reason is that stock price data are very volatile and often show noisy characteristics as well as non-stationary patterns. The inclusion of more technical indicators, a large volume of the training dataset, financial news, and posts can be used to denoise the data.

Finally, this study also recommends the utilization of stacked models, as this study solely compared models with each other. Future researchers could discover more by stacking models to see if they can improve prediction ability.

While our study focuses on predicting stock prices in the technology sector, the advanced deep learning (DL) and machine learning (ML) models we employ have broad applicability across various industries. For example, these models can predict patient outcomes and optimize treatment plans in the healthcare sector. The energy sector can utilize our solution to forecast consumption patterns and optimize grid operations. In retail, our models can forecast sales and manage inventory. Financial institutions can leverage these techniques for credit scoring, fraud detection, and risk management. By demonstrating the versatility of our solution, we highlight its potential to address diverse challenges across different sectors, underscoring the broad impact and utility of our approach.

## 7. Conclusions

This study has made important contributions to financial market analysis by utilizing cutting-edge machine learning and deep learning techniques. We successfully designed ML models that predict stock prices with precision and assess associated risks, offering critical insights into the factors that shape financial markets and, by extension, the broader economy. By applying big data approaches to analyze extensive historical stock data, we showcased the effectiveness of modern computational methods in revealing intricate patterns within economic datasets. Our comparison of LSTM and GRU models demonstrated that the GRU model excels in both prediction accuracy and computational efficiency, particularly within the technology sector. Additionally, the study's analysis of the risk–return relationship provided actionable insights for investors, highlighting the distinct behaviors of major technology stocks. These findings not only enhance our understanding of stock market dynamics but also provide a strong foundation for future research in financial forecasting and investment strategy development.

**Author Contributions:** Conceptualization, V.C. and A.C.; methodology, V.C.; software, A.C.; validation, V.C., Q.A.X. and H.W.; formal analysis, V.C., A.C. and H.W.; investigation, V.C. and Q.A.X.; resources, V.C.; data curation, A.C.; writing—original draft preparation, V.C. and A.C.; writing—review and editing, V.C., Q.A.X. and H.W.; visualization, V.C. and A.C.; supervision, V.C.; project administration, V.C.; funding acquisition, V.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is partly supported by VC Research (VCR 000221) and Leverhulme Trust (VP1-2023-025).

**Data Availability Statement:** Data is available upon requests. Readers can also download data from Yahoo Finance or Google Finance.

**Acknowledgments:** We thank Akash Prasad and Akram Dehnokhalaji for spending some time to help improve part of this research project.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A

**Table A1.** Summary statistics of Apple stock price features.

	OPEN	HIGH	LOW	CLOSE	ADJ CLOSE	VOLUME
COUNT	2268	2268	2268	2268	2268	2268
MEAN	57.2297	57.84394	56.63649	57.2662	55.63746	166,807,034
STD	44.14435	44.68723	43.6103	44.17467	44.78484	107,981,557
MIN	13.97714	14.29536	13.88821	14.06357	12.30091	41,000,000
25%	27.14937	27.35375	26.86813	27.14625	24.96922	95,148,700
50%	39.47	39.845	39.02875	39.36125	37.76188	133,553,600
75%	68.54375	69.69688	67.44	68.765	67.7272	202,644,200
MAX	182.63	182.94	179.12	182.01	181.7784	1,065,523,200



**Table A2.** Summary statistics of Google stock price features.

	OPEN	HIGH	LOW	CLOSE	ADJ CLOSE	VOLUME
COUNT	2268	2268	2268	2268	2268	2268
MEAN	1139.757	1150.551	1128.898	1139.985	1139.985	1,932,454
STD	674.3742	681.2976	667.1945	674.1677	674.1677	1,261,224
MIN	398.8052	400.4789	386.053	398.5611	398.5611	7922
25%	607.7186	611.8685	602.0832	607.2103	607.2103	1,222,225
50%	986.725	990.855	976.655	984.065	984.065	1,557,583
75%	1307.397	1319.705	1304.069	1313.13	1313.13	2,191,964
MAX	3037.27	3042	2997.75	3014.18	3014.18	23,219,507

**Table A3.** Summary statistics of Microsoft stock price features.

	OPEN	HIGH	LOW	CLOSE	ADJ CLOSE	VOLUME
COUNT	2268	2268	2268	2268	2268	2268
MEAN	114.5614	115.6476	113.4146	114.5935	110.6238	31,954,477.16
STD	84.94871	85.80088	84.00433	84.94511	86.28801	16,886,826.19
MIN	30.3	30.9	30.27	30.6	25.62621	7,425,600
25%	47.1775	47.6675	46.695	47.26	41.86476	22,205,600
50%	77.63	77.9	77.36	77.78	73.41168	28,010,550
75%	157.185	158.8025	156.0725	157.6175	154.3496	36,433,425
MAX	344.62	349.67	342.2	343.11	342.402	248,428,500

**Table A4.** Summary statistics of Amazon stock price features.

	OPEN	HIGH	LOW	CLOSE	ADJ CLOSE	VOLUME
COUNT	2268	2268	2268	2268	2268	2268
MEAN	1454.869	1470.281	1437.496	1454.153	1454.153	4,047,869
STD	1075.819	1088.213	1061.821	1074.703	1074.703	2,153,351
MIN	248.94	252.93	245.75	248.23	248.23	881,300
25%	482.5175	489.3	474.9075	482.1525	482.1525	2,686,300
50%	1023.14	1032.22	1016.75	1026.27	1026.27	3,464,050
75%	1949	1975.377	1931.703	1954.335	1954.335	4,693,025
MAX	3744	3773.08	3696.79	3731.41	3731.41	23,856,100

## References

1. Ariyo, A.A.; Adewumi, A.O.; Ayo, C.K. Stock Price Prediction Using the ARIMA Model. In Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, UK, 26–28 March 2014; pp. 106–112.
2. Nicholas Refenes, A.; Zapranis, A.; Francis, G. Stock performance modeling using neural networks: A comparative study with regression models. *Neural Netw.* **1994**, *7*, 375–388. [[CrossRef](#)]
3. Malki, A.; Atlam, E.-S.; Hassanien, A.E.; Ewis, A.; Dagnew, G.; Gad, I. SARIMA model-based forecasting required number of COVID-19 vaccines globally and empirical analysis of peoples' view towards the vaccines. *Alex. Eng. J.* **2022**, *61*, 12091–12110. [[CrossRef](#)]
4. Paliari, I.; Karanikola, A.; Kotsiantis, S. A comparison of the optimized LSTM, XGBOOST and ARIMA in Time Series forecasting. In Proceedings of the 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA), Chania, Crete, Greece, 12–14 July 2021; pp. 1–7.

5. Cheng, Y.; Yi, J.; Yang, X.; Lai, K.K.; Seco, L. A CEEMD-ARIMA-SVM model with structural breaks to forecast the crude oil prices linked with extreme events. *Soft Comput.* **2022**, *26*, 8537–8551. [[CrossRef](#)]
6. Chatterjee, A.; Bhowmick, H.; Sen, J. Stock Price Prediction Using Time Series, Econometric, Machine Learning, and Deep Learning Models. In Proceedings of the 2021 IEEE Mysore Sub Section International Conference (MysuruCon), Hassan, India, 24–25 October 2021; pp. 289–296.
7. Escudero, P.; Alcocer, W.; Paredes, J. Recurrent Neural Networks and ARIMA Models for Euro/Dollar Exchange Rate Forecasting. *Appl. Sci.* **2021**, *11*, 5658. [[CrossRef](#)]
8. Liang, F.; Liang, F.; Zhang, H.; Zhang, H.; Fang, Y.; Fang, Y. The Analysis of Global RMB Exchange Rate Forecasting and Risk Early Warning Using ARIMA and CNN Model. *J. Organ. End User Comput. (JOEUC)* **2022**, *34*, 1–25. [[CrossRef](#)]
9. Rao, A.R.; Reimherr, M. Modern non-linear function-on-function regression. *Stat. Comput.* **2023**, *33*, 130. [[CrossRef](#)]
10. Thakkar, A.; Chaudhari, K. A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. *Expert Syst. Appl.* **2021**, *177*, 114800. [[CrossRef](#)]
11. White, H. Economic prediction using neural networks: The case of IBM daily stock returns. In Proceedings of the IEEE 1988 International Conference on Neural Networks, San Diego, CA, USA, 24–27 July 1988; Volume 452, pp. 451–458.
12. Sonkiya, P.; Bajpai, V.; Bansal, A. Stock price prediction using BERT and GAN. *arXiv* **2021**, arXiv:2107.09055.
13. Maqsood, H.; Mehmood, I.; Maqsood, M.; Yasir, M.; Afzal, S.; Aadil, F.; Selim, M.M.; Muhammad, K. A local and global event sentiment based efficient stock exchange forecasting using deep learning. *Int. J. Inf. Manag.* **2020**, *50*, 432–451. [[CrossRef](#)]
14. Patil, P.; Wu, C.-S.M.; Potika, K.; Orang, M. Stock Market Prediction Using Ensemble of Graph Theory, Machine Learning and Deep Learning Models. In Proceedings of the 3rd International Conference on Software Engineering and Information Management, Sydney, NSW, Australia, 12–15 January 2020; pp. 85–92.
15. Hochreiter, S. Untersuchungen zu Dynamischen Neuronalen Netzen. Bachelor’s Thesis, Technische Universität München, München, Germany, 1991.
16. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [[CrossRef](#)]
17. Hochreiter, S. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **1998**, *6*, 107–116. [[CrossRef](#)]
18. Schmidhuber, J.; Wierstra, D.; Gagliolo, M.; Gomez, F. Training recurrent networks by Evolino. *Neural Comput.* **2007**, *19*, 757–779. [[CrossRef](#)] [[PubMed](#)]
19. Chen, J.; Chaudhari, N.S. Segmented-Memory Recurrent Neural Networks. *IEEE Trans. Neural Netw.* **2009**, *20*, 1267–1280. [[CrossRef](#)]
20. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; A Bradford Book; The MIT Press: Cambridge, MA, USA; London, UK, 2018.
21. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
22. Jiang, L.; Subramanian, P. Forecasting of Stock Price Using Autoregressive Integrated Moving Average Model. *J. Comput. Theor. Nanosci.* **2019**, *16*, 3519–3524. [[CrossRef](#)]
23. Kavzoglu, T.; Teke, A. Predictive Performances of Ensemble Machine Learning Algorithms in Landslide Susceptibility Mapping Using Random Forest, Extreme Gradient Boosting (XGBoost) and Natural Gradient Boosting (NGBoost). *Arab. J. Sci. Eng.* **2022**, *47*, 7367–7385. [[CrossRef](#)]
24. Lilly Sheeba, S.; Neha, G.; Anirudh Ragavender, R.M.; Divya, D. Time Series Model for Stock Market Prediction Utilising Prophet. *Turk. J. Comput. Math. Educ. (TURCOMAT)* **2021**, *12*, 4529–4534.
25. Yun, K.K.; Yoon, S.W.; Won, D. Interpretable stock price forecasting model using genetic algorithm-machine learning regressions and best feature subset selection. *Expert Syst. Appl.* **2023**, *213*, 118803. [[CrossRef](#)]
26. Han, C.; Fu, X. Challenge and Opportunity: Deep Learning-Based Stock Price Prediction by Using Bi-Directional LSTM Model. *Front. Bus. Econ. Manag.* **2023**, *8*, 51–54. [[CrossRef](#)]
27. Zhao, Y.; Yang, G. Deep Learning-based Integrated Framework for stock price movement prediction. *Appl. Soft Comput.* **2023**, *133*, 109921. [[CrossRef](#)]
28. Quadir, A.; Kapoor, S.; Junni, A.V.C.; Sivaraman, A.K.; Tee, K.F.; Sabireen, H.; Janakiraman, N. Novel optimization approach for stock price forecasting using multi-layered sequential LSTM. *Appl. Soft Comput.* **2023**, *134*, 109830. [[CrossRef](#)]
29. Lu, M.; Xu, X. TRNN: An efficient time-series recurrent neural network for stock price prediction. *Inf. Sci.* **2024**, *657*, 119951. [[CrossRef](#)]
30. Salah, S.; Alsamamra, H.R.; Shoqeir, J.H. Exploring Wind Speed for Energy Considerations in Eastern Jerusalem-Palestine Using Machine-Learning Algorithms. *Energies* **2022**, *15*, 2602. [[CrossRef](#)]
31. Li, Z.; Yu, H.; Xu, J.; Liu, J.; Mo, Y. Stock Market Analysis and Prediction Using LSTM: A Case Study on Technology Stocks. *Innov. Appl. Eng. Technol.* **2023**, *2*, 1–6. [[CrossRef](#)]

32. Liu, Y. Analysis and forecast of stock price based on LSTM algorithm. In Proceedings of the 2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI), Fuzhou, China, 24–26 September 2021; pp. 76–79.
33. Jabeur, S.B.; Mefteh-Wali, S.; Viviani, J.-L. Forecasting gold price with the XGBoost algorithm and SHAP interaction values. *Ann. Oper. Res.* **2024**, *334*, 679–699. [[CrossRef](#)]
34. Kumar, G.; Jain, S.; Singh, U.P. Stock Market Forecasting Using Computational Intelligence: A Survey. *Arch. Comput. Methods Eng.* **2020**, *28*, 1069–1101. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.