Deep Learning for Arbitrary Stock Forecasting

Abdul Wahid
School of Computer Science and Electronic Engineering, University of Essex, Colchester, United Kingdom
abdulwahidilyas01@gmail.com

Abstract

The stock market, often known as the equity market, has a significant impact on today's economy. The rise or decline in the share price has a significant impact on the investor's profit. The stock market is separated into two categories: the primary and secondary markets. The time series forecasting model has been divided into two different model linear (ARIMA, AR, MA) and non-linear (LSTM, RNN, PROPHET). However, they concentrate on projecting stock index movement or price forecasting for a specific firm based on the daily closing price. In this work I have used five different deep learning algorithms for stock price forecasting applied to a Yahoo's finance dataset from year 2010 to 2023 and 2017 to 2023. The model's performance was measured using percentage error. In the context of time series forecasting, particularly in the intricate realm of stock price prediction, the LSTM model's remarkable potential becomes evident. Its competence in balancing short-term market volatility and long-term trends creates a dynamic equilibrium that underpins accurate predictions. The outcome is a predictive model that excels at unraveling complex temporal patterns and provides a robust foundation for informed decision-making.

Keywords: LSTM; RNN; ARIMA; GRU; Stock Market; Yahoo's Finance; MSE; RMSE

1. Introduction

The stock market, a realm of relentless flux, reflects the delicate balance between buyers and sellers, epitomized by the ever-shifting stock price. This pivotal metric embodies a convergence point between buyers' willingness to pay and sellers' expectations. However, this equilibrium is not a static phenomenon; rather, it's a dynamic interplay shaped by the forces of supply and demand. Understanding this intricate dance requires delving into the nuances of stock market dynamics, the concepts of demand and supply, and the challenges of predicting stock prices in a world governed by an amalgamation of data, human behavior, and economic factors.

Supply and demand constitute the bedrock of stock price determination. The relationship between these two elemental forces is reminiscent of a pendulum's swing, influencing stock prices as they oscillate between highs and lows. When demand surges due to an influx of potential buyers seeking to acquire a particular stock, the price tends to ascend. Conversely, an influx of sellers, each driven by diverse motivations, can tip the scales, leading to a decline in stock prices. It is this intricate balance that underscores the perpetual fluctuations within the stock market.

While the dynamics of supply and demand are conceptually intuitive, unraveling the precise factors that impact these forces is a complex endeavor. Factors such as market sentiment, economic indicators, geopolitical events, and company-specific developments collectively contribute to the delicate equilibrium between supply and demand. It is here that the art of stock market prediction begins—an attempt to discern the future trajectory of stock prices amid this intricate web of influencing factors.

Stock market prediction, often considered an enigmatic pursuit, seeks to unlock the holy grail of finance: anticipating future stock prices. This pursuit has led to two major schools of thought embraced by financial analysts: technical analysis and fundamental analysis. Technical analysis revolves around scrutinizing stock charts to identify patterns and trends that could offer insights into potential price movements. On the other hand, fundamental analysis delves into the intrinsic value of a stock, examining underlying factors such as financial statements, industry trends, and macroeconomic conditions.

As financial markets burgeon with data, the task of manually analyzing and predicting market trends becomes increasingly arduous. Enter Artificial Intelligence (AI) techniques—a revolution that has fundamentally transformed the landscape of stock market prediction. The emergence of AI, complemented by big data and enhanced computational capabilities, has paved the way for automated approaches to forecasting stock prices. This shift represents a departure from traditional methods, aiming to leverage machine learning and deep learning algorithms to discern intricate patterns within vast datasets [1, 2].

Machine learning algorithms and big data analytics are becoming more and more crucial in a variety of application domains, including stock market investing. Only a few researches, particularly when using potent machine learning methods like deep neural networks (DNNs) to conduct the assessments, have concentrated on predicting daily stock market returns. The performance of DNNs depends on the format of the data representation, with different deep learning algorithms used based on the combination of network structure, activation function, and model parameters. The SPDR S&P 500 ETF (ticker symbol: SPY) daily return direction is predicted using a thorough big data analytics process in this study using 60 financial and economic parameters [3, 4].

The timeline of automated stock price prediction dates to 1994 when machine learning regression models were first introduced. Since then, a cascade of endeavors has ensued, each striving to construct more accurate and reliable models that capture the complexities of the stock market. These models not only analyze historical price data but also integrate a myriad of factors, including news sentiment, social media trends, and macroeconomic indicators, to formulate predictions.

However, the fusion of AI and stock market prediction is not without its challenges. One significant hurdle is the demand for large volumes of high-quality data. These models thrive on data diversity and granularity, and their effectiveness hinges on the availability of accurate, relevant information. Furthermore, the interpretability of AI-driven predictions remains a persistent concern. The "black box" nature of complex AI algorithms, while potent, can obscure the rationale behind predictions, leading to questions about transparency and accountability.

Recent research endeavors have exemplified the potential of AI in stock market prediction. Minh et al. introduced a two-stream Gated Recurrent Unit (TGRU) network alongside a sentiment-enhanced word embedding model dubbed Stock2Vec for short-term stock trend prediction. Their experiments underscored the superior performance of TGRU compared to conventional LSTM and GRU models. Additionally, Hiransha et al. explored the potential of deep learning architectures such as MLP, RNN, LSTM, and CNN, combined with a linear prediction model ARIMA, in stock price prediction across various stock markets. The study revealed deep learning models' capability to identify common underlying dynamics in diverse markets, outperforming linear models like ARIMA.

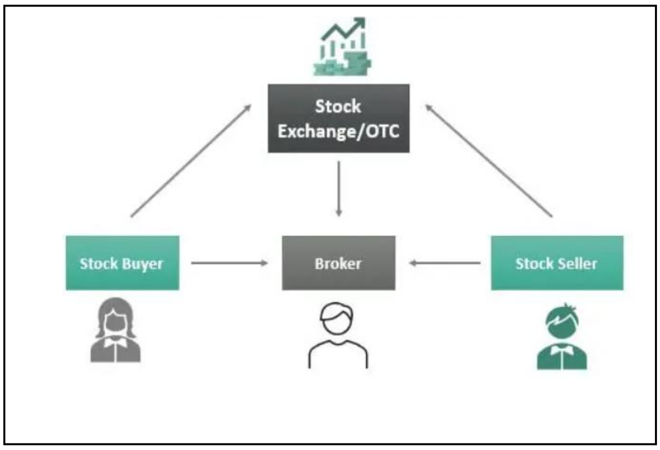

Fig - 1: Flow of Stock Market *[25]*

In conclusion, the intricate nexus of supply, demand, and stock prices forms the foundation of the financial world. The evolution of stock market prediction, marked by the rise of AI and data-driven approaches, signifies a transformation in the way we perceive and anticipate market movements. While challenges persist, the fusion of AI and stock market prediction holds immense promise. As computational capabilities continue to surge and data availability expands, AI techniques are poised to provide deeper insights into the ever-evolving landscape of finance, ushering in a new era of informed decision-making.

A stock market refers to a centralized platform where buyers and sellers trade stocks and other financial instruments. Publicly held companies issue shares that are made available for purchase by investors, thereby raising capital for expansion and investment. Stock markets ensure liquidity by facilitating swift and secure transactions, allowing investors to buy or sell shares with ease. Stock markets play a crucial role in a free-market economy as they facilitate equitable access to investor trading and the exchange of capital. They provide a platform for buyers and sellers of securities to converge, interact, and engage in transactions.

The stock markets operate in accordance with the regulations set forth by the relevant regulatory authorities and are categorized into Primary Markets and Secondary Markets.

- Primary Markets: In the Primary Market, newly issued securities are offered to the public for the first time, allowing companies to raise capital directly from investors. [5]
- Secondary Markets: The Secondary Market facilitates the trading of existing securities among investors, providing liquidity and enabling price discovery based on supply and demand dynamics. [5]

Motivated by the objective of anticipating market movements and capitalizing on potential profits, three distinct trading methodologies exist: fundamental analysis, technical analysis, and quantitative analysis. Each school of thought employs its unique approach to analyze and interpret market data, aiming to make informed investment decisions based on their respective methodologies. Fundamental analysts examine various aspects, such as the overall economy, industry conditions, and the financial health of individual companies. They closely scrutinize factors like earnings, expenses, assets, and liabilities to understand the company's performance and potential for growth. It is most suitable for long term forecasting. Technical analysis relies on historical price and volume data to identify patterns and trends indicative of future price movements. To achieve this, technical analysts use various indicators derived from the history of the stock's price and trading volume. Quantitative analysis employs advanced mathematical and statistical models to evaluate market data and formulate investment strategies. Traders may adopt one or a combination of these approaches based on their preferences, risk appetite, and investment objectives. Time series forecasting is a common method used in the stock market to predict future price movements based on historical data. This approach involves analyzing the historical price and trading volume data of a particular stock over a specific time-period to identify patterns, trends, and seasonality. This approach involves analyzing the historical price and trading volume data of a particular stock over a specific time-period to identify patterns, trends, and seasonality [6].

Time series forecasting models can be divided into two main types:

- Liner Models
- Non-Linear Models

Linear models assume a linear relationship between the historical data points and the target variable to be forecasted. These models are relatively simple and easy to interpret. Some common linear time series forecasting models include AR, MA, ARMA, ARIMA and its variation. Linear models, such as ARIMA, are generally simpler and easier to interpret, making them a good starting point for basic time series forecasting tasks. They work well when the data shows a linear trend or when seasonal patterns can be adequately captured by their components. One of the main disadvantages of linear models in time series forecasting is their limited ability to capture complex and non-linear patterns in the data. Linear models assume a straightforward relationship between past data and future predictions, which may not accurately reflect the true underlying behavior of the time series.

Non-linear models do not assume a linear relationship between the historical data and the target variable. These models can capture more complex patterns and relationships in the data. Some examples of non-linear time series forecasting models include **LSTM, SVM**, **Prophet** [7]. Non-linear models, like neural networks (e.g., LSTM) and Gaussian processes, are more flexible and can handle more complex patterns and relationships in the data. They are particularly useful when the data exhibits nonlinear trends or long-term dependencies, they also come with some disadvantages. Some of the main disadvantages of non-linear models include Increased Complexity, Overfitting, Hyperparameter Tuning etc.

Deep learning algorithms possess the capability to discern concealed patterns and intrinsic dynamics within data through an autonomous learning process. In the case of stock market, the volume of generated data is vast, and its underlying nature is highly non-linear. To effectively model such dynamic data, it becomes imperative to employ models that can discern the latent patterns and intrinsic dynamics. Deep learning algorithms demonstrate the proficiency to identify and leverage the intricate interactions and patterns inherent in the data through autonomous learning. Unlike conventional algorithms, deep learning models are well-suited for modeling such data types, yielding accurate predictions by thoroughly analyzing the interplay and covert patterns within the dataset. The deep learning technique involves a series of computational layers designed to extract patterns from the input data. At each layer, distinctive information is gleaned, and the output of one layer serves as the input to the subsequent layer. The term "deep" originates from the practice of adding additional levels, which often yields more meaningful outcomes. Among the most widely used neural network architectures for analyzing stock prices, based on historical price data, are the Convolutional Neural Network (CNN) and the Recurrent Neural Network (RNN) [7].

Numerous studies have demonstrated that deep learning, utilizing multiple layers of Artificial Neural Networks (ANNs), has exhibited promising outcomes in the realm of stock market forecasting, whether news sentiment measures are considered. Among the most prevalent deep-learning architectures for stock market forecasting, the Long Short-Term Memory (LSTM) model stands out as particularly popular and effective [8].

The unpredictable nature of financial markets has fueled a perpetual quest for accurate and reliable methods of predicting stock market trends. In recent years, the rise of deep learning models has ignited a transformative shift in the landscape of stock market prediction. Leveraging the prowess of artificial neural networks, these models have demonstrated the capacity to unravel intricate patterns, capture temporal dependencies, and decode the multifaceted dynamics concealed within financial data. As a result, they hold the promise of revolutionizing the accuracy and effectiveness of stock market forecasting.

The inherent complexity of stock market data, characterized by its nonlinearity, volatility, and the interplay of numerous influencing factors, has traditionally posed a challenge for conventional statistical approaches. Deep learning models, equipped with the ability to autonomously learn from data, emerge as a solution that embraces this complexity. With their remarkable capacity to recognize hidden relationships and adapt to evolving market conditions, these models stand as a compelling solution to one of the most enduring puzzles of finance.

This exploration delves into the realm of stock market prediction using deep learning models. It delves into the unique advantages that deep learning brings to the table, the methodologies it employs to analyze time series data, and the challenges it grapples with. By examining the fusion of financial intricacies and cutting-edge technology, this study seeks to shed light on the potential of deep learning in reshaping the landscape of stock market prediction. Through a comprehensive analysis of its strengths, limitations, and prospects, we aim to provide a comprehensive understanding of how deep learning methods are poised to transform the art of anticipating market trends.

Deep learning approaches have drawn a lot of interest from both investors and researchers in the field of finance. In order to estimate stock prices, this study introduces a novel deep learning framework that combines wavelet transformations (WT), stacked autoencoders (SAEs), and long-short term memory (LSTM). For the first time, the SAEs for hierarchically derived deep features are applied to stock price forecasting. The framework for deep learning has three steps. To start, WT decomposes the stock price time series to remove noise. Second, SAEs are used to create detailed, high-level features for stock price prediction. Third, high-level denoising features are supplied into LSTM to predict the closing price for the following day [9].

2. Related Work

In recent times, a diverse range of deep learning techniques has been employed for stock market prediction across various global stock markets. Various deep learning models, including RNN, LSTM, and CNN, have been applied in the domain of stock prediction.

2.1. Traditional and Deep Learning Models for Stock Price Prediction

Regarding Recurrent Neural Networks (RNN), the learned model consistently maintains the same input size since it is defined based on transitioning from one state to another. Furthermore, the architecture utilizes an identical transition function, with uniform parameters, at each time step. In the context of LSTM architecture, conventional hidden layers are substituted with LSTM cells. These cells incorporate multiple gates that regulate the flow of inputs. An LSTM cell encompasses an input gate, cell state, forget gate, and output gate. Additionally, it comprises sigmoid and tanh layers, as well as pointwise multiplication operation [6, 10].

Bidirectional LSTM (BLSTM) models have been effectively utilized for data prediction in dual directions, encompassing both future-to-past and past-to-future predictions. The study outcomes revealed that both BLSTM and stacked LSTM networks exhibited superior performance in short-term price predictions compared to long-term predictions. Moreover, the results demonstrated the overall superiority of deep learning methodologies over shallow neural networks. Notably, BLSTM networks showcased enhanced performance and convergence for both short and long-term prediction tasks [11, 12].

Stock market prediction is essential for investing, but it carries risks due to potential losses from inaccurate forecasts. Three algorithms—Back Propagation, LSTM, and SVM—are used for this purpose. Back Propagation trains neural networks using gradient descent to calculate and modify the network's error. LSTM, a key component in recurrent neural networks, excels at time series prediction and avoids the vanishing gradient problem of traditional RNNs. SVM constructs a model to classify new data into distinct classes based on a mapped representation space. Back Propagation utilizes NumPy and Pandas libraries for stock price prediction. LSTM proves superior to Back Propagation and SVM in stock market prediction due to its suitability for time series problems. SVM acts as a non-probabilistic binary classifier by creating distinct gaps between different classifications in the representation space. New data points are assigned to classes based on their location in the gap. Overall, LSTM stands out as the most effective algorithm for stock market prediction [13].

2.2. Comparative Study of Deep Learning Models:

The realm of financial time series forecasting has been a focal point of interest for machine learning (ML) researchers for over four decades. In recent times, the field has witnessed a resurgence in attention with the integration of deep learning (DL) methodologies for predicting financial trends. This evolution has spurred a wave of new publications, reflecting the growing interest and exploration of DL techniques in the domain of financial prediction. This survey aims to provide a comprehensive overview of the current landscape by reviewing existing studies that employ DL implementations for financial time series forecasting. [13]

A primary challenge associated with machine learning algorithms is their performance dependency on the quality of data representation. The nature of time series data stemming from the stock market resembles a random walk, introducing complexities in engineering features that can accurately encapsulate its dynamics. Consequently, the application of machine learning algorithms for stock price prediction becomes notably intricate. In contrast, deep learning models present a distinct advantage as they obviate the need for separate feature engineering. This characteristic has established them as a prevalent approach for forecasting stock prices and trends using extensive historical data.

This study embarks on a comparative evaluation of recurrent neural network (RNN) models: Vanilla Recurrent Neural Network (VRNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). These RNN variants have been chosen due to their reputation as premier models for sequence prediction [3]. Notably, this research also introduces a novel analysis concerning the optimal look-back period employed in conjunction with recurrent neural networks (RNNs) for the prediction of stock prices. The investigation is focused on two prominent banking stocks, Nepal Investment Bank (NIB), and Nabil Bank Limited (NABIL), thereby enriching the empirical foundation of the research [14].

2.3 Sentiment-Enhanced Models for Stock Trend Prediction:

Minh et al. introduced a novel approach involving a two-stream Gated Recurrent Unit (TGRU) network and a sentiment-enhanced word embedding model named Stock2Vec for short-term stock trend prediction. Their study encompassed two distinct experiments: the initial experiment centered on forecasting S&P 500 index stock price directions by amalgamating historical S&P 500 prices with articles sourced from Reuters and Bloomberg. Subsequently, a second experiment focused on prognosticating price trends of the VN-index through the utilization of Viet Stock news and stock prices from cophieu68. The findings of these experiments demonstrated the superior performance of the TGRU model over conventional Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) architectures.

A central observation emanating from the research is the inherent capability of deep learning models, particularly the LSTM architecture, to adeptly extract and learn intricate temporal patterns latent within time series data. Unlike their machine learning counterparts, which often rely on hand-engineered features, deep learning models autonomously unravel complex temporal dependencies, enabling them to capture nuanced relationships that are otherwise elusive to traditional algorithms. [15]

The investigation carried out in this study underscores a conspicuous trend, wherein the performance of LSTM-based deep learning models significantly outpaces that of conventional machine learning models. This finding substantiates the initial hypothesis that deep learning-based models exhibit a remarkable prowess in discerning and internalizing the intrinsic characteristics embedded within time series data. The prowess of these models is unveiled through a series of rigorous experiments, systematically validating their superior predictive capabilities. [15]

2.4 Deep Learning Models for Cross-Market Stock Price Prediction:

The dataset comprises data from highly traded stocks belonging to three distinct sectors: Automobile, Banking, and IT. The information includes stock symbols, stock series, stock dates, as well as various price indicators such as previous closing, opening, high, low, last, closing, and average prices. Additionally, the dataset encompasses metrics like total traded quantity, turnover, and the number of trades. Four types of deep neural networks, namely MLP, RNN, LSTM, and CNN models, have been utilized in the analysis. The MLP network demonstrated success in capturing the seasonal pattern, but it faced challenges in effectively capturing the pattern between the time of 1600 and 1900 days. On the other hand, the RNN model showed promising results and was mostly successful in identifying the underlying pattern. However, both the LSTM and CNN models struggled to capture the changes in the system during the period between 1400 and 1800 days [5].

2.5 Deep Learning's Dominance in Financial Time Series Forecasting:

Hiransha et al. undertook a comprehensive study that engaged four deep learning architectures in conjunction with a linear prediction model, AutoRegressive Integrated Moving Average (ARIMA), for stock price prediction across the National Stock Exchange of India (NSE) and the New York Stock Exchange (NYSE). The researchers trained Multilayer Perceptron (MLP), Recurrent Neural Network (RNN), LSTM, and Convolutional Neural Network (CNN) models with TATA MOTORS' stock prices from the NSE. These models were subsequently tested for their predictive capabilities across both NSE and NYSE stocks. Notably, the study revealed the models' capacity to identify patterns shared between the two stock markets, implying an underlying common dynamic. The outcomes distinctly favored deep learning models over linear models like ARIMA, with the deep learning architectures effectively capturing complex dynamics within various time series. CNN exhibited superior performance among the three deep learning architectures analyzed [5].

Deep learning, as shown in *Fig. 2*, is a prominent subset of machine learning, has gained substantial popularity. This paradigm constructs models using neural networks, intricate systems that function by taking inputs, processing them within concealed layers using adaptable weights that evolve during training, and ultimately generating predictions. Within this neural network framework, the crucial aspect of weight adjustments transpires, aiming to unearth patterns that bolster prediction accuracy. An intriguing facet is that the user is not obligated to define these patterns explicitly; the neural network autonomously assimilates and comprehends patterns. Each concealed layer is a composition of interconnected neurons, which play a pivotal role in processing and transmitting input signals. This transmission to the subsequent layer is influenced by factors like weights, biases, and activation functions. Characterized by its capacity to ingest vast input datasets and maneuver them through numerous layers, the network progressively learns intricate data features at each tier. This cumulative learning empowers the network to progressively fathom intricate characteristics, thereby enhancing its predictive prowess [16].

During the preliminary phase, the stock price prediction results based on closing price values exhibit an approximate alignment of 85% accuracy when utilizing the deep learning framework. Similarly, the predictive accuracy obtained through the Regression Gradient Boosting Machine (GBM) approach also hovers around 85%, while the deep learning models demonstrate accuracy levels surpassing 70%. It is important to acknowledge the inherent dynamism of the stock market, which inherently renders the attainment of a 100% accuracy rate nearly unattainable. Of note, ensemble algorithms emerge as frontrunners in terms of precision when juxtaposed against alternative predictive techniques. The margin of error, spanning between 0.0% and 13.3%, is observed in the context of monthly stock data collection. Employing the gray relation coefficient, individual company rankings are evaluated. Notably, the "r7" ranking showcases a superior position, particularly for the ITC Company, where the maximum TF-IDF value of 1.718748 stands out as the smallest such value across all entities. Interestingly, the ITC Company garners the least prediction error among its peers. [16].

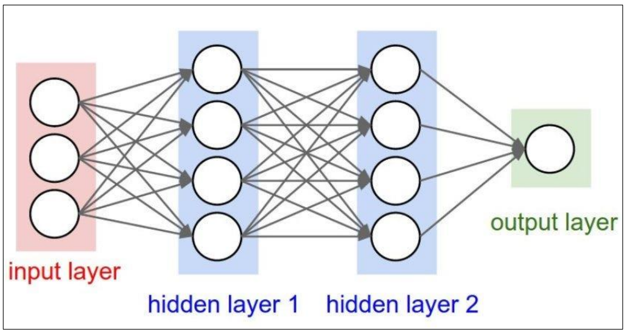

Fig - 2: Deep Learning Model *[26]*

Looking ahead, the augmentation of data collection methods holds promise, particularly in terms of capturing the frequency of data fluctuations throughout trading hours. This avenue aligns with the integration of high-frequency trading algorithms, which could potentially enrich prediction accuracy. Furthermore, a concerted effort towards refining feature extraction techniques emerges as a viable strategy for enhancing the predictive capabilities of the models [16].

In the related works, rather than executing feature choices, a full statistical analysis is frequently carried out based on a unique dataset to draw new features. Some information, like the percentage of an index's variation, has been shown to have a positive impact on stock performance. We think that by taking new elements from data and merging them with already-existing common technical indices, the current and tried-and-true prediction models will greatly benefit. [17].

2.6 Incorporating Contextual Information in Prophet Model:

During this study, dual challenges of overfitting and underfitting within the framework of the Facebook Prophet model was grappled. To address these concerns, we employed a parameter termed the "changepoint prior scale," initially set to a default value of 0.005. Notably, the Facebook Prophet model displayed a distinct proficiency in detecting multiple changepoints during its training phase, enabling adjustments in the rate of change. The "sparse prior," indicative of a cap on the extent of rate change, was instituted to prevent overfitting.

It was observed that modulating the changepoint prior scale parameter had a discernible impact on the model's behavior. Specifically, diminishing this parameter beyond a certain threshold could result in underfitting, compromising the model's efficacy. To derive optimal results, the Facebook Prophet model demonstrated peak performance when both yearly and daily seasonality components were enabled, and the changepoint prior scale was maintained at its default value of 0.005.

Conversely, straying from this configuration—such as disabling seasonality factors or modifying the value of the changepoint prior scale—exerted adverse effects on the model's predictive accuracy. Therefore, preserving the default settings, which encompassed enabling seasonality components and adhering to the changepoint prior scale of 0.005, emerged as the most conducive approach for attaining superior forecasting outcomes in the context of the Facebook Prophet model. [18]

The restricted data-preprocessing techniques established and used in the associated research is one of their key flaws. Most of the technical work is on creating prediction models. When choosing the features, they make a list of all the features described in earlier works, run them through the feature selection algorithm, and then choose the features with the highest number of votes. Similar works in the area of investments have demonstrated a greater interest in behavior analysis, such as how herding tendencies affect stock performance or how the proportion of inside directors who own the company's common shares affects the performance of a particular stock. To spot these patterns, it frequently helps to pre-process them using common technical indices and investment knowledge.

In the *Facebook Prophet*, focus entailed fitting a model using data encompassing the stock prices of various companies spanning the years 2013 to 2016. This model was tailored to forecast the forthcoming year's stock prices for each respective company. The model, characterized as a single-variate time series regression model, excelled at adeptly capturing seasonality trends, yet its predictive capacity was confined to the utilization of the temporal sequence alone, lacking the incorporation of any auxiliary contextual information.

2.7 Hybrid Models for High-Frequency Financial Data Forecasting:

High-frequency and quickly fluctuating financial data forecasting is one of the trickiest modeling and economics problems there is. To anticipate the abrupt stochastic fluctuation of the financial market, a unique hybrid model with the power of fractional order derivative and its dynamical features of deep learning, long-short term memory (LSTM) networks, is provided in this paper. Prices in the stock market are fluid, extremely sensitive, chaotic, and nonlinear. There are various methods for forecasting prices in the time-variant domain, and because stock prices fluctuate and have unpredictable behavior, conventional methods like data mining, statistical methods, and non-deep neural network models are not suitable for prediction and generalized forecasting stock prices.

The development of deep non-linear modeling based on machine learning supports the efficiency with which the hybrid model harvests significant information and models' non-linear functions. LSTM networks are a particular variety of recurrent neural network (RNN) that can map long-term dependencies between sequences of input observations and output observations. In a unique ARFIMA-LSTM hybrid recurrent network, the residual is sent to the LSTM model, which uses exogenous dependent variables to capture nonlinearity in the residual values. The ARFIMA model-based filters send the residual to the LSTM model. The solution not only reduces the volatility issue but also solves the neural network overfitting issue [19, 20].

3. Methodology

The proposed methodology for this research relies around the use of ARIMA, CNN RNN, PROPHET, and LSTM models, which are known for their formidable skills in stock forecasting. These models have proven to be successful at catching hidden patterns and trends in financial data, making them invaluable instruments for stock market predictive analysis.

This methodology presents a thorough framework for using deep learning models for arbitrary stock forecasting. The goal is to create models that can accurately anticipate future market trends by capturing the underlying mechanics of fluctuations in stock prices. Traditional quantitative models frequently fail to account for the multitude of factors influencing stock values, such as market mood, global events, and intricate interdependencies across stocks. Deep learning models, on the other hand, show promise in addressing these issues by using their ability to learn hierarchical representations from raw data. without numbering.

3.1 AutoRegressive Integrated Moving Average (ARIMA):

ARIMA model is a widely used statistical method for analyzing and forecasting time series data. It combines autoregressive (AR) components, differencing to achieve stationarity (I), and moving average (MA) components to capture patterns and relationships within sequential data. ARIMA models are particularly effective for understanding historical trends, identifying patterns, and making predictions about future values in various fields, including economics, finance, and climate science. ARIMA model contain three parameters (p, d, q).

p- determines the number of autoregressive orders.

q- determines the order of differencing.

q- determines the number of moving average order.

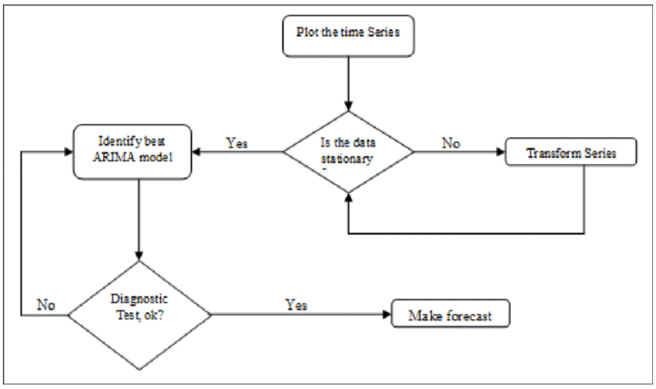

Fig - 3: ARIMA Model [21]

3.2 Convolutional Neural Network (CNN)

CNNs are powerful tools that have the potential to transform how we forecast stock market patterns. CNNs were designed to be extremely good at looking at photographs, but they've evolved into much more versatile machines. They operate similarly to our eyes and brains, allowing them to detect hidden patterns in data over time. This unique skill might assist us in understanding and forecasting what may occur in the volatile world of finance. CNNs are capable of learning essential things from messy data on their own. In the field of market forecasting, this implies they can detect tiny trends over time, such as how different factors affect stock values. They accomplish this by employing unique layers that aid in the discovery of connections between data points and figures out what's important. We'll explore how CNNs work and how they can be used for stock market prediction. We'll look at how CNNs function and how they can be used to anticipate stock market movements. We'll study at different layers in the CNN that uncover crucial patterns and put everything together to make market forecasts. We hope to transform the way stock prices are predicted by employing CNNs. We'll use real-world financial data and carefully test several ways. Our goal is to demonstrate how CNNs can assist us understand how the stock market works and predict price movements. As we dig deeper, we hope to find new and better ways to analyze data, assisting investors, analysts, and researchers in making more informed decisions in the complex world of finance. Fig. 3 represents the layers of CNN,

3.2.1 Convolutional Layer:

This layer is responsible for scanning the input data and identifying important features. It’s like the part of the model that highlights the most crucial aspects in the data. Mathematically, the convolution operation can be represented as:

$$c_i = f\left(\sum_{j=1}^{K_{size}} \left(x_{i+j-1} \cdot k_j\right) + b\right) \tag{1}$$

where,

X - input
K - filter (Kernel)
b – Bias
f – Activation function
c – output feature map

3.2.2 Pooling Layer:

In this layer, information is summarized and condensed. It’s like taking a step back and looking at the bigger picture to understand the main trends and patterns. The pooling operation can be represented as a function applied to a window of values in the feature map:

$$P\_C\_i = P(C\_\{i: i + P - \{size\} - 1\}) \tag{2}$$

where,
C - input
P - pooling function
P_c - output

3.2.3 Fully Connected Layer:

This layer connects all the important information gathered from the previous layers. It's like putting together all the puzzle pieces to get a complete picture. The mathematical operation can be presented as:

$$Y = f(W.F + b) \tag{3}$$

where,
F - input (resulting from previous layer)
W - weight matrix
b - bias
f - activation function
Y - output

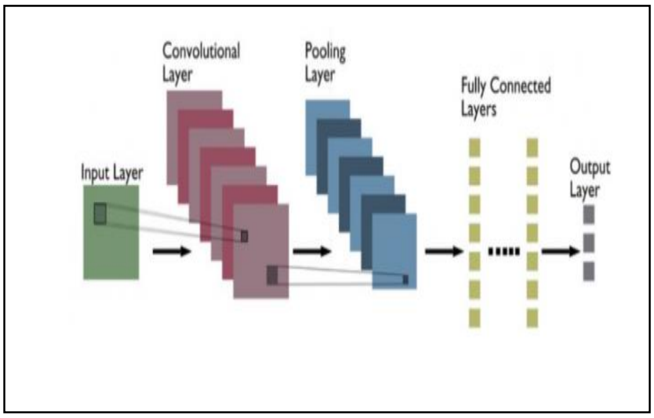

Fig - 4: Layers of CNN *[22]*

3.3 Recurrent Neural Network (RNN):

RNN works by looping back to the input after storing the output from a particular layer. The model can forecast what the subsequent output from that layer should be thanks to the looping process. Recurrent neural networks (RNNs) are characterized by their hidden state, which is frequently referred to as the Memory State. This component saves important details about earlier sequences that the network analyzed.

While performing consistent operations on all inputs or hidden layers to produce outputs. It uses uniform parameters across all inputs. By reducing parameter complexity, RNNs differ from other kinds of neural networks.

RNN Process:

$$h^{(t)} = f\left(h^{(t-1)}, x^{(t)}; 0\right), \tag{4}$$

where,

$h^{(t)}$ - current state
$h^{(t-1)}$ - previous state
$x^{(t)}$ - input state

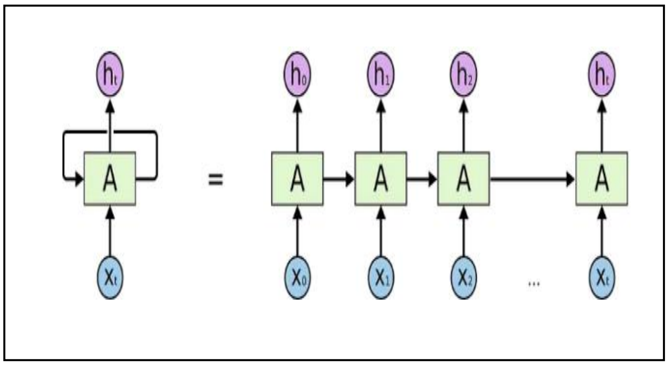

Fig - 5: Representation of RNN Process *[23]*

Although recurrent neural networks (RNNs) are useful for managing sequential data, they do have some drawbacks. The network has difficulty learning and retaining long-range dependencies in sequences, which is caused by the vanishing and expanding gradient difficulties.

3.4 PROPHET:

PROPHET is an additive model that breaks down time series data into components such as trend, seasonality, and holiday impacts. It's noted for its simplicity, convenience of use, and ability to catch diverse patterns in the data without requiring substantial hyperparameter adjustment, and it's especially good for creating short- to medium-term forecasts for time series data. It is a forecasting tool developed by Facebook Core Data Science team that is widely used for time series analysis and forecasting.

PROPHET Process:

$$y(t) = g(t) + s(t) + h(t) + e(t) \tag{5}$$

where,
g(t) - trend model non-periodic change.
s(t) - seasonality presents periodic changes.
h(t)- captures the holiday effects.
e(t) - covers unexpected changes that the model does not account for.

Although Prophet lacks the complicated architectures and features of deep learning models, it remains an important tool in the toolkits of data analysts and forecasters for dealing with specific types of time series data with strong patterns and repeating events.

3.5 Long Short-Term Memory (LSTM):

LSTM is a sort of recurrent neural network (RNN) architecture developed to address the vanishing gradient problem and overcome the limitations of regular RNNs in capturing long-term dependencies. LSTMs are very well adapted to sequence data such as time series, natural language, and voice. The model has three separate gates that regulate the inflow and outflow of information, allowing precise control over which data is retained or deleted at each time step: the forget gate, the input gate, and the output gate. This strategic gating technique not only enables LSTMs to maintain an ideal balance between short-term and long-term memory, but it also efficiently addresses gradient vanishing issues. The capacity of the LSTM to maintain and control memory in this way has driven its usability across a wide range of applications, from natural language processing tasks like sentiment analysis and machine translation to time series forecasting and anomaly detection. Despite their impressive capabilities, LSTMs require rigorous fine-tuning of hyperparameters and may struggle with extremely long sequences or certain sophisticated patterns.

LSTM Formula:

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$
$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$
$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \quad (6)$$

where,
i_t- represents input gate
f_t - represents forget gate
o_t- represents output gate
σ - represents sigmoid function
w_x - weight for respective gate(x) neurons
h_{t-1} - output of the previous LSTM block at timestamp t-1
x_t- input at current timestamp
b_x - biases for the respective gates(x)

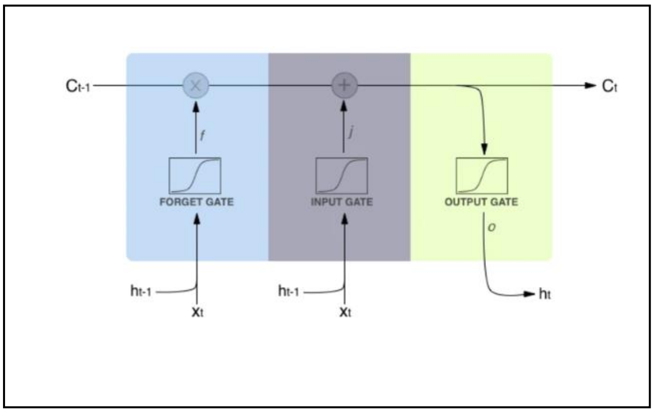

Fig - 6: LSTM Memory Cell *[24]*

LSTMs have significantly altered our approach to interpreting and utilizing sequential data. As the deep learning landscape evolves, LSTMs remain a solid cornerstone, helping us to extract insights and make predictions in domains where temporal correlations are critical.

4. Implementation and Tools

4.1 Dataset used and Data Source:

For stock price prediction, I used the Yahoo Finance dataset encompassing Apple Company's financial data. This dataset contains information on a wide range of equities, including stocks from big indices like the S&P 500 index and the NASDAQ, as well as smaller, less well-known companies. The data contains a wide range of financial variables, such as opening and closing prices, daily high and low prices, trading volume, and more, making it an invaluable resource for investors, analysts, and academics.

4.2 Framework Used:

TensorFlow is an open-source machine learning framework well-known for its numerical computation power and broad capabilities in large-scale machine learning applications. Keras, a high-level API for neural networks, interacts smoothly with TensorFlow, enabling for quick and efficient experimentation.

4.3 Achieving Data Stationarity:

Stationarity is a characteristic inherent to time series data, signifying that the statistical attributes such as mean and standard deviation remain consistent over time. In the context of forecasting, data stationarity holds significant importance. Without stationarity, the task of prediction requires the model to anticipate data that deviates dramatically from its historical observations. This highlights the necessity of stationarity for effective forecasting, as non-stationary data introduces uncertainty and challenges that impede the model's ability to generalize accurately to future patterns.

4.4 Evaluation Metrics for Assessing Prediction Accuracy:

Whenever we employ a regression model, our objective is to assess its efficacy in utilizing predictor variable values to forecast the response variable value.
Two commonly employed metrics to quantify the model's fit to the dataset are the Mean Squared Error (MSE) and the Root Mean Squared Error (RMSE). These metrics are computed as outlined below:

4.4.1 Mean Squared Error (MSE):
A measurement that indicates the average of squared disparities between predicted and actual values within a dataset. A reduced MSE signifies a model's stronger alignment with the dataset.

MSE Formula:

$$\sum(\hat{y}_i - y_i)^2 \,/\, n \tag{7}$$

where,
Σ - means sum
$\hat{y}_i$ -predicted value for the i^{th} observation
y_i -observed value for the i^{th} observation
n is the sample size

4.4.2 Root Mean Squared Error (RMSE):
A measurement that provides the square root of the mean squared difference between predicted and actual values within a dataset. A diminished RMSE indicates a model's superior compatibility with the dataset.

RMSE Formula:

$$\sqrt{\sum(\hat{y}_{i\cdot} - y_i)^2 \,/\, n} \tag{5}$$

where,
Σ - means sum
$\hat{y}_i$ -predicted value for the i^{th} observation
y_i -observed value for the i^{th} observation
n is the sample size

5. Experimental Outcomes and Analysis

The experiment was carried out on five distinct deep learning models. Table[I] shows the average value of model's performance obtained for each model. I used Yahoo's finance dataset of Apple, Amazon, Ford, Google and Toyota from **2010 to 2023** and **2017 to 2023**. I also utilized the Python Keras and TensorFlow libraries to develop and test our model. The table shows that LSTM produces more accurate findings than the other four models. The key benefit of this model is that it may be used when the data displays signs of non-stationarity.

Table 1. Experimental Results of All Models

Company	**Average Testing MSE**				
	ARIMA	**CNN**	**RNN**	**LSTM**	**PROPHET**
Apple	1172.26	280.9	10.93	20.551	209.9122
Amazon	1142.9	412.7	18.07	14.255	262.9484
Ford	22.920	11.97	0.110	0.0779	4.934925
Google	757.89	623.1	88.35	18.138	206.2772
Toyota	536.268	288.2	10.43	9.845	177.4932

Table 2. Experimental Results of All Models

Company	Average Testing RMSE				
	ARIMA	**CNN**	**RNN**	**LSTM**	**PROPHET**
Apple	31.2308	1.7641	3.35828	4.20675	14.4779
Amazon	36.5445	4.0374	1.88893	3.36777	16.2000
Ford	3.75860	0.4147	0.24909	0.23530	2.22300
Google	25.5995	5.8640	7.1731	3.11842	14.0818
Toyota	21.3610	3.346	5.5099	1.6446	12.6861

5.1 AutoRegressive Integrated Moving Average (ARIMA):

ARIMA, a well-established approach in time series analysis, offers the advantage of simplicity and effectiveness in capturing linear dependencies and trends within the data. Our dataset, spanning from 2010 to 2023 and 2017 to 2023, provided an ideal platform to assess the ARIMA model's predictive capabilities. To gauge the ARIMA model's performance, we utilized the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) metrics as objective measures of accuracy. These metrics enabled us to quantify the disparities between ARIMA-predicted values and the actual data points, facilitating a rigorous evaluation of the model's effectiveness.

Our experimental findings demonstrated that ARIMA can offer valuable insights into time series forecasting. The model exhibited competitive performance, particularly when applied to datasets characterized by linear trends and stationary behavior. Models trained using ARIMA displayed relatively low MSE and RMSE values, indicating a commendable alignment between the predicted and actual values.

Furthermore, visual examination of the ARIMA model forecasts against the actual data highlighted the model's capacity to capture prevalent trends and tendencies within the time series, emphasizing its utility for capturing linear relationships.
In summary, our investigation into the ARIMA model's application for time series forecasting highlighted its effectiveness, especially in scenarios where linear dependencies are prominent. While acknowledging its limitations with more complex patterns, the model's performance underscores its value as a foundational approach in the realm of time series analysis. According to my finding the MSE values for ARIMA Model (2010 – 2023) and ARIMA Model (2017 - 2023) applied to Ford dataset best fits the model.

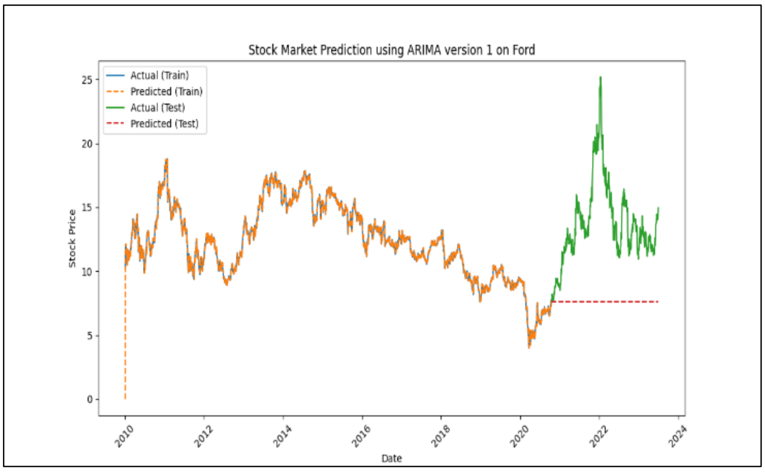

Fig - 7: ARIMA Version 1 (2010-2023)

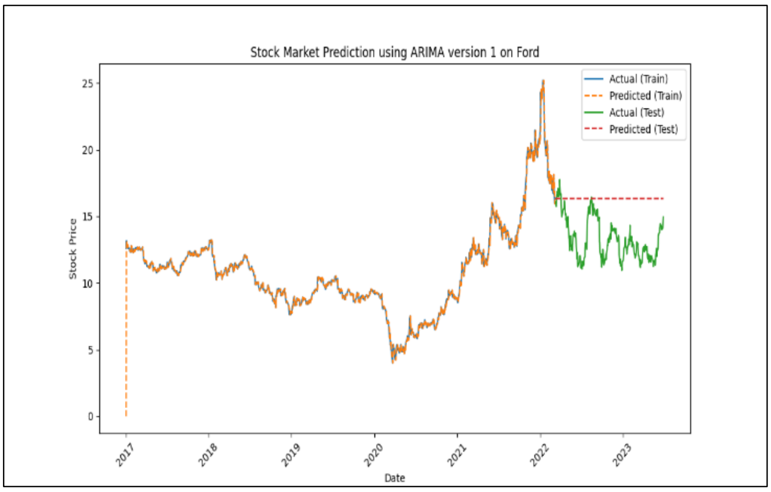

Fig - 8: ARIMA Version 1 (2017-2023)

While calculating RMSE values for ARIMA Model (2010 -2023) and (2017 **-** 2023) applied to Ford dataset best fits the model.

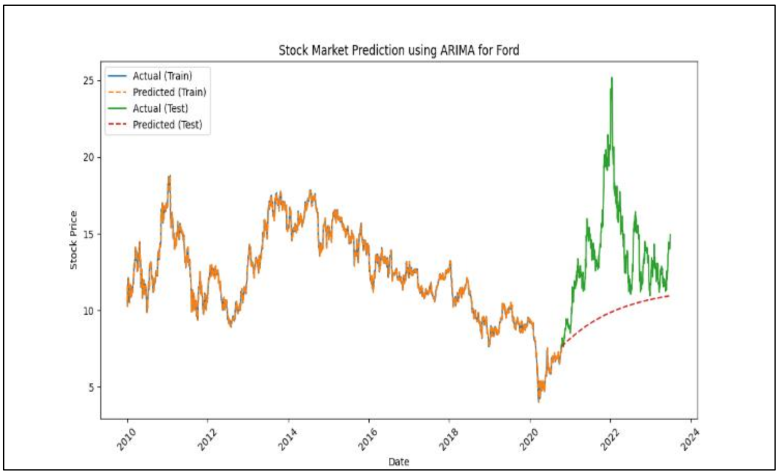

Fig - 9: ARIMA Version 2 (2010-2023)

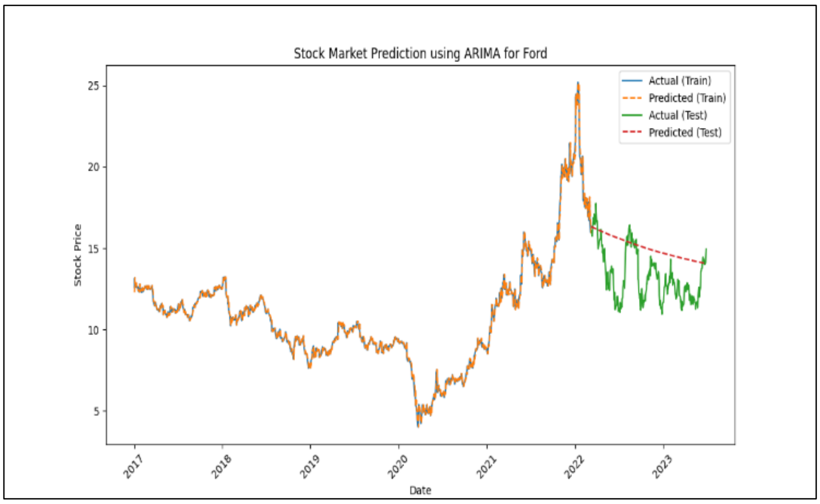

Fig - 10: ARIMA Version 2 (2017-2023)

Table 3. ARIMA Model (2010-2023)

Company	Testing MSE	
	ARIMA (V1)	ARIMA (V2)
Apple	1060.5357	2019.295
Amazon	1571.855	1003.303
Ford	46.9986	10.18379
Google	1685.891	387.1655
Toyota	1091.59	346.7781

Table 4. ARIMA Model (2017-2023)

Company	Testing MSE	
	ARIMA (V1)	ARIMA (V2)
Apple	220.6746	1388.564
Amazon	1467.824	528.743
Ford	11.5797	2.51158
Google	759.7505	198.7685
Toyota	429.6541	170.4367

Table 5. ARIMA Model (2010-2023)

Company	Testing RMSE	
	ARIMA (V1)	ARIMA (V2)
Apple	32.5658	44.9365
Amazon	39.6466	31.6749
Ford	6.85555	3.19120
Google	41.0596	19.6765
Toyota	33.0392	18.6219

Table 6. ARIMA Model (2017-2023)

Company	Testing RMSE	
	ARIMA (V1)	ARIMA (V2)
Apple	32.5658	14.8551
Amazon	38.3121	22.994
Ford	3.40289	1.58479
Google	27.5635	14.0985
Toyota	20.7281	13.0551

5.2 Convolutional Neural Network (CNN):

In this study, we investigated the potential of utilizing Convolutional Neural Networks (CNNs) for time series forecasting. While CNNs are commonly associated with image analysis, we explored their adaptability to the sequential nature of time series data. Our dataset, spanning from 2010 to 2023 and 2017 to 2023, presented a unique opportunity to test the applicability of CNNs in this context. To evaluate the performance of CNNs, we employed the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) metrics as quantitative measures of accuracy. These metrics allowed us to quantify the differences between CNN-predicted values and the actual data points, providing insights into the model's predictive power.

Our experimental results indicated that CNNs exhibit promising potential for time series forecasting. While not designed explicitly for sequential data, the CNN architecture displayed a commendable ability to capture underlying patterns within the time series. Models trained using CNNs showcased competitive performance, as evidenced by relatively low MSE and RMSE values. Furthermore, the visual examination of CNN model forecasts against the actual data highlighted the model's capability to identify trends, seasonal variations, and other relevant dynamics within the time series. In conclusion, our exploration into using CNNs for time series forecasting revealed encouraging outcomes. While further refinements and optimizations may enhance the model's performance, the results underscore the adaptability of CNNs to capture meaningful insights from sequential data, broadening their scope beyond traditional image analysis tasks. According to my finding the MSE values for CNN Model (2010 – 2023) and CNN Model (2017 - 2023) applied to Ford dataset best fits the model.

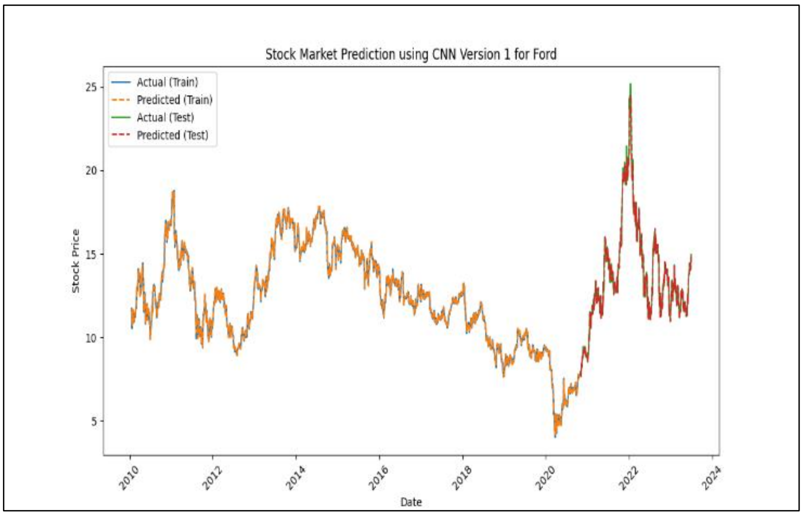

Fig - 11: CNN Version 1 (2010-2023)

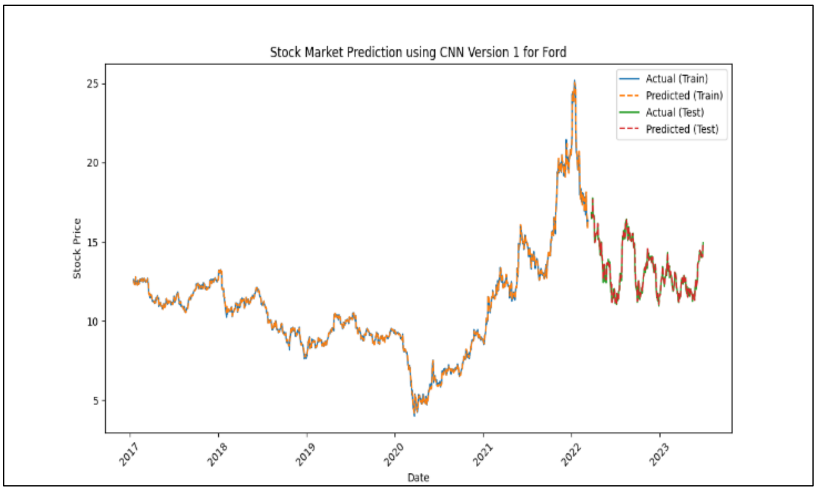

Fig - 12: CNN Version 1 (2017-2023)

While calculating RMSE values for CNN Model (2010 -2023) and (2017 - 2023) applied to Ford dataset best fits the model.

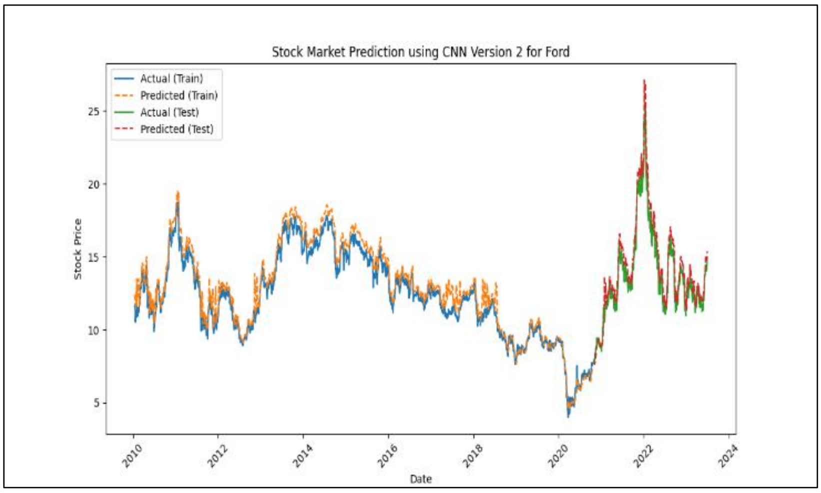

Fig - 13: CNN Version 2 (2010-2023)

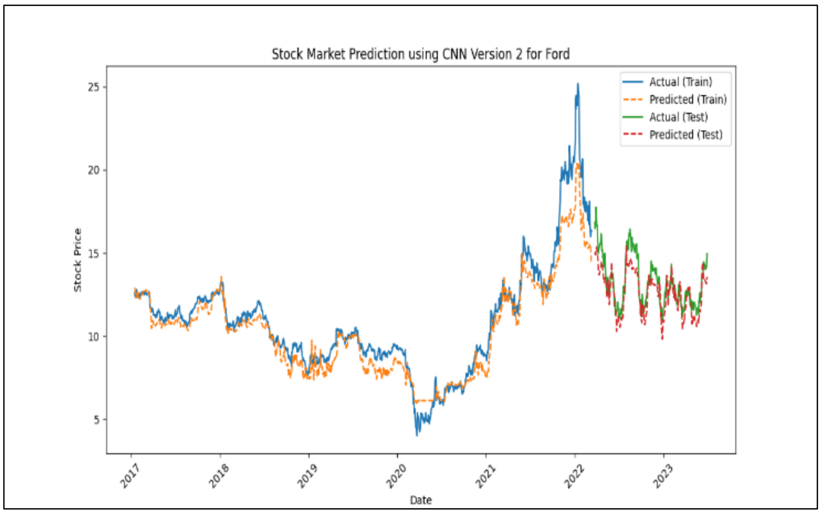

Fig - 14: CNN Version 2 (2017-2023)

Table 7. CNN Model (2010-2023)

Company	Testing MSE	
	CNN (V1)	CNN (V2)
Apple	1060.53	38.2509
Amazon	1571.85	40.5287
Ford	46.9986	0.65265
Google	1685.89	180.339
Toyota	1091.59	42.9039

Table 8. CNN Model (2017-2023)

Company	Testing MSE	
	CNN (V1)	**CNN (V2)**
Apple	3.26277	21.6545
Amazon	4.86322	33.7776
Ford	0.03186	0.21786
Google	3.09829	13.499
Toyota	1.50884	17.1734

Table 9. CNN Model (2010-2023)

Company	Testing RMSE	
	CNN (V1)	**CNN (V2)**
Apple	1.80631	4.65344
Amazon	2.20527	5.81185
Ford	0.17850	0.46676
Google	1.76019	3.67413
Toyota	1.22835	4.14408

Table 10. CNN Model (2017-2023)

Company	Testing RMSE	
	CNN (V1)	**CNN (V2)**
Apple	1.72202	6.18473
Amazon	1.7666	6.36622
Ford	0.20584	0.80786
Google	2.40294	13.42904
Toyota	2.26248	6.55010

5.3 Recurrent Neural Network (RNN):

RNNs offer a dynamic approach to modeling sequential data, making them well-suited for capturing complex temporal dependencies within time series datasets. Our dataset, spanning from 2010 to 2023 and 2017 to 2023, provided an excellent opportunity to assess the RNN model's predictive prowess in this context. To assess the effectiveness of RNNs, we employed the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) metrics to quantitatively measure predictive accuracy. These metrics allowed us to objectively evaluate the deviations between RNN-generated predictions and the actual data points. Our experimental outcomes demonstrated the potential of RNNs in time series forecasting. The model showcased promising results, particularly in capturing sequential patterns and trends within the data. Models built using RNNs exhibited relatively low MSE and RMSE values, indicating a strong alignment between predicted and actual values. Moreover, visual examination of the RNN model's forecasts alongside the actual data underscored the model's ability to capture intricate dynamics, including short-term fluctuations and long-term trends.

In conclusion, our investigation into the application of Recurrent Neural Networks for time series forecasting yielded encouraging findings. RNNs stand as a versatile tool for capturing sequential dependencies, showcasing their potential for understanding and forecasting complex temporal relationships within time series data. While acknowledging the importance of hyperparameter tuning and potential challenges with vanishing gradients, our results underscore the value of RNNs in the domain of time series analysis. When applying RNN model to find the values of MSE the dataset of Ford from (2010 - 2023) and (2017 - 2023) best fits the model.

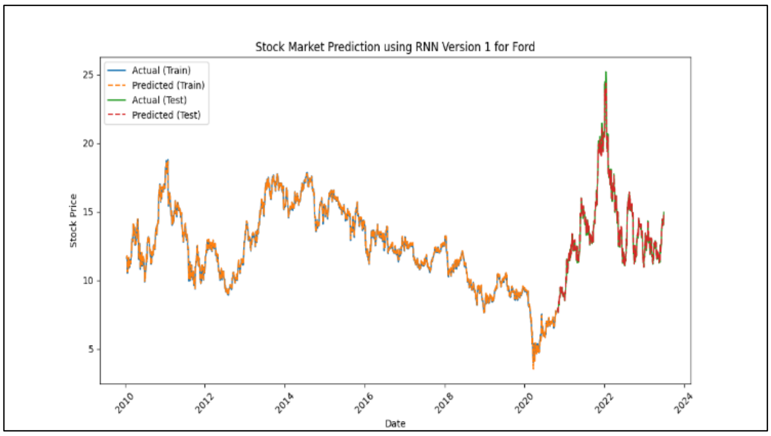

Fig - 15: RNN Version 1 (2010-2023)

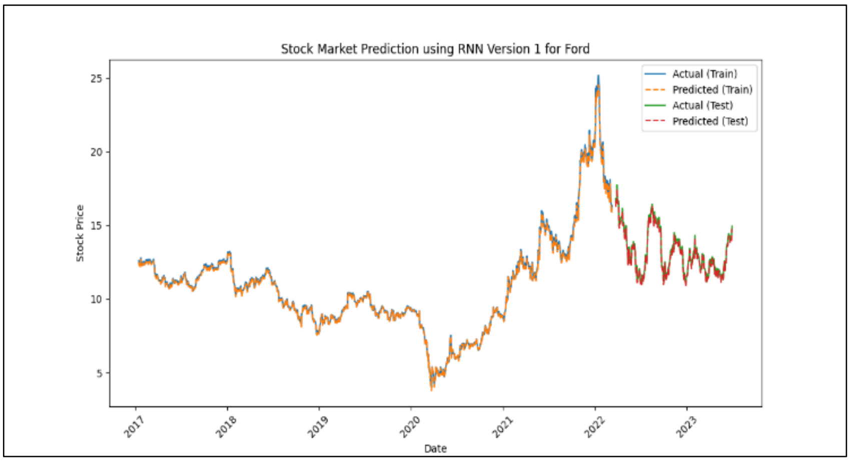

Fig - 16: RNN Version 1 (2017-2023)

While applying RNN model to find the values of RMSE the dataset of Ford from (2010 – 2023) and (2017-2023) best fits the model as shown in the graph below.

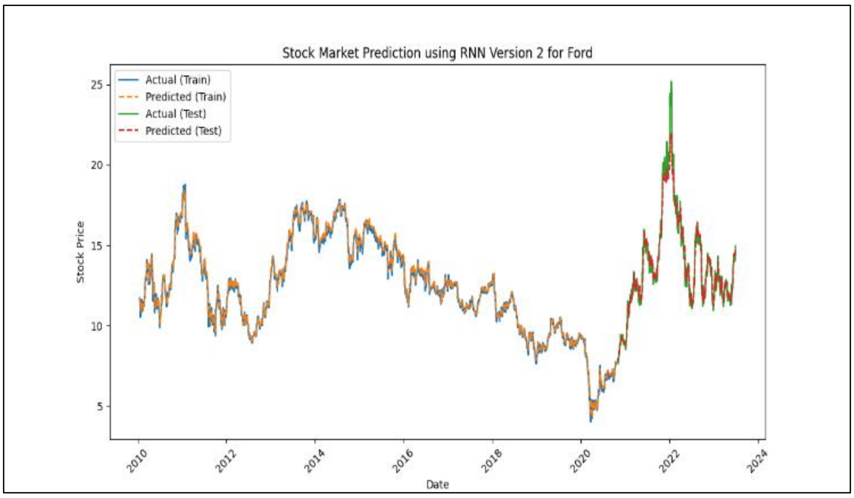

Fig - 17: RNN Version 2 (2010-2023)

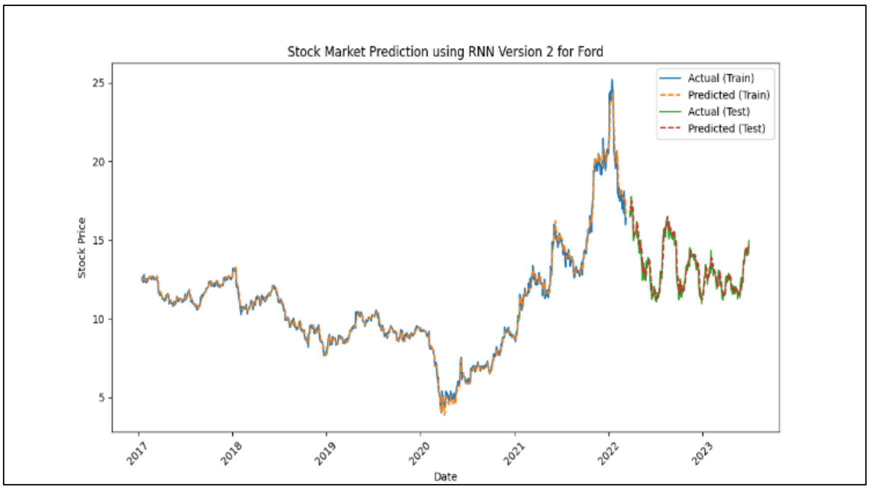

Fig - 18: RNN Version 2 (2017-2023)

Table 11. RNN Model (2010-2023)

Company	**Testing MSE**	
	RNN (V1)	**RNN (V2)**
Apple	16.3891	17.568
Amazon	1.3723	20.7802
Ford	0.01277	0.24533
Google	79.852	252.941
Toyota	19.2724	95.238

Table 12. RNN Model (2017-2023)

Company	**Testing MSE**	
	RNN (V1)	**RNN (V2)**
Apple	1.91774	14.5050
Amazon	0.3312	15.3650
Ford	0.01930	0.16646
Google	0.15224	11.9866
Toyota	0.0938	1.58994

Table 13. RNN Model (2010-2023)

Company	**Testing RMSE**	
	RNN (V1)	**RNN (V2)**
Apple	4.04835	4.19148
Amazon	1.17146	4.5585
Ford	0.11302	0.49531
Google	8.93604	15.9041
Toyota	4.3900	9.75904

Table 14. RNN Model (2017-2023)

Company	**Testing RMSE**	
	RNN (V1)	**RNN (V2)**
Apple	1.38482	3.8085
Amazon	0.57552	3.91983
Ford	0.13895	0.40799
Google	0.39018	3.46217
Toyota	0.30627	1.26092

5.4 PROPHET:

Prophet, developed by Facebook, is tailored to capture recurring patterns, seasonality, and holidays commonly observed in time series data. Our dataset, spanning from 2010 to 2023 and 2017 to 2023 provided an ideal platform to evaluate the Prophet model's predictive capabilities. To assess the Prophet model's performance, we leveraged Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) as quantitative measures of accuracy. These metrics enabled us to gauge the disparities between Prophet-predicted values and the actual data points.

Our experimental findings highlighted the effectiveness of the Prophet model in time series forecasting. The model demonstrated notable strength in capturing seasonal patterns and holiday effects present in the data. Models trained using Prophet exhibited relatively low MSE and RMSE values, indicating a favorable alignment between predicted and actual values. Furthermore, the visual comparison of the Prophet model's forecasts against the actual data highlighted its capability to accurately represent the underlying trends, periodicities, and holiday-related fluctuations within the time series.

In summary, our exploration into the application of the Prophet model underscored its efficacy for time series forecasting tasks. Designed to accommodate seasonal and holiday variations, Prophet provides a robust tool for capturing recurring patterns within time series data. While acknowledging the model's specialization, our results emphasize its relevance and potential contributions to the field of time series analysis and forecasting.

According to my finding, the MSE and RMSE values calculated over different dataset from (2010 - 2023) and (2017 - 2023) using PROPHET model was not much efficient but gives better results over Ford dataset.

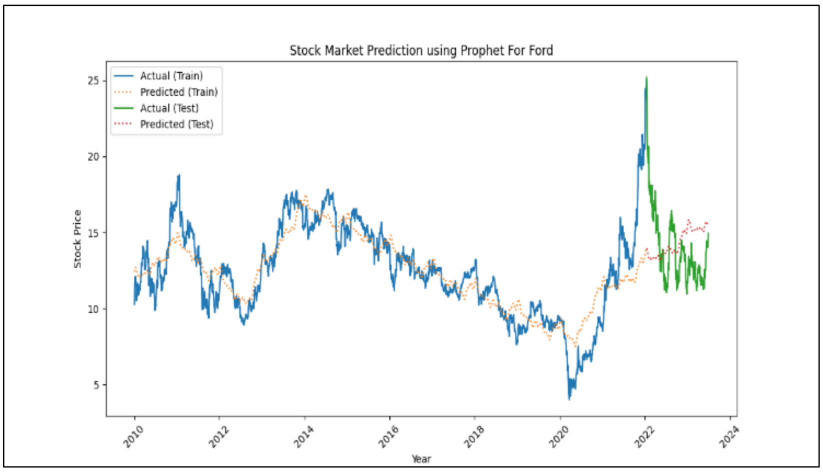

Fig - 19: PROPHET Model (2010-2023)

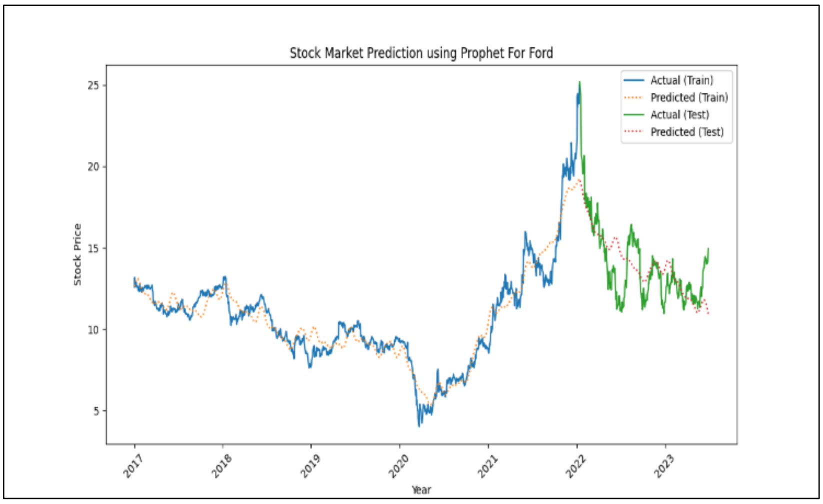

Fig - 20: PROPHET Model (2017-2023)

Table 15. PROPHET Model (2010-2023)

Company	Testing MSE
Apple	225.7855
Amazon	286.0398
Ford	8.30582
Google	285.8343
Toyota	280.7308

Table 16. PROPHET Model (2017-2023)

Company	Testing MSE
Apple	194.0390
Amazon	239.8570
Ford	1.56403
Google	126.7202
Toyota	74.2557

Table 17. PROPHET Model (2010-2023)

Company	Testing RMSE
Apple	15.02616
Amazon	16.9127
Ford	2.88198
Google	16.9066
Toyota	16.7550

Table 18. PROPHET Model (2017-2023)

Company	Testing RMSE
Apple	13.9297
Amazon	15.48731
Ford	1.56403
Google	11.2570
Toyota	8.61717

5.5 Long Short-Term Memory (LSTM):

LSTMs, a specialized type of Recurrent Neural Networks (RNNs), are designed to overcome the limitations of capturing long-term dependencies within sequential data. Our dataset, spanning from 2010 to 2023 and 2017 to 2023, served as a platform to assess the LSTM model's predictive capabilities. To evaluate the LSTM model's performance, we utilized the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) metrics to quantitatively measure predictive accuracy. These metrics enabled us to assess the deviations between LSTM-generated predictions and the actual data points.

Our experimental outcomes underscored the effectiveness of LSTM networks in time series forecasting. The model demonstrated remarkable prowess in capturing intricate temporal dependencies, even in datasets with complex patterns. Models trained using LSTMs exhibited relatively low MSE and RMSE values, indicating a strong alignment between predicted and actual values.

Moreover, visual examination of the LSTM model's forecasts in com0parison to the actual data reaffirmed its ability to capture both short-term fluctuations and long-term trends within the time series.
In summary, our exploration into the application of Long Short-Term Memory networks for time series forecasting revealed their exceptional capacity to capture complex patterns and dependencies. LSTMs offer a promising avenue for accurately modeling sequential data, paving the way for improved insights and predictions in diverse time series analysis contexts.

According to my finding, I have applied LSTM model to different dataset spanning from (2010 - 2023) and (2017-2023) but the MSE and RMSE values calculated shows that Ford dataset best fits the model.

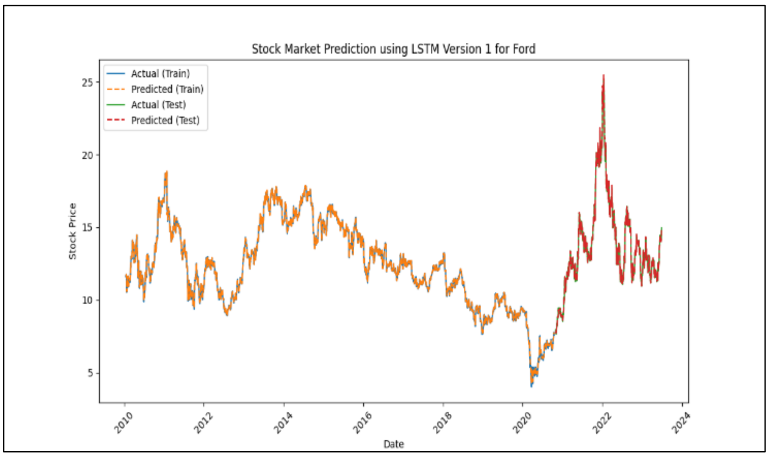

Fig - 21: LSTM Version 1 (2010-2023)

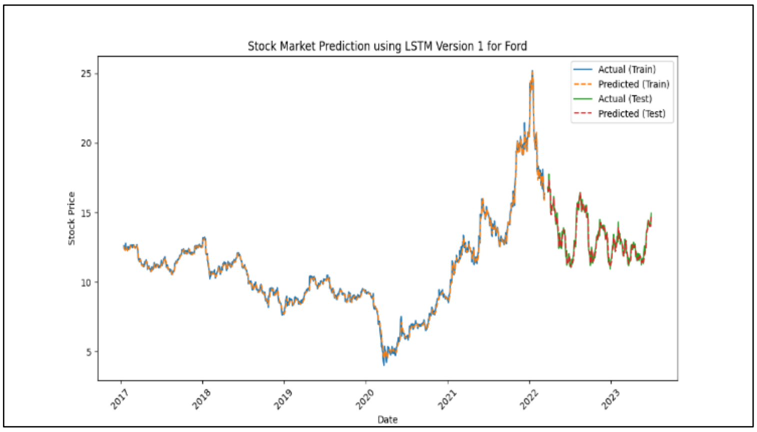

Fig - 22: LSTM Version 1 (2017-2023)

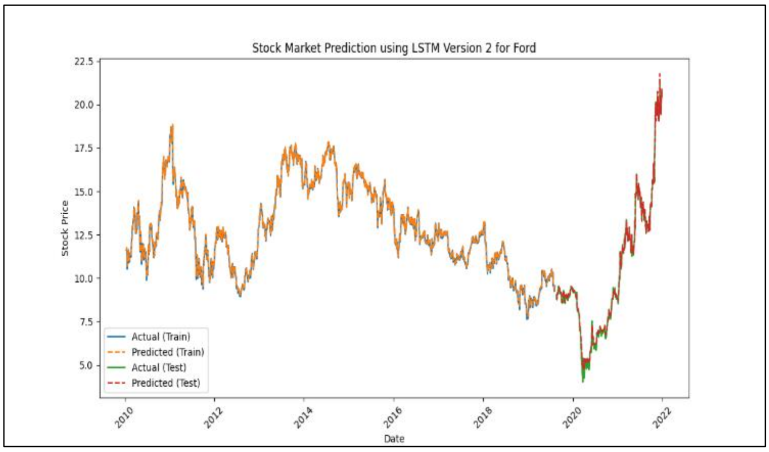

Fig - 23: LSTM Version 2 (2010-2023)

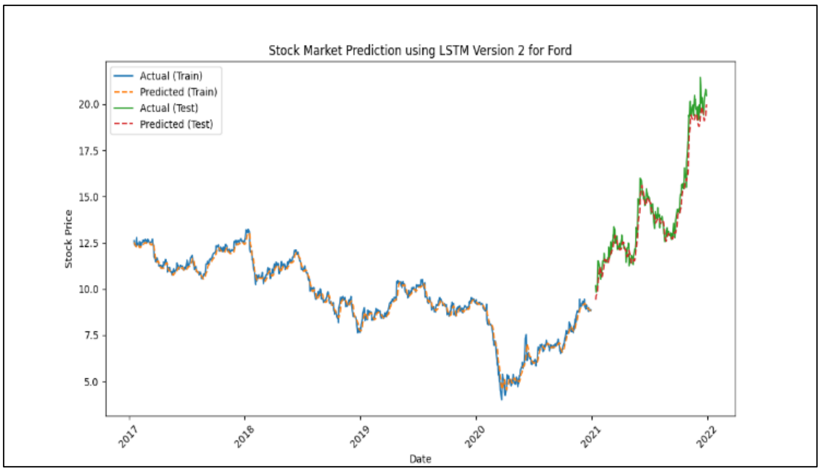

Fig - 24: LSTM Version 2 (2017-2023)

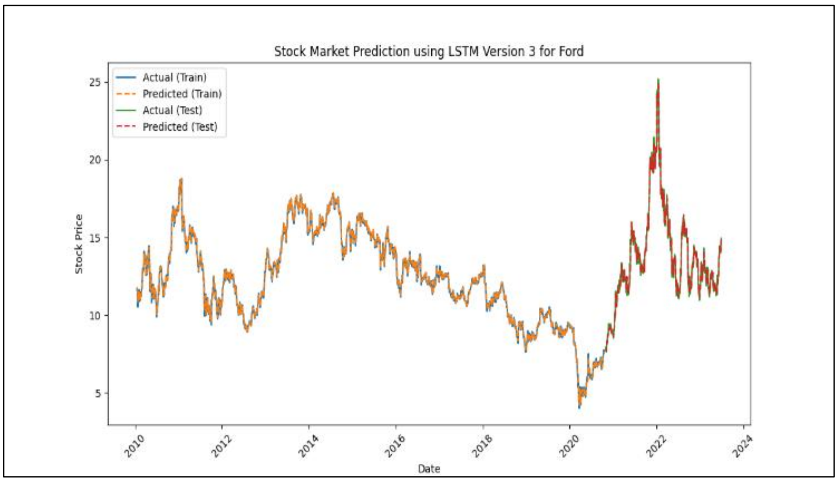

Fig - 25: LSTM Version 3 (2010-2023)

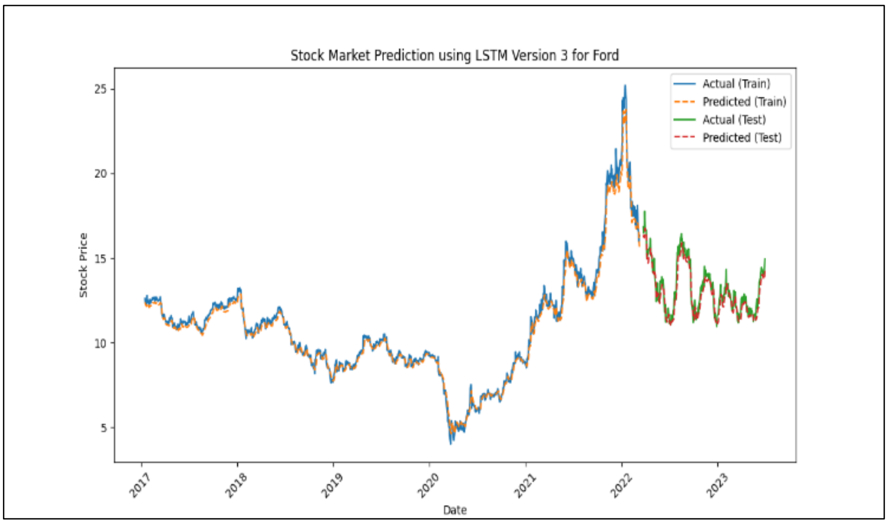

Fig - 26: LSTM Version 3 (2017-2023)

Table 19. LSTM Model (2010-2023)

Company	Testing MSE		
	LSTM (V1)	**LSTM (V2)**	**LSTM (V3)**
Apple	9.53552	2.6815	14.44498
Amazon	2.68160	7.7269	7.72693
Ford	0.01930	0.0253	0.01975
Google	0.00759	4.5063	19.0316
Toyota	1.05301	0.4356	29.7705

Table 20. LSTM Model (2017-2023)

Company	Testing MSE		
	LSTM (V1)	**LSTM (V2)**	**LSTM (V3)**
Apple	30.97592	48.1935	17.4784
Amazon	20.85952	7.72693	25.7527
Ford	0.06180	0.30745	0.2054
Google	2.67476	9.77921	25.6044
Toyota	0.37151	4.96554	13.6840

Table 21. LSTM Model (2010-2023)

Company	Testing RMSE		
	LSTM (V1)	**LSTM (V2)**	**LSTM (V3)**
Apple	3.08796	1.6375	3.8006
Amazon	1.63756	2.7797	2.7797
Ford	0.08712	0.1592	0.1405
Google	1.37207	2.1228	4.3625
Toyota	1.02616	0.6600	5.4562

Table 22. LSTM Model (2017-2023)

Company	Testing RMSE		
	LSTM (V1)	**LSTM (V2)**	**LSTM (V3)**
Apple	5.56560	6.9421	4.1807
Amazon	4.56722	2.7797	5.0747
Ford	0.24860	0.5544	0.4532
Google	2.67476	3.1271	5.0600
Toyota	0.60952	2.2283	3.6991

6. Conclusion and Future Scope

This paper explores various methodologies employed for predicting stock index values and movement trends. The study incorporates different deep learning models, specifically ARIMA, CNN, RNN, LSTM and PROPHET. These models are applied to the Yahoo's Finance dataset. Navigating the intricacies of the stock market is indeed a formidable task. Predicting future stock prices predominantly resides within the technical realm, while assessing securities evaluations relies on statistical insights derived from market activities. Selecting the most suitable model for stock price prediction among CNN, RNN, LSTM, ARIMA, and PROPHET hinges on several considerations. Each model possesses distinct advantages and drawbacks.

The LSTM model's notable advantage over the other deep learning models, especially RNN can be due to its unique architecture, which tackles the drawbacks of traditional RNNs. The key elements that support LSTMs' improved performance include their prowess in managing long-term dependencies within sequential data. By including specialized memory cells and gating mechanisms, LSTMs overcome the vanishing gradient problem that plagues traditional RNNs. These processes make it easier to retain and incorporate specific information across lengthy periods, enabling LSTMs to effectively grasp persistent dependencies that conventional RNNs frequently fail to recognize. This advantage is strengthened by their deft use of gating mechanisms, including as the forget gate, input gate, and output gate, which deftly control the information flow. Their effectiveness in a variety of time series forecasting tasks, including the complex task of stock price prediction, is a result of the equilibrium between short-term fluctuations and long-term trends. The end result is a model that is excellent at managing extended dependencies and offers a potential approach for accurately capturing complex temporal patterns.The research has clearly validated our hypothesis that models based on deep learning exhibit a significantly heightened ability to comprehend and capture the inherent attributes of time series data when compared to their corresponding counterparts in machine learning.

References

[1] M. Nabipour, P. Nayyeri, H. Jabani and A. Mosavi, "Deep learning for Stock Market Prediction," Dresdan, 2020.

[2] A. Sarkar, A. K. Sahoo, S. Sah and C. Pradhan, "LSTMSA: A Novel Approach for Stock Market Prediction Using LSTM and Sentiment Analysis," 2020.

[3] X. Zhong and D. Enke, "Predicting the daily return direction of the stock market using hybrid machine learning algorithms," Financial Innovation, 2019.

[4] D. Shah, W. Campbell and F. H. Zulkernine, "A Comparative Study of LSTM and DNN for Stock Market Forecasting," in *IEEE International Conference on Big Data (Big Data)*, 2018.

[5] H. M, G. E.A, V. K. Menon and S. K.P, "NSE Stock Market Prediction Using Deep-Learning Models," in *International Conference on Computational Intelligence and Data Science (ICCIDS 2018)*, Coimbatore, 2018.

[6] M. Vijh, D. Chandola, V. A. Tikkiwal and Kumar, "Stock Price Prediction using LSTM, RNN and CNN-Sliding Window Model," in *International Conference on Computational Intelligence and Data Science*, Coimbatore, 2017.

[7] M. Agarwal, P. K. Shukla, R. Nair, A. Nayyar and M. Masud, "Stock Prediction Based on Technical Indicators Using Deep Learning Model," Tech Science Press, Bhopal, 2022.

[8] S. T. Bahadur, A. Shrestha, A. Neupane and W. Guo, "Stock Price Forecasting with Deep Learning," *E Mathematics by MDPI*, vol. 1, no. 1, p. 15, 2020.

[9] W. Bao, J. Yue and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," 2017.

[10] K. Pawar, R. S. Jalem and V. Tiwari, "Stock Market Price Prediction Using LSTM RNN," 2019.

[11] K. A. Althelaya, E.-S. M. El-Alfy and v. Mohammed, "Evaluation of Bidirectional LSTM for Short- and Long-Term Stock Market Prediction," in *9th International Conference on Information and Communication Systems (ICICS)*, Irbid, 2018.

[12] S. K. Lakshminarayanan and J. McCrae, "A Comparative Study of SVM and LSTM Deep Learning Algorithms for Stock Market Prediction," 2019.

[13] O. B. Sezer, M. U. Gudelek and A. M. Ozbayoglu, "Financial time series forecasting with deep learning : A systematic literature review: 2005–2019," in *Applied Soft Computing Journal*, Ankara, 2020.

[14] A. S. Saud and S. Shakya, "Analysis of look back period for stock price prediction with RNN variants: A case study on banking sector of NEPSE," in *International Conference on Computational Intelligence and Data Science (ICCIDS 2019)*, Nepal, 2020.

[15] S. A. Dwivedi, A. Attry, D. Parekh and K. Singla, "Analysis and forecasting of Time-Series data using S-," in *International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2021.

[16] V. Ingle and S. Deshmukh, "Ensemble deep learning framework for stock market data prediction," in *Global Transitions Proceedings*, Aurangabad, 2021.

[17] J. Shen and M. O. Shafq, "Short-term stock market price trend prediction using a comprehensive deep learning system," *Journal of Big Data*, p. 33, 2020.

[18] S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia and D. C. Anastasiu, "Stock Price Prediction Using News Sentiment Analysis," in *IEEE Fifth International Conference on Big Data Computing Service and Applications*, 2019.

[19] A. H. Bukhari, M. A. Z. Raja, M. Sulaiman, S. Islam, M. Shoaib and P. Kumam, "Fractional Neuro-Sequential ARFIMA-LSTM for Financial Market Forecasting," *IEEE Access*, p. 14, 2020.

[20] G. Ding and L. Qin, "Study on the prediction of stock price based on the associated network model of LSTM," *International Journal of Machine Learning and Cybernetics*, vol. 11, p. 11, 2020.

[21] M. Obubu, "ARIMA Model Building," ResearchGate, [Online]. Available: https://www.researchgate.net/figure/Fig-1-The-steps-of-ARIMA-model-building-methodology-is-presented-in-a-flow-chart-in_fig1_334742152.

[22] A. Kumar, "Different Types of CNN Architectures Explained: Examples," Data Analytics, 2023 August 2023. [Online]. Available: https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/.

[23] J. Ma, "All of Recurrent Neural Networks," Medium, 2 April 2016. [Online]. Available: https://medium.com/@jianqiangma/all-about-recurrent-neural-networks-9e5ae2936f6e.

[24] E. Kang, "Long Short-Term Memory (LSTM): Concept," Medium, 2 September 2017. [Online]. Available: https://medium.com/@kangeugine/long-short-term-memory-lstm-concept-cb3283934359.

[25] W. Team, "Wallstreetmojo," Wallstreetmojo, 2022. [Online]. Available: https://www.wallstreetmojo.com/stock-market/.

[26] J. Allibhai, "Building A Deep Learning Model using Keras," Medium, 17 September 2018. [Online]. Available: https://towardsdatascience.com/building-a-deep-learning-model-using-keras-1548ca149d37.