Stock Market Prediction Using Encoder Decoder ConvLSTM

Author

KHURUM IQBAL

MS-18-CE-00000273957

Supervisor

DR. ALI HASSAN

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

March, 2022

Stock Market Prediction Using Encoder Decoder ConvLSTM

Author

KHURUM IQBAL

MS-18-CE-00000273957

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Computer Engineering

Thesis Supervisor

Dr. Ali Hassan

Thesis Supervisor's Signature: ____

DEPARTMENT OF COMPUTER ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, ISLAMABAD

MARCH, 2022

Declaration

I certify that this research work titled "*Stock Market Prediction Using Encoder Decoder ConvLSTM*" is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources has been properly acknowledged/referred.

Signature of Student

KHURUM IQBAL

MS-18-CE-00000273957

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical, and spelling mistakes. The thesis is also according to the format given by the university.

Signature of Student

KHURUM IQBAL

MS-18-CE-00000273957

Copyright Statement

Acknowledgments

All praise and glory to Almighty Allah (the most glorified, the highest) who gave me the courage, patience, knowledge, and ability to carry out this work and to persevere and complete it satisfactorily. Undoubtedly, HE eased my way and without HIS blessings I can achieve nothing.

I would like to express my sincere gratitude to my Thesis Supervisor Dr. Ali Hassan and Co-Supervisor Dr. Ahsan Shahzad for boosting my morale and for his continual assistance, motivation, dedication, and invaluable guidance in my quest for knowledge. I am blessed to have such a cooperative advisor and kind mentor for my research.

Along with my advisor, I would like to acknowledge my entire thesis committee: Dr. Farhan Riaz, Dr. Farhan Hussain and Dr. Umar Farooq for their cooperation and prudent suggestions.

My acknowledgment would be incomplete without thanking the biggest source of my strength, my family. I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout every department of my life. I also want to say thanks to my supportive Brothers, Sister, who were with me through my thick and thin.

Finally, I would like to express my gratitude to all my friends and the individuals who have encouraged and supported me through this entire period.

Dedicated to my exceptional parents, supportive brothers, and Sisters whose tremendous support and cooperation led me to this accomplishment. In the end, this thesis is dedicated to all those who believe in the richness of learning

Abstract

Stock market prediction is a hot topic these days and predicting the price of a stock is both difficult and important due to the numerous variables at play. There were numerous Machine Learning models used for Stock Market Prediction, but Hybrid Models were successful in making accurate predictions. The goal of this study is to create a hybrid Deep Learning model (Encoder-Decoder ConvLSTM) to predict stock market prices. We used historical stock price data from the Standard and Poor (S&P 500) from Yahoo's finance website. This dataset consisted of daily values with different features i.e., open price, high price, closing price, adj close, and volume. We also used a six-month dataset from the State Bank of Pakistan (SBP). The dataset included closing prices of different currency exchanges. In this research AUDUSD currency exchange has been used to predict the closing price of the next hour. Different prediction models have been tested on the S&P 500 dataset and their results have been compared with the proposed model, and the proposed model has been applied to the SBP dataset as well after it was discovered that it performed well on the publicly available dataset. To determine the effectiveness of the proposed model, we used the following performance metrics, root means square error (RMSE), mean absolute error (MAE), mean square error (MSE), and mean absolute percentage error (MAPE). Experiments indicate that the proposed model has the best performance metrics values when compared to other comparable studies, as the previous results for the S&P 500 dataset for RMSE, MAE, and MAPE using the traditional LSTM model was 16.471, 11.40, and 0.53. Similarly, in another study using the Deep LSTM model the experiments were performed on the same dataset, and the best results for performance metrics RMSE and MAPE were 8.416 and 0.143 respectively. In comparison with these results, the proposed model performed well and the results for RMSE, MAE, and MAPE which have been collected using the same dataset are 4.0471, 2.662, and 0.282. As a result, we can conclude that our model is suitable for the accurate prediction of the stock market in general.

Keywords—Stock Market Prediction, Long Short Term Memory (LSTM), Encoder-Decoder ConvLSTM, Time Series, Performance Metrics.

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Introduction of ANN Models:

Stock price prediction is an unpredictable problem that is simulated using machine learning to forecast stock returns. There are several ways and instruments available for stock market prediction. The stock market is thought to be extremely active and complicated. Accurate forecasting of future prices may result in a greater profit return for stockholders through stock purchases. According to predictions, stockholders will be able to select equities that will provide a better return. Forecasting may be divided into two categories: explanatory (causal) and time series [1]. Explanatory forecasting presupposes that the inputs and outputs have a cause and effect connection as altering the system's inputs will have a predictable influence on the system's output, providing the cause and effect connection is consistent. Time-series forecasting, as opposed to explanatory forecasting, examines the system as a black box and attempts to uncover the elements influencing the behavior. There are two reasons why a framework ought to be treated as a black box [1]. The primary problem may be simply predicting what will happen and what will not happen. So the monetary time series is one of the most un-comprehended areas that has been under assessment for quite a while for notable reasons (to improve investment returns) as well as forecasting the performance of the stock market is difficult owing to numerous factors. For many decades, Time series forecasting has been a subject of research, even though its intricacy, dynamism, and chaos have proven to be a difficulty [3]. As a result, time-series foretelling is the emphasis of this work.

Time series data has natural ordering which distinguishes time series examination and cross-sectioned research, which has no natural ordering (for example, explaining incomes concerning their various education stages, where statistics may be put in any form). Time series data differs from spatial data in that the observations are generally related to physical places as well as they can be used on real-valued. Furthermore, they frequently make use of time's intrinsic one-way ordering, such that values for a particular period are generated from past values, not from future values [2]. With the growing amount of data and more precise prediction expectations, deep learning models are now being employed, which have shown to be superior to standard machine learning approaches in terms of accuracy and prediction speed [26]. Some common models which have been utilized for time series anticipating are Autoregressive Moving-Average (ARMA), and the most popular models are Long Short Term Memory (LSTM) to estimate the stock market future prices [4] [5]. The LSTM network is not the same as a traditional Multilayer

Perceptron (MLP) network. The network, like an MLP, is made up of layers of neurons. To create a forecast, input data is transmitted through the network.

New methods are being deployed, and their advantage over conventional machine learning approaches is proven in terms of accuracy and speed of prediction. Here, we will examine the Long-Short-Term Memory (LSTM), which is widely used in the stock market. This task involves automating stock data retrieval via Python libraries, training the LSTM model on that data, and using it to predict future stock prices. Financial forecasting has been a subject of interest for academics for a long time and is particularly important in the stock market. The premise is that public information in the past correlates with future stock performance [1]. Stock price prediction is an unpredictability problem that is forecasted using machine learning. There are numerous methods and instruments available for stock market forecasting. The stock market is thought to be both active and complex. Accurate forecasting of future prices may result in a higher profit return for stock purchases. Investors, according to forecasts, will be able to select equities that will provide a higher return. Forecasting can be divided into two types: explanatory (causal) forecasting and time series forecasting [1]. Explanatory forecasting assumes that the system's inputs and outputs have a consistent cause and effect relationship, as changing the system's inputs will have a predictable effect on the system's output. In contrast to explanatory forecasting, time-series forecasting examines the system as a black box and attempts to uncover the elements influencing the behavior. A system should be treated as a black box for two reasons [1]. First, the system may not be understood, and even if it is, measuring the relationships that are supposed to control its behavior may be extremely difficult. Second, the primary issue may simply be predicting what will and will not happen. [3]. As a result, the emphasis of this work is on time series forecasting. Time series data has natural ordering, which distinguishes it from cross-sectional research, which lacks natural ordering (for example, explaining incomes concerning their various education levels, where data may be put in any form). Time series data differs from spatial data in that the observations are generally related to physical locations and can be used on a real-valued scale. Furthermore, they frequently exploit time's inherent one-way ordering, so that values for a given period are generated from past values rather than future values [2]. As various machine learning approaches have been used in stock market prediction over the years, stock market forecasting is one of the most popular models for forecasting time series data. Deep learning models, which have shown to be superior to standard machine learning approaches in terms of accuracy and prediction speed, are now being used with the growing amount of data and more precise prediction expectations. Autoregressive Moving-Average (ARMA) is a popular time series forecasting model, and Long Short Term Memory (LSTM) is the most popular model for forecasting stock market future prices [4] [5]. A traditional Multilayer Perceptron (MLP) network is not the same as an LSTM network. The network is made up of layers of neurons, similar to an MLP. To create a forecast, input data is transmitted over the network.

The goal of this study is to develop the most accurate stock market prediction model possible, as contrasted to other models. It has been hypothesized that the proposed model encoder-decoder ConvLSTM has performed well and generated the best results when compared to the relative prediction models. Two datasets for this research Standard and Poor (S&P 500) which is the publicly available dataset that is obtained from Yahoo Finance and the other dataset of State Bank of Pakistan (SBP) have been used. To measure the effectiveness different performance measures have been used and compared the results with previous research undertaken.

1.2 Motivation

The main reason for attempting to predict the stock market is to make money. It would be lucrative for a financial trader to uncover a model that consistently forecasts stock prices. So, because researchers, investors, and investment professionals are always looking for a stock market model that will earn them more than their peers, they seek a better return on investment. Even the most skilled day trader finds it challenging to predict the movement of stocks in the competitive financial markets. Even in a matter of seconds, the price of a stock can dramatically rise or fall, so whoever can see it first and act upon it will be rewarded with a fortune, while the rest will suffer financial ruin. While stock market speculation is an imperfect science, a variety of methods have been used over the years in an attempt to gain an edge and turn a profit.

1.3 Aims and Objectives

The major objectives of the research are as follow:

- To use the Neural Networks for forecasting the future prices of the stock market.
- To construct a profitable financial trading strategy that uses the LSTM network's predictive capabilities
- To show the comparisons of different LSTM models and find the best model for forecasting Stock Market.
- LSTM Neural Networks can compete with conventional time series models.

1.4 Problem Definition

The research will be performed on the S&P500 index and its components, and by using it the next day closing price of the stock market will be predicted It will also be applied to the State Bank of Pakistan dataset and the AUDUSD Exchange rate for the next day will be predicted. It's vital to emphasize that both the regression and classification models, which predict future values based on previously predicted values, are accurate.

1.5 Publications

- Forecasting stock market using machine learning approach Encoder-Decoder ConvLSTM
  (K. Iqbal, A. Hassan, S. S. M. U. Hassan, S. Iqbal, F. Aslam and K. S. Mughal, "Forecasting Stock Market Using Machine Learning Approach Encoder-Decoder ConvLSTM," *2021 International Conference on Frontiers of Information Technology (FIT)*, 2021, pp. 43-48, doi: 10.1109/FIT53504.2021.00018)

- Time-series Prediction of Cryptocurrency Market Using Machine Learning Techniques
  (Iqbal, Mahir & Iqbal, Shuaib & Hassan Jaskani, Fawwad & Iqbal, Khurum & Hassan, Ali. (2021). Time-Series Prediction of Cryptocurrency Market using Machine Learning Techniques. EAI Endorsed Transactions on Creative Technologies. 4. 6. 10.4108/eai.7-7-2021.170286.)

1.6 Structure of the Report

The rest of the thesis is structured as follows:

Chapter 2 discusses the different analyses of the stock market and the various stocks that have been conducted.

Chapter 3 describes the information about stocks and datasets that we used in our research.

Chapter4 describes details of different models' architectures that were utilized in this research.

Chapter 5 shows the results and experiments.

Chapter 6 conclude the thesis with future recommendations.

CHAPTER 2 : LITERATURE REVIEW

In addition to the stock market's intricacy and vibrancy, stock price prediction has been the subject of much discussion due to its significant financial rewards it has attracted researchers for a long time. Among the several approaches used thus far, the most commonly used system is the Artificial Neural Network (ANN) developed by Verma et al. [6]. The over-fitting problem mostly affects ANNs. Furthermore, Support Vector Machines (SVMs) can be used to avoid such an overfitting problem [7]. Usmani et al. predict the trend of the Karachi Stock Exchange (KSE) by suggesting the major focus of this investigation on day closing using several machine learning algorithms [8]. To forecast stock prices, they used classic statistical models such as ARIMA and SMA. According to Fatma's premise, an asset's present price resembles all previously accessible information. According to the study [9], the random-walk hypothesis also states that stock price fluctuations do not influence its history. This indicates that regardless of the price today, the next day's price will only be determined by the next day's information. This demonstrates the limitation of accurate methods for forecasting the price of a stock. Furthermore, Biondo's experiment study demonstrates how a random technique may outperform some of the most traditional technical trading strategies, such as the Moving Average Convergence Divergence (MACD) and Relative Strength Index (RSI) [10]. Different types of forecasting are classified as short-term, medium-term, and long-term forecasting and various models such as MLP, Support Vector Machine (SVM), Radial basic Function (RBF) as well as Single Layer Perceptron (SLP) has been used for these forecasting. When compared against other techniques, the MLP algorithm fared the best [8]. A study of ANNs for predicting Chinese stock has also been performed in [18]. RNNs [19, 20], LSTM [21, 22], and reinforcement learning (RL) [23] techniques have been utilized in stock price forecasting. [24] also discusses stock market forecasting using Generative Adversarial Network (GAN). Throughout the previous few decades, various financial institutions have made use of Stock market forecasts using artificial intelligence (AI), and deep learning, one of the AI approaches, has turned into a significant and prominent tool in financial market analysis which produced several encouraging results in stock price forecasting that is nonlinear, multivariate, and data-driven [11]-[13]. RNNs are a subset of ANNs that are designed to learn time-varying or sequential patterns. The LSTM recurrent neural network is a unique and significant RNN design that excels in remembering values for either short or long periods [14]. Jia [15] studied the efficacy of LSTM for time series stock market forecasting. When compared to other models, this deep learning model produces the greatest results. A deep learning algorithm for forecasting the stock market using events and practical indicators has been unveiled. To enhance the prediction model, they employed a convolutional neural network (CNN) and an LSTM [16]. The prediction performance results were enhanced by using this combo model. [17] proposes a novel investing approach based on a RoboAdvisor (RA) using deep learning market prediction. Daily price forecasting on S&P and DJI has also been proposed in [29,30] and different performance metrics have been utilized to evaluate the effectiveness of models. The RA system described above includes market prediction, automatic investing, and fund-selection techniques. The findings show that it best results for market prediction using S&P data when compared to other prediction models. In reference [31], the authors developed a flexible yet reliable statistical model to anticipate meteorological conditions in the vicinity of an Indonesian airport. They also used single and multilayer LSTM to investigate the impact of weather on airplane departure and takeoff. Persio et al. [32] demonstrated the appropriateness and proficiency of introducing LSTM to financial time series forecasting. Using this paradigm, the gate layers have direct access to the state of the cell's internal architecture. In another instance, the model included features such as forget and input gates, among other things. This is a situation in which all parties involved must agree on whether new information should be included or ignored. If something else has to be entered in its place, it only forgets about the previous entry. When it erases anything older than the current values, this design adds new values to the cell's state as a result of the erase operation. The Gated Recurrent Unit, a popular LSTM variant developed by the authors of the reference [33], is described in detail in [33]. (GRU). It is possible to create a single "update gate" by combining the forget and input gates into a single unit. To make the final model more fundamental than the original LSTM, the cell state and hidden state are combined, as well as a few other minor modifications. As a result of the factors mentioned above, this approach is becoming increasingly popular. These aren't your typical ‘small potatoes,' as the saying goes. RNN, CNN, and LSTM were the deep learning network architectures used by Hiransha et al. to estimate stock prices based on day-wise previous closing prices in their study [34], which was published in Nature Communications. The RNN architecture was the most widely used. As part of the experiment, they selected two companies from the information technology industry (TCS & Information Systems) and one from the pharmaceutical industry to take part in it (Cipla). The study is unique in that it trained the models using data from a single company, which makes it unique in the field. Next, the models were used to forecast the future values of five different NSE and NYSE stocks, each of which had a different price history (Newyork Stock Exchange). On the other hand, they assert that deep networks uncover the fundamental dynamics of stock prices, whereas linear models attempt to fit data to the model in consideration. They discovered that CNN outperforms all other models, including conventional linear models, according to their findings. Despite learning from the NSE dataset, the DNN was able to predict the behavior of companies that were listed on the NYSE. Some speculate that this is because both stock markets have internally similar dynamics, which could explain their similarities.

2.1 Background and Technical Analysis

The stock market is a complex system because of its nonlinear and nonstationary characteristics [35]. The stock market's complexity is linked to a variety of factors, according to Ticknor, including political events, market news, quarterly earnings releases, worldwide impact, and contradictory trading behavior. The former holds that the price of a company's share is solely determined by the interaction of demand and supply factors [36]. Mathematic-statistical modeling has been used in technical analysis as a collection of methods for predicting future returns in financial assets by focusing on previous market data, primarily stock price and volume.

2.2 Time Series Based Prediction

As described in Section 2.1, technical analysis is the practice of making trading decisions using past stock market data. Using historical stock information can increase above-average returns without increasing the risk of loss. Regardless of the contradictory hypothesis, the authors in [37] demonstrated that this assumption is heavily used in the investment sector. Park and Irwin's (2004) meta-analysis discovered that a majority of technical analysis articles found profitable returns, contradicting the EMT. However, we must bear in mind that the possibility of a publication bias exists, and such studies should be read with caution. According to [34], using the stock market data for training a single-layer neural network model offers a statistically significant advantage when applied to a collection of randomly generated data and a small set of historical stock prices. Despite the availability of extensive stock market prediction research in the decades preceding the 2010s, artificial neural networks have remained scarce until recently. Batres-Estrada used a stacked RBF followed by a feedforward ANN with one hidden layer, which was highly influenced by Takeuchi and Lee [38]. Using the last year's worth of stock price return data as the input, this algorithm forecasts the binary trend of the following month's price with the addition of daily price return data. A test set accuracy of 52.89% is achieved through this method, which bests naive baselines and basic logistic regression and is comparable to Takeuchi and Lee. Dixon et al. [39] experimented with a feedforward artificial neural network using five hidden layers for trinary classification, which they say is similar to prior studies. A single model, trained on CME-listed commodities and foreign exchange futures data, generates a range of engineered features, such as moving correlations, instead of having a model for each target instrument. The average accuracy of 42.0 percent for the three-class prediction task was obtained from this. The economic conclusions should be validated by further research, however, and no cross-validation is performed.

2.3 Prediction based on text analysis

Even though the experiments in this thesis do not include alternative stock market prediction approaches, another computational approach to this problem should be included if this chapter is to provide a comprehensive summary of current stock market prediction trends. Understanding the numerous implications for economic theory in terms of empirical evidence for the usefulness of various techniques is also required, as is the establishment of an entirely novel benchmark against which future comparisons can be made. Text-based prediction techniques based on machine learning models have gained popularity in recent years as a preferred method to supplement the relatively limited literature on deep learning for stock market prediction based on time series. Lavrenko et al. proposed using news items, which provide new information rather than historical data, to predict stock values. Since then, the concept has become a popular starting point for future research. AZFinText, a method developed by Schumaker and Chen [40], has received a lot of attention because it achieves a directional accuracy of 57.1 percent for the best-performing model. This result was produced using a support vector machine with a proper nouns scheme rather than a basic bag-of-words method and the current price of a stock, with news articles and stock prices from five weeks as inputs. The basic argument is that five weeks of data may not be enough to conduct a thorough evaluation of performance. This is a legitimate point of contention. The experiment also failed after only 20 minutes, which is within the time frame of a previous study that discovered the majority of market responses to new information occurred within a ten-minute time frame (Patell and Wolfson, 1984). Ding et al. [41] propose using a neural tensor system to extract event descriptors from financial news stories, which would then be fed into a deep convolutional neural network to produce a two-class prediction of a stock's future movement. This method has a 65.9 percent accuracy rate for 15 different S&P 500 stocks and daily trend predictions. On the other hand, no clear evidence exists as to how these reported stocks were chosen. Fehrer and Feuerriegel (2015) used recursive autoencoders to extract emotions from financial news headlines and businesses' financial disclosure statements for the test set and stock price forecasts following a financial disclosure statement, achieving an accuracy of 56.5 percent.

2.4 Financial time series trend analysis

Notwithstanding the persistence of the effective-market assumption in [42] find that technical assessment is widely used in today's investment production, though the term usually refers to simpler methodologies. One of the most common approaches used as a lagging pointer for stock trend estimating is an exponential moving average, which is based on infinite impulse response. It is the same as moving averages, which is a more widely used word in the broader study of time series, and may be computed recursively as follows, with as the time step indicator beginning at 1 and as the flattening factor with (0,1):

$$\mathrm{EMA0} = \mathrm{x0} \tag{2.1}$$

$$\mathrm{EMAi} = \alpha\mathrm{xi} + (1-\alpha)\mathrm{EMAi} - 1\,, \mathrm{s.t.\,i} > 0 \tag{2.2}$$

Other, less studied approaches employed by technical analysts include perceived patterns, such as the head-and-shoulders example. The latter is employed as a trend indicator, with the negative rate of the height shown in Figure 2.1 serving as the aim price for a market made at the pattern's completion point. Lo et al. [43] present indications for applications in foreign currency markets for several currencies, finding that such patterns may have some useful utility.

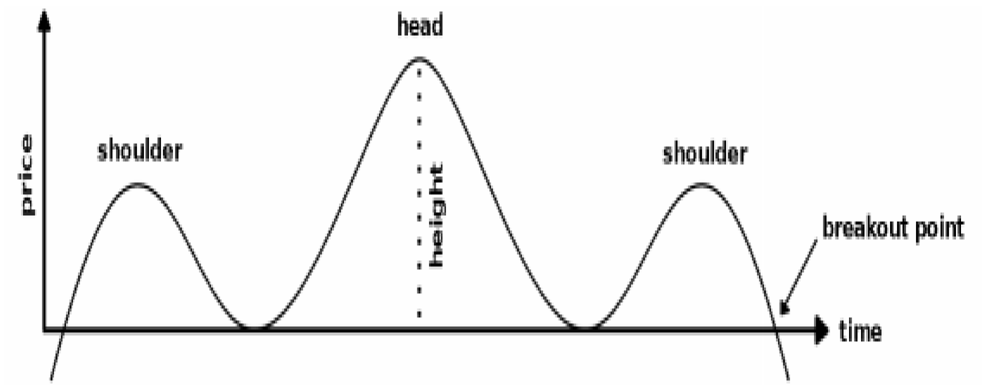

Figure 2. 1 : Pattern of Head and Shoulders in Stock Market Data

2.5 Stock market data is unique.

The core of price fluctuations in the stock market is dependent on the belief about the upcoming execution of a stock, which is itself determined by human views about the upcoming performance of a stock, as outlined in the previous section. In part, investment decisions quantify the future views of other investors, a process that can be repeated over and over again. Due to the influence of the investment decisions and the various forecasting methodologies used, time sequence in stock market places are intrinsically noisy. Since they are all time series, stock markets represent a complex and intriguing example of real-world time series.

2.6 Approaches based on gradients

Mierswa [44] uses a moving window for his audio classification algorithm, and regression models of the frequency spectrum's gradients are input features. The constantly shifting nature of stock markets has presented numerous difficulties to accurately predicting stock prices over the years. Thanks to new and innovative technologies, many avenues have been used to try and solve this problem [49]. However, this work employs time intervals with the derivative of linear regression functions as features, which was demonstrated in past research. Gorecki and Uczak adopt a novel method for time series classification, based on the Keogh and Pazzani work on dynamic time warping, which uses derivatives to assess the similarity of possibly variable-speed time series. Although there is a lack of research in the context of searching for complex correlations in time series with deep-layered ANN, first derivatives for sorting responsibilities in time series are employed for machine learning even when studied within the framework of step-wise linear regression gradients completed with set intervals. The research described in this thesis provides the foundation for future application, as the results here will provide an overall framework for this deep learning approach to time-shifted correlations.

2.7 Using Deep Networks Approach

2.7.1 Artificial Neural Networks: An Overview

A brief overview of artificial network creation is provided in this section to ensure that the reader understands both the process of creating artificial networks that is used in the whole thesis and why neural network models are adequate at hand. These models are described in such a way that readers with no prior knowledge of the subject can grasp later ideas because this thesis is also relevant to economics and finance. A thorough examination of the state of the art is far beyond the scope of this report and is not required to understand the hypotheses and experiments presented here; therefore, the explanations and depictions in this section are restricted to supervised learning with feedforward-type networks only. These models rely heavily on artificial neurons, which were first suggested for combinatorial optimization by McGulloch and Pitts (1943) in the context of thresholds for logical calculations, which is the idea that an artificial neuron must fire rather than remain dormant at a certain strength of activation. The next stage in this progression is represented by perceptrons. Perceptrons are algorithms that perform a linear classification for binary distinctions, similar to how neural networks do it. This type of artificial neural network was among the first types of artificial neural networks to be developed, and it is also the most fundamental example of a feedforward neural network. Rosenblatt was the one who came up with them (1958). The mathematical formulation that occurs for each perceptron is summarized in the following section of the document.

$$f(x) = \{ 1 \text{ i f } w \cdot x + b > 0 \qquad 2.3$$

W and x are R vectors, with x denoting the vector of artificial neuron inputs and w denoting the vector of weights for each input in the vector of artificial neurons. When b is used, it refers to the bias term that reflects the artificial neuron's firing threshold, and when f(x) is used, it refers to the Heaviside step function, which outputs 1 when the argument is positive and zeroes when the argument is negative. The w and x dot product can be written as:

$$w \cdot x = \sum_{i=0}^{N} w_i x_i \qquad 2.4$$

Directed acyclic graphs that use artificial neurons to feed data to the outputs in a single direction are called feedforward neural networks. Typically, these models are linked between layers of artificial neurons in the layer directly next to the one they're in, but they aren't connected across multiple layers. Despite the name's use of the word "single-neuron model," the idea of an artificial neural network is generally stated as either a single- or multi-layer perceptron. In addition, the word "perceptrons" is only used in this thesis in a historical context and artificial neural networks are the preferred term for the models mentioned in the rest of the thesis. In figure 2.2, we see a straightforward artificial neural network with no hidden layers.

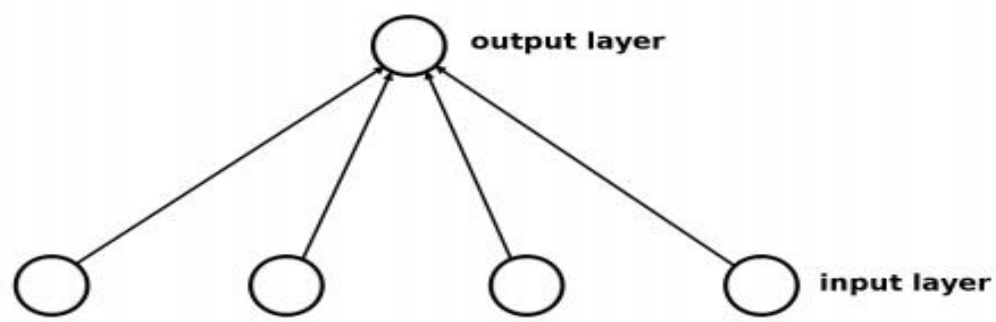

Figure 2. 2 : Neural Network

The input layer reflects the input vector as in formulae (2.3) and (2.4), in this instance values for 04 variables, in this style of depiction. The outcome of passing the input layer's values through the model is represented by the output layer. Because each input is multiplied by weight to get the appropriate output, the ANN is comparable to linear regression in its most basic form. The weights are shown as layers in various depictions, however, this representation will be used all over this thesis to ensure a uniform reading process for all parts. This section also focuses on supervised learning, which involves training a model with data that has already been properly labeled, although other forms of learning, such as unsupervised learning for unlabeled datasets and reinforcement learning, have applications in a variety of fields. Artificial neurons in these models use activation functions, which allow inputs to be modified using weights and, in most cases, a bias component. Non-linear activation functions, such as the Heaviside step function, allow for the solution of non-trivial issues since the outputs are not limited to logical values. Similarly, for non-linear separation tasks, linearly rising activation functions necessitate a high number of artificial neurons, making them computationally costly.

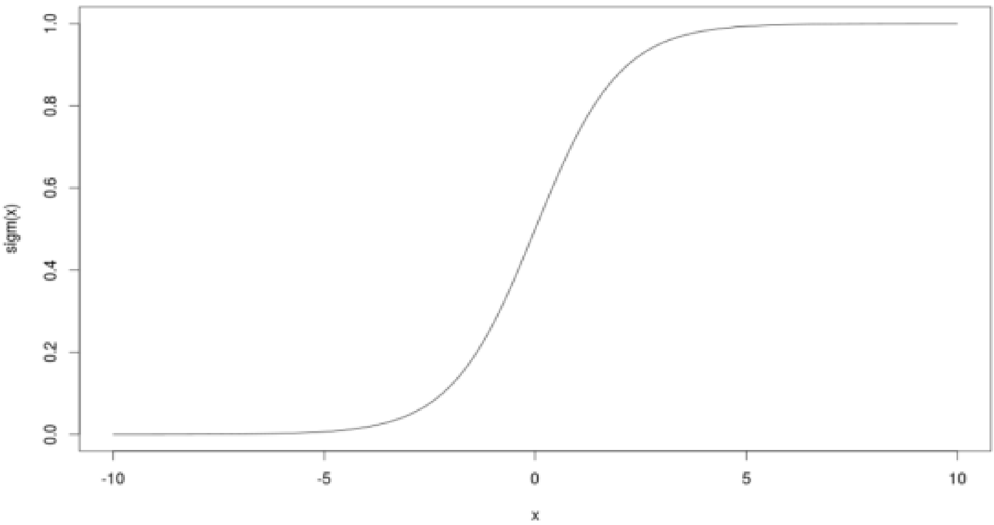

Figure 2. 3 : Sigmoid Function

Instead, the most widely used activation functions are designed to boost output initially, after that progressively asymptotically approach their limit as the value increases. The sigmoid function is a famous example of such a function. Sigmoid functions are a particular example of the logistic function depicted in Figure 2.2. They are used in the training of artificial neural networks. The following formula is used to compute the sigmoid function:

$$\text{sigm(x)j} = 1/(1 + \text{e}^\wedge(-\text{k} \cdot (x_j - x_o)) \qquad 2.5$$

It's worth noting that the sigmoid function's values level off at 0 on the lower end, which can result in weight saturation in the upper layers of multi-layered ANNs [45]. The hyperbolic tangent function, comparable to the sigmoid function but is centered on 0 as a replacement for 0.5 and has the same lower and higher limits for its values as the sigmoid function:

$$\tanh(x) = \left(1 - \frac{e^{-2x\,\mathrm{i}}}{1} + e^{-2x\,\mathrm{j}}\right) \qquad 2.6$$

The softmax function, which is a commonly used technique for understanding the outputs of neural network models used for classification assignments as probabilities, is the final activation function worth mentioning at this point because it is the final activation function worth mentioning at this point. Because it is the most recent, the most recent activation function is worthy of mention at this time (Bishop, 2006). This function differs from the others discussed in this article in that the values between 0 and 1 in the softmax layer add up to one because it uses all of the inputs from the layer before it. Depending on the circumstances, the probabilities of mutually exclusive classes or percentages of a total of 100 percent can be used in the calculations that follow. Additional layers appear between the output and input layers in Figure 2.2, with one hidden layer appearing between the two output and input layers. The basic feedforward artificial neural network depicted in Figure 2.4 can solve a binary classification problem with four inputs and only one hidden layer. It produces two outputs and can also be used to solve other classification problems. Figure 2.4 depicts a simple feedforward artificial neural network:

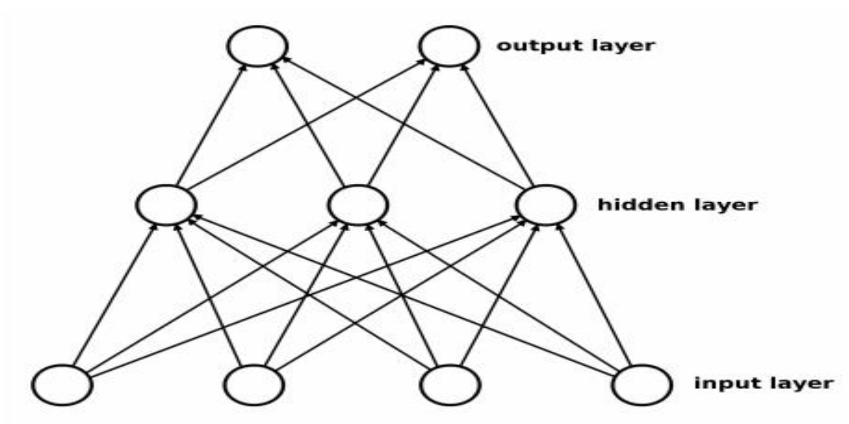

Figure 2. 4 : Neural Network with Hidden Layer

After a lengthy historical time of stagnation in the study of these models, backpropagation of the errors was created as a way to time-effectively train multilayered ANNs in the 1970s (Werbos, 1974). Loss functions are utilized to assign a real-valued number to the overall error underneath a given set of layer weights. By utilizing the quadratic cost function as an example, the total error for this situation may be computed as follows:

$$\frac{1}{2}\sum_{ij}(y^{j},\mathrm{i}\ -y_{j}\ ,\mathrm{i})\ 2 \qquad (2.7)$$

The output units are labeled as j, while the training instances and their associated outputs are represented by I. y and y, respectively, represent the computed outputs and genuine labels. The network's output, after applying the activation function, was achieved by running the inputs through the network, using the last layer's outputs, weighted connections, and bias. After the forward pass, the weights and bias are revised. Backpropagation is a popular way to adjust the parameters of an optimizer, and gradient-based optimizers are frequently used in this regard. Although there is a lack of research in the context of searching for complex correlations in time series with deep-layered ANNs, Even when studied in the context of step-wise linear regression gradients placed above a white specified intervals, first derivatives for classification techniques in time series are used for machine learning to improve classification accuracy. The research described in this thesis provides the foundation for future application, as the results here will provide an overall framework for this deep learning approach to time-shifted correlations. Previous studies have generally employed deep learning frameworks that are dependent on underdone data or limited feature sets, as in [46]–[48]. Our approach has combined both financial technical analysis and deep learning to better predict future stock prices. This is because macroeconomic conditions and technical indicators are maintained in the system like the multivariate signal.

2.7.2 Deep Learning Models

Deep-learning models were often used in the forecasting of energy. In [62] Kong and Dong were using a sequence-to-sequence approach to forecast energy usage in residences and obtained possible better results. As the activities of households vary, residential loads are typically too variable to precisely predict. To deal with such a volatile situation, they employed an LSTM-based forecasting model using appliance usage sequences. It is demonstrated that adding appliance observations in the dataset improves predicting accuracy significantly. Extensive comparative tests on a real-world dataset confirm the efficiency of this approach.

Another author applied stacked AE to remove noise, inconsistency, and instability in the energy consumption data using deep learning models [63]. To make use of these individual characteristics, the author integrates stacked autoencoders (SAEs) with extreme learning machines (ELM). A stacked autoencoder has been used to extract power usage parameters for buildings, while the extreme learning machines were employed as the predictor for reliable forecast results. The auto-correlation evaluation method is used to identify the input parameters of the extreme deep-learning framework. In furthermore, it compares to several traditional approaches of machine learning, such as support vector regression, backward neural propagation network, generalized radial basis function neural network, and multiple linear returns to evaluate the performance of the method. Experimental results show that in various situations of energy consumption in buildings, this technique has the optimum forecast performed.

2.7.3 Hybrid Models

In recent years the researchers concatenate different models for forecasting time series. In [64] they combined CNN with the LSTM for forecasting time series. Layers of CNN extract essential features and these features were used by LSTM to model spatial features. The CNN-LSTM technique delivers reliable outcomes for power usage that was before a challenging task. The model gives the lower value of root mean square error (RMSE) in comparison with other traditional models.

The author in [65] combines CNN and Bidirectional-LSTM. Spatial features were extracted by CNN layers and Bi-LSTM was using these extracted features both in the forward as well as in the backward direction for predicting time series data. They used a deep learning network where modified data sequence through the M-BDLSTM network is sent to the CNN to efficiently understand the sequences trend. The results obtained from this approach were better than the previous ones. Furthermore, it has given the lowest values of mean square error and root mean square error values.A hybrid CNN with LSTM-AE was also used in recent research [66]. The CNN architecture primarily extracts input data features that were then passed to the LSTM encoder to produce an encoded sequence. The encoded sequence was decoded to the last dense layer of power forecasting by another LSTM decoder.

Another author in [67] used a hybrid model to forecast time series data. They have developed a hybrid CNN-LSTM network that can extract spatial features to efficiently forecast power consumption. Results have demonstrated that hybrids CNN-LSTM networks, combining the neural convolution network (CNN), long-range memory (LSTM), and deep neural network (DNN) can extract inconsistent features. The convolutional neural networks layer is used to reduce the spectrum of spatial data, the LSTM layer is appropriate for modeling temporal features, and the DNN layer produces a forecasted time-series dataset. The combination CNN-LSTM method was used to predict future values. Finally, using time series datasets given by the UCI repository, the CNN-LSTM hybrid approach shows a lower root mean square error than other standard prediction algorithms. Another research carried by Tuong Le and Minh Thanh Vo to forecast energy consumption [68]. They used the CNN-Bi-LSTM hybrid model. The first module contains two CNNs that extract essential data from various variables of the Time Series dataset.

2.8 Pattern Recognition

Although pattern recognition and machine learning are synonymous terms, the two techniques are used in different ways when analyzing stocks. Pattern recognition is defined as the detection of patterns and trends in [50]. The OHLC (Open-High-Low-Close) chart is one of the most well-known and is used to indicate a variety of patterns. For years, traders have used the OHLC chart to determine when to buy and sell based on patterns they have observed (Velay and Daniel 2018). Technical analysis examines price, volume, and other metric patterns directly in stock data, which can then be displayed on charts to show price, volume, and other changes [52]. Charting is a technical analysis technique that uses a comparison of historical prices and capacity data to determine patterns and forecast future price movement [53]. According to research, the shapes of price and volume change from a variety of familiar chart patterns [54]. Stock prices exhibit patterns that predict future movement [55]. The first method is the Perceptually Important Points (PIP) approach, which involves dropping time-series proportions (i.e., the number of data points) by conserving the salient facts, while the second is template matching, which involves matching a specified stocks pattern with a symbolic image for object ID. Numerous research projects, according to (Velay and Daniel 2018), have demonstrated that there are correlations between patterns and future developments [56].

2.9 Summary

In this chapter, we discuss existing ANN models with respect to data and methodology used. We find the different types of data and conventional and machine learning approaches are used to predict the underline time series data. We present an overview of the deep network approaches with different type of input data which includes stocks historical price data. In the end we discuss the hybrid models as well as pattern recognition and the results of experiments utilizing various assessment criteria demonstrate the success of the hybrid approaches for the financial time series data.

CHAPTER 3: STOCK INFORMATION AND DATASETS

3.1 Information about the Stock Market

3.1.1 Stocks

The ownership of a corporation's or organization's equity is determined by the number of shares of stock held by its shareholders. In a corporation, a single share of stock represents a proportionate percentage of ownership concerning the total quantity of shares in the corporation. When a corporation is formed, its stocks are identified and the total number of shares in the business is determined. Shareholders who already own stock in the company can vote on whether or not the company should issue additional shares. The declared par value of each share, which is the authorized accounting price and which represents the impartiality on the organization's total sheet, is known as the par value in some cases. Additionally, there are instances in which stocks are issued with no monetary value attached to them.

3.1.2 Shares of a Stock

The shares of a corporation represent a percentage of the corporation's tenure. In a corporation's share capital, it can issue many different types of shares, each with its own set of ownership rules, privileges, and shared values, all of which are distinct from the other types. A documented stock certificate is provided to each shareholder as evidence of their ownership interest in the company. Shareholders can obtain a stock certificate, which documents the number of shares held by them and other information about those shares, and the class into which the shares were issued.

3.1.3 Stock Market

A stock market is a gathering of sellers and buyers of stocks, which reflect ownership rights in businesses and are traded on a stock exchange. The vast majority of these are securities traded on a public stock exchange, but they can also include securities traded privately. Private trades in the financial markets include the sale of shares in private companies to investors via equity crowdfunding platforms, which are examples of private trades. Investors can trade not only common equity shares on stock exchanges but also other types of securities including corporate debt and convertible bonds, among others.

3.1.4 Stock market prediction

Stock market prediction refers to the act of attempting to predict the future value of a particular value or another financial instrument that is traded on a trading platform. In the last 20 years, the field of neural networks has progressed a lot, with significant progress being made in a variety of fields. Traditional artificial intelligence systems (AI) have been the most distinguishable areas of machine learning (such as classification and data analysis) for many years (artificial intelligence). A wide range of APIs for neural network applications has been designed for a variety of computer languages, with Google's Tensorflow serving as an example. When combined with the rapid advancement of computing power, neural networks have found widespread application in our daily lives. While using Google Translate, a substantial percentage of mobile phone applications (e.g., applications with face detection), or trying to read about self-driving automobiles, it is easy to see how neural nets are being used to improve accuracy.

3.2 Shareholders

The term "shareholder" refers to an individual or a legal entity who has the legal right to a specific amount of shares in a publicly-traded joint-stock firm. Shareholders are entitled to a variety of benefits, which vary depending on the type of stock in which they invest and the amount of money they invest. For example, shareholders have the option to vote on matters, and they can receive a share of the company, the right to purchase new shares delivered by the corporation, and even the right to access the corporation's assets if the corporation is liquidated. A company's holdings are only as valuable as its liabilities, and the rights of shareholders to those assets are hampered by the rights of creditors of the corporation.

3.3 Moving averages

The use of moving averages, as recommended by Gerald Appel in his book "Technical Analysis: Power Tools for Active Investors," is used to deal with the noise of shorter-term price fluctuations and to identify significant trends with greater precision, as opposed to the use of simple moving averages. Moving averages can be divided into different types:

3.3.1 Simple moving average (SMA)

If you use this form of moving average, every piece of data is treated in the same way. Take for example the assumption that x = (x1,....xn) represents a subcategory of our data. Following that, the simple moving average of such a subset is determined, and the results are as follows:

$$SMA = 1/n \sum_{i=1}^{n} x_i \qquad (3.1)$$

3.3.2 Weighted moving averages (WMA)

When using a moving average of this type, multiple parameters are assigned to different pieces of data, resulting in a more accurate representation of the data. Consider the following example:

some weighted averages give greater weights to more recent data points while penalizing elderly data points by offering them lower weights, as illustrated in the chart below. Remember that x = (x1,....xn) represents the subcategory of our data, and w = (w1,...., wn) represents the weights in our model. Following that, the weighted average of that small subsection is calculated in the following way:

$$WMA = 1/n \sum_{i=1}^{n} x_i \, w_i \qquad (3.2)$$

where n denotes the size of the window that was used in the calculation of the average You can create a diverse array of weighted averages using different methods, with the major difference being the method by which they are calculated.

3.4 Time-based separation of Moving average

Additionally, moving averages can be divided into three classes based on the type of market movements we are interested in observing.

3.4.1 Long-Term Moving average

It is used to reflect longer-term trends in the market and is one of the most common types of moving average. Averaging over a long interval of time is typically used to determine the behavior of stock and to reduce the likelihood of a market turnaround shortly, for example, the 200 days moving average, which is a useful indicator.

3.4.2 Intermediate-Term Moving average

In the medium-to-long term, this form of moving average indicates the trends in the market and is used in technical analysis. The stock market is typically more favorable during periods of strong upward stock market cycles, and they serve as entry points into the market. When intermediate-term moving averages are declining, it is possible to find selling opportunities. For example, 50 days moving average can be used to illustrate the concept.

3.4.3 Short-Term Moving average

It is used to reflect changes in the market that occur over a short period by using a moving average of this type. In most cases, they are used in conjunction with an extended average to provide more context. When a short-term moving average intersects over a long-term moving average, it indicates a change in momentum, indicating that it is time to take decisive action. An example of a short-term moving average, such as the 10 days moving average, can be used to demonstrate the concept. While these indicators are commonly used in day trading, they can also be used to model intraday trading on a variety of timeframes, including hourly, 30 minutes, 15 minutes, and even 5 minutes timeframes. They can also be used to model intraday trading daily, which is a useful tool for traders. In addition, the short-term moving average is utilized in this study.

3.5 Types of Learning Algorithms

3.5.1 Supervised Learning

In this form, the preferred response in which the neural network would have provided if it had been trained to do so. It is then necessary to make adjustments to the network's parameters, which are influenced by the training vector as well as the error signal, and the process must be repeated. According to the error signal, both the actual network response and the desired output at its maximum value are taken into consideration. It is necessary to adjust the network parameters one at a time, and the process is repetitive until the network's actual output is close enough to the desired output to be considered acceptable. Classification and regression are the two tasks for which supervised learning is most commonly used [59]. Classification is the most common task, followed by regression.

3.5.2 Classification:

In this task, we want to divide our data points into a specific number of classes based on their characteristics, and we want to do so as quickly as possible. The following classification example can be used when we are unable to distinguish between data points that are above or below a curve.

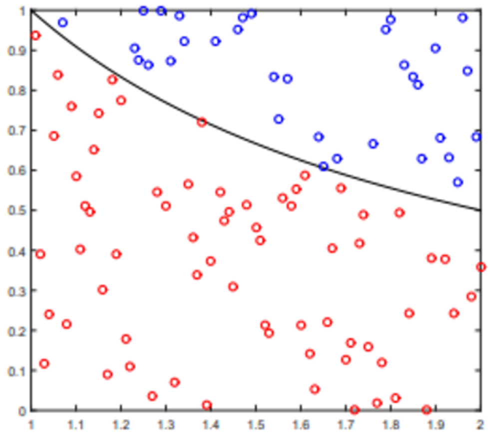

Figure 3. 1: Classification Problem [58]

In the illustration above, the function 1x is used as a separator between two lines of code. Our school is distributed into two clusters of students: blue and red. In this case, points that are above the separator should be grouped into the blue one, and spots that are below the partition should be grouped into the red class. In this problem, the primary goal is to estimate an unidentified filter function using only a small number of statements, and the secondary goal is to attempt to reproduce the same behavior in the common case [58].

3.5.3 Regression:

It is a basic concept in the area of machine learning that everyone should be familiar with. supervised learning, in which the algorithm is instructed using both input characteristics and output label information, is the category in which this algorithm falls. A relationship between two variables can be established with the help of this technique, which estimates the extent to which one variable influences the other.

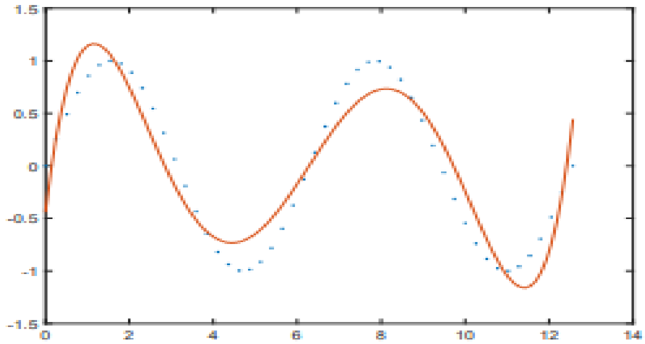

Figure 3. 2: Regression Problem Representation [59]

The relationship estimation task [59] is the task in which we are attempting to estimate relationships between our data points. In the case of function approximation, we are attempting to understand a function that can explain our data points as accurately as we can. A polynomial function is exemplified by the red line in Figure 3.2 above, which attempts to approximate points from a function called the sine (x).

3.5.4 Unsupervised Learning

In this situation, it is impossible to provide the network with the preferred reaction to the training vector, so the network must be shut down. The parameters of the network are optimized for a task-independent assessment of the interpretation of what the network is expected to find out from the data. Because of the network's optimization, it may eventually learn how to encode features of the input and become capable of creating new classes on its own without any human intervention. Unsupervised learning is commonly used in the development of auto-encoders and auto-decoders [57], where the goal is for our neural network to obtain the essential features from the data while discarding the less valuable features, and to produce a useful data interpretation in a lesser dimension.

3.6 S&P 500 Index Dataset

The Standard & Poor's 500 index of huge capitalization U.S. stocks is widely regarded as the best single indicator of the overall market's performance. Overall, there is more than USD 7.8 trillion worth of assets that have been benchmarked to the index, with the index properties accounting for around 2.2 trillion of the whole. A total of 500 leading firms are integrated in the index, which represents approximately 80% of the total marketplace capitalization that is available. The S&P 500 index was established for the first time on March 4, 1957. Its constituents are primarily based in the United States, as the name implies. Generally speaking, the S&P 500 index represents the 500 largest corporations in the United States of America, which is a widely held belief. Rather than being chosen at random, the S&P 500 is selected by a select committee that takes into account factors such as "market size, liquidity, and group representation," among other things. No matter that the vast majority of S&P 500 constituents are large capitalization companies, the required minimum capitalization to be included in the index is $3 billion. A further point to mention is that there are at least 20 firms in the S&P 500 that are classified as small and mid-capitalization in their nature. Several major industry categories from which the constituents are derived. Furthermore, within each of these industries, there are over 100 subgroups that determine the criteria for inclusion and exclusion from participation.

Additionally, numerous other arguments can be used to support the inclusion of the United States stock market in this research. It goes without saying that the United States is one of the world's most populous, wealthiest, and most influential countries. The country's social, political, and economic conditions have the potential to have far-reaching consequences on a global scale. The United States financial markets, more specifically, and in relation to my research topic, have long been recognized as the world's largest and most highly developed, and they are also widely regarded as industry leaders in terms of modernization and the use of emerging technologies.

3.7 Why the S&P 500 Composite Index for the U.S market

Following previous information, the index was established in its existing form on March 4, 1957, and it currently contains 500 leading firms that account for approximately 80% of the total market capitalization available. Each company's market capitalization weights are used to calculate the index price, which is then adjusted to account for the float adjustment. It is possible to calculate the index price in real time for a number of different currencies. Today, the Standard & Poor's 500 Composite Index is commonly considered as the most accurate single indicator of the health of large cap U.S equities as well as the overall health of the United States economy. Over 7.8 trillion USD in assets are benchmarked to the index in the arrangement of a diverse range of third-party products, with index holdings accounting for approximately 2.2 trillion USD of the whole. In light of these facts, as well as the fact that data is freely available and transparent, it is an excellent choice for my investigation in the United States. In the following section, I will go over the specifics and summary statistics of the datasets that were used in order to give a clear picture of their nature and serve as a motivation for the research that will be conducted in the following section.

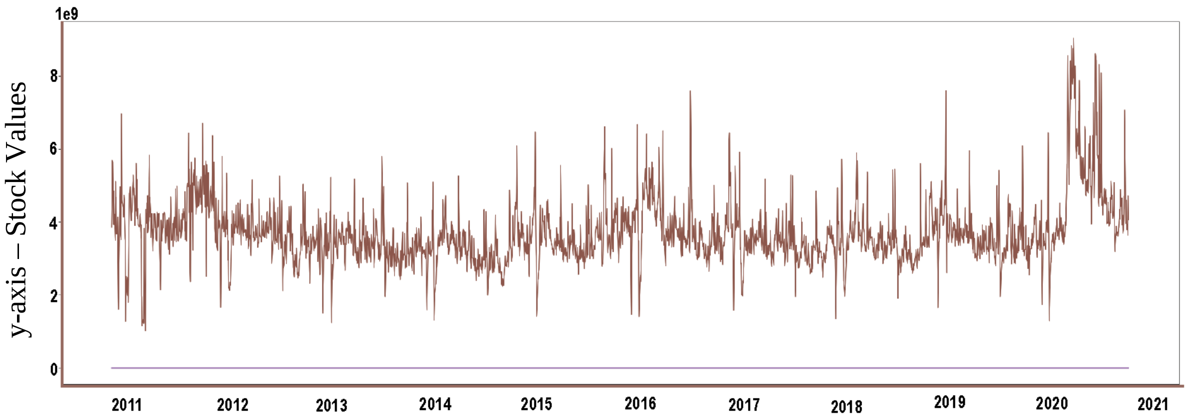

Figure 3. 3 Complete Duration plotting of S&P 500

3.8 S&P 500 Dataset Description

We first make use of the historical of Standard and Poor 500 index ranging from 11-02-2010 to 10-02-2020. The price data obtained from Yahoo Finance contains the daily closing prices of each stock for 10 years for each day the respective stock was traded during this period. This data set contains nearly a decade's worth of information and is used to test and evaluate the effectiveness of our first proposed trading algorithm, which aims to forecast future profits solely based on price history. To get ready for the implementation of an ANN, divide the data into two groups: training and testing. The forecasting model is created with the assistance of the training set. It is used to compute the gradient and to update the ANN's weights and biases. The validation set measures how well the model interpolates over the training set. 80 percent of the data has been utilized for training the model and the remaining 20 percent of data has been used for the training of the model. The dataset consists of open, close, low, high, adj close, and volume information of the Standard and Poor 500 index.

3.8.1 Open and Close Prices

During the trading day, the "open" refers to the price where stock began trading. Not always, but occasionally, the stock is trading in the same location as when it closed the previous night. During after-hours trading, certain events such as company earnings reports can sometimes have a substantial impact on the price of the stock the following day. It is referred to as the close price if the price of a particular stock was at its highest point at the time the stock trade closed. Currently, it symbolizes the most recent buy and sells order that was executed amongst two traders at the time this article was written. These occurrences are common in the final seconds of a trading session when the market is at its most volatile. Although a great deal depends on when the last buy and sell orders were matched, in less consciously shares traded, the last trade of the day may occur well before the end of trading [61].

3.8.2 High and Low prices

Regularly, financial periodicals and financial websites publish the current and historical high and low prices of a company's stock. The peak price to which a stock has traded throughout a given period is referred to as the superior price of the full-day stock. The lower price is the price that was offered at the lowest point in time during the specified period. If a stock has a high and low point during the day, the high and low points of the stock are often referred to as the stock's daily maximum and low points.

The 52 weeks high and low of stock are often displayed on the same page, which is another common occurrence. This represents the highest and lowest daily closing prices for the stock over the course of a year. Using it, we can get a general idea of the stock's trading range on an annualized basis. In a one-year price chart, we can easily identify these points because they are the peak and low points of the mountain and valley, respectively [61].

3.8.3 Adjusting values and volume

In order to account for changes in stock price as a result of corporate actions such as dividend payouts, stock splits, and the issuance of additional shares of stock, among other things, it is necessary to adjust the value of stock on a regular basis. It is defined as the number of shares of a company's stock that have changed hands during a specified period of time in order to purchase or sell that company's shares [61].

3.9 Why is stock’s closing price significant?

Despite the fact that the stock market opening prices garner a great deal of attention, it is the closing price of a stock that decides how well a stock performed throughout the whole day.

It is common practice to use the close price as a reference point when discussing any time frame. Following a flurry of activity throughout the day, this is the price that traders came to an agreement on. Whenever financial institutions, regulators, and individual investors conduct historical stock price data research, they use the closing estimate of the stock as the standard extent of the stock's significance as of a particular date in their research. Take, for example, the closing price of a stock on December 31, 2019, which was the ending price not merely for that day, but the following week, month, and following year.

The percentage return on a stock is calculated by dividing the difference between its close and open prices by the price at which the stock was first traded (i.e., when it was first traded). We would use the stock closed from a year ago and start comparing it to the current share price from today to figure out how much an investment is worth over a longer period of time, such as one year, in order to figure out how much an investment is worth over a longer period of time.

3.10 Data Preprocessing

Immediately following data transformation, data cleansing is carried out in order to ensure that the information is accurate. In the context of data cleansing, there are two primary tasks to be completed. Two different issues are addressed: one is dealing with missing data, and the other is dealing with data alignment. The first thing to consider is how to deal with data that isn't available. A variety of assets and areas were used in the creation of this paper, which came from a variety of different sources. Because different time zones have different holiday arrangements and different trading day arrangements, some data is missing in different time zones due to the differences in holiday and trading-day arrangements. Due to the extensive use of historical data in this paper, it is reasonable to exclude data that was not available at certain points in time during the research process. Similar to the above, the Dataset contained various attributes such as open, close, low, and high values, adj close values, and volume values to name a few. The reasons for this are discussed in greater depth in the sections that came before it. The total amount of shares traded in protection over the course of a given period is referred to as the security's volume in the marketplace. It is added to the total amount of money exchanged during the period for each time buyers and sellers exchange shares. So, the values for the volume attribute are very large as compared to the other attributes which affected the training as seen in figure 3.3. As we already discussed the significance of closing price attribute and why we are going to predict the closing price for next day so for the prediction of next day closing price, removing of volume attribute has not any effect on the model performance that's why for better training volume attribute has also been dropped during preprocessing. The plotting of data after preprocessing step can be found in Figure 3.4.

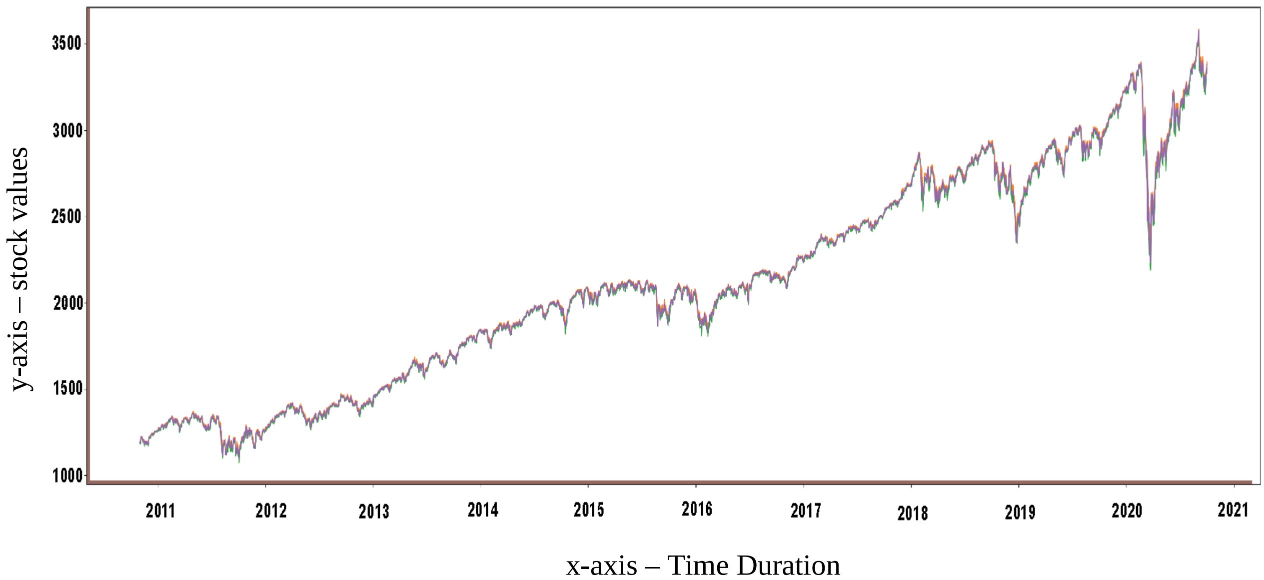

Figure 3. 4: Dataset plotting after preprocessing

3.11 State Bank of Pakistan Dataset

Moreover, the State Bank of Pakistan dataset has been used in this study, which was obtained through the proper channels and has been made available to researchers. The State Bank of Pakistan (SBP), which was established in 1956 and has the ability to serve as the country's dominant bank, as stipulated by the State Bank of Pakistan Act of 1956, was established. Among other things, the State Bank of Pakistan Act mandates the Bank to legalize Pakistan's budgetary and credit system and to stimulate their development in the finest nationwide interest, to maintain fiscal stability, and maximize utilization of the country's industrious resources. To achieve the country's macroeconomic policy objectives, SBP as the country's fundamental Bank has charged with the accountability of formulating and implementing fiscal and credit strategy in a way that is reliable with the government's growing and inflation targets, as well as recommendations from the Monetary and Fiscal Policies Coordination Board.

3.12 EXCHANGE RATE MANAGEMENT

In addition to a number of other important responsibilities, the State Bank is also responsible for the preservation of the currency's external value. For example, the Bank is required to regulate the country's foreign exchange reserves in accordance with international standards, among other responsibilities. Consequently, the Bank is responsible for maintaining a reasonable level of the rupee's exchange rate and preventing it from experiencing significant instabilities in order to sustain the effectiveness of our transfers and the steadiness of the international financial markets. The implementation of various exchange policies from time to time, while taking into consideration the current economic environment, has been undertaken in order to achieve this goal.

3.13 Dataset Description and preprocessing

In this research the dataset of SBP has also been used for the stock market estimate, ultimately the forecast of the exchange rate of the next day. This dataset consists of different exchange rates such as AUDUSD, EURUSD, GPBUSD, NZDUSD, CADUSD, CHFUSD, JPYUSD, AAAUSD, and BBBUSD. The dataset collected depends upon six months dataset from October 01, 2019, to March 31, 2020. This was minutely dataset that has been further preprocessed to hourly dataset and daily dataset as well. The daily dataset was finally used for the research purpose and next day exchange rate has been predicted for the AUD/USD exchange rate. AUD/USD Currency trading takes place between the Australian dollar and the US dollar, and this is referred to as the AUD/USD currency pair in the world of currency trading. Trading the AUD/USD rate, which can be seen in the real-time price chart, allows traders to determine how many US Dollars are required to purchase one Australian Dollar. On the interactive chart, the AUD/USD currency pair is being tracked in real time as it changes. You can also stay up to date with the most recent forecasts and AUD/USD news, which will aid you in improving your technical and fundamental analysis when trading this currency pair.

3.14 Summary

In this chapter we discuss the basics of the stock market and differentiate the moving averages specifically. We present an overview of the different type of learning algorithms. We provide a high-level summary of all the datasets that were used in the research for this dissertation. There are two datasets used in this thesis to examine various methods of uncovering inefficiencies in the financial markets, and they are both used in this thesis. The collection and significance of the datasets has been discussed and, in the end, we present the description and the preprocessing of the datasets.

CHAPTER 4: METHODOLOGY

4.1 Developing LSTMs for Time Series Forecasting

Time series prognostication can be accomplished through the use of Long Short-Term Memory networks, also recognized as LSTMs. There are many different types of LSTM models available, each of which can be used to solve a certain type of time series projecting problem, and each of which has its own advantages and disadvantages. We will learn to develop a collection of LSTM models for a variety of normal time series foretelling problems in the proposed work, which will include Serving as a template for future time series forecasting problems, this study will provide impartial examples of the individual model on all types of time series problem. This pattern can then be copied and adapted for any individual time series estimating difficulty that arises in the future.

4.2 LSTM Models

When modeling univariate time series forecasting problems, such as those involving weather forecasting, linear support vector machines (LSTMs) can be used to help. In order to solve these problems, a specific series of explanations is used, and a model is used to be taught from the sequence of previous observations in order to foresee the following value in the series. For the prediction of univariate time series, this paper will present several different variations of the LSTM model, each of which will be discussed in detail. Listed below are the divisions of this section, which are organized as follows:

- **Data Preparation**
- **Recurrent Neural Network**
- **Vanilla LSTM**
- **ConvLSTM**
- **Encoder-Decoder Model**
- **Encoder-Decoder ConvLSTM**

All of these models are determined for one step multivariate time series forecasting; however, by modifying the input parameters, they can be easily adapted and used as the input part of a model for other types of time series forecasting problems.

4.2.1 Data Organization

For the best results from modeling, it is necessary to prepare a multivariate series before beginning the modeling process. A function that outlines a sequence of previous values used as input to an output observation using data from the previous observations and information learned from the previous observations in the sequence is learned by the LSTM model through use of previous observations in the sequence, with the help of previous observations in the sequence. This necessitates the transformation of the sequence of observations into a collection of examples from which the LSTM can draw its conclusions, which is referred to as the transformation step. If you want to learn how to make a one-step prediction, you can break the sequence down into multiple input/output patterns, which are collectively referred to as samples. It is necessary to use three time steps as input for the one-step prediction that is being learned in this manner, and one time step as output for the three-step prediction. You can achieve this behavior on your computer by making use of the split-sequence() function that is available. This function will split a given univariate sequence into multiple samples, each of which has a specified number of time steps, and the output will be a single time step; the output will be a single time step. This function will split a given univariate sequence into multiple samples, each of which has a specified number of time steps. When given a single univariate sequence, this function will divide it into multiple samples, each of which will have a specific number of time steps.

Following our discussion of how to prepare a multivariate series for modelling, let us take a look at the development of LSTM models that can learn the mapping between inputs and outputs, beginning with a Vanilla LSTM and progressing to more complex models as time permits.

4.2.2 Recurrent Neural Network

A recurrent Neural Network is an internally memorized induction of a feedforward neural network. RNN is frequent in character, as it conducts a very similar role for each data input while the output of an existing input relies on the previous one. It is replicated and then directed again into the recurrent network after the output has been generated. This evaluates the present input and the feedback it has accumulated from the preceding input for building a choice.

Except for feedforward neural networks, RNNs are allowed to be using their internal state (memory) to interpret input series. It renders them relevant to activities such as unsegmented processing, recognizing joined writing, or voice identification. All inputs are distinct from another one in different neural networks. In RNN, however, all inputs are connected to one another.

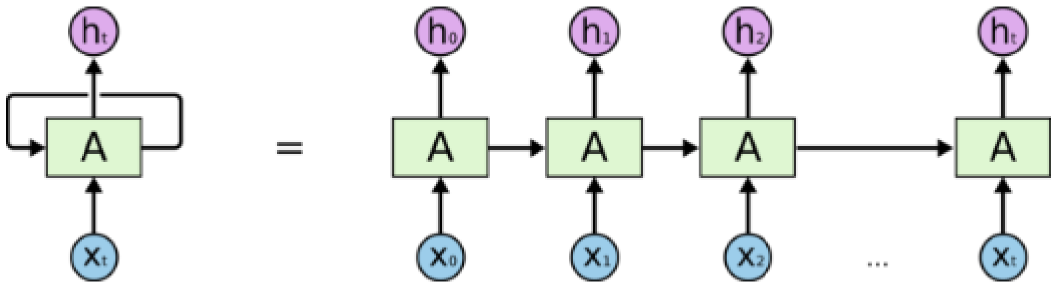

Figure 4. 1: An Unrolled form of Recurrent Neural Network [71].

Initially, it ends up taking the X(0) from the input sequence and then outputs h(0) which would be the input for the next step along with X(1). Therefore, the input towards the next phase is h(0) and X(1). Likewise, the input with X(2) for the next stage is h(1) from the next stage and so forth. In this way, while training it keeps reminding of the context.

$$h_t = f(h_{t-1}, x_t) \tag{4.1}$$

Applying the activation function

$$h_t = tanh(W_{hh}h_{t-1} + W_{xh}x_t) \tag{4.2}$$

Where W is said to be the weight, h is said to be the single hidden vector, W_h is the weight at the former hidden state, W_{xh} is the weight at the current input state, tanh is the Activation function, which imposes a non-linearity which squashes the activations to the range [-1.1]. The output will be

$$y_t = W_{hy} * h_t \tag{4.3}$$

The output state is the Y_t and weight at the output state is denoted by W_{hy}.

4.2.2.1 Advantages of Recurrent Neural Network

i. RNN should model data sequence such that it could be concluded that every observation depends on preceding ones.
ii. Recurrent neural networks are also utilized to expand the efficient cluster of pixels with coevolutionary layers.

4.2.2.2 Disadvantages of Recurrent Neural Network

i. Gradient problems are bursting and disappearing.

ii. It is a tough process to prepare an RNN

iii. If tanh or ReLu are used as an activation function, this can not proceed with extensive sequences.

RNN Encoder-Decoder is a new neural network architecture that encodes a variable-length series in a fixed-length vector representation again into a variable-length series. From a deterministic viewpoint, this latest model is a resource for training conditional distribution over an analyzed variable-length conditioned on another series of variable-lengths, e.g. $p(y_1, \ldots, y_{t'} | x_1, \ldots, x_t)$ where one should observe that the input and output sequence lengths t and t' may differ. The encoder is an RNN that consecutively extracts every sign of an input sequence x. The hidden state of the RNN varies as it recognizes each symbol as per Eq. (6).

The presented model encoding is another RNN equipped to produce the output sequence by a nticipating the next symbol y_t provided the hidden state $h_{(t)}$. Though, unlike the description of RNN, both y_t and h_t are also conditioned on y_{t-1} and the summary c of the input sequence. The two components of the proposed RNN Encoder-Decoder are jointly trained to maximize the conditional log-likelihood

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log p_{\theta}(y_n | x_n) \qquad (4.4)$$

To understand this we have to keep in mind the previously described Equation 1 where θ is the set of the model variables, and each (x_n, y_n) is an (input sequence, output sequence) set from the training set. In this situation, we could use a gradient-based technique to calculate the model values as the output of the decoder, beginning from the input, is distinguishable. The design could be used in forms, once the RNN Encoder-Decoder is equipped. One approach is to use the design, representation of the input sequence to produce a target sequence. In another end, the design could be used to grade a specified couple of sequences of input and output, where the score is merely a likelihood $p_{\theta}(y_n | x_n)$ from Equation (9).

4.2.3 LSTM

LSTM networks are an expansion of existing neural networks (RNNs) that are primarily used to deal with conditions in which RNNs are failed [69]. Speaking about RNN, it is a system that operates on the current input by getting into account the prior output (response) and preserving for a limited period (immediate memory) in its memory. The most famous of its numerous technologies are in the spoken word processing, non-Markovian control, and music production fields. However, RNNs do have disadvantages. Firstly, data is not processed for a lengthy period. A link to various data that was deposited a very long time ago is sometimes necessary to estimate the contemporary output. RNNs are, however, absolutely unable to handle such "long-term dependencies."

Furthermore, there is no more significant influence on what part of the meaning will be taken forth and how much of the history will be 'forgotten.' Other problems regarding RNNs are the bursting and vanishing gradients (explained later) that appear throughout a network's learning phase by backtracking. So Long Short-Term Memory (LSTM) was put in the frame. This was constructed in such a way that the issue of the vanishing gradient is almost eliminated, although the training system remains unchanged. Longer-term trails in some problems are reconciled using LSTMs wherein sound, divided presentations, and continuous values are also handled. For LSTMs, there is no need to maintain as needed in the hidden Markov model (HMM) a finite number of states from beforehand. LSTMs give us a wide variety of measures, including training speeds, and input and output unfairness. Therefore, no need for finer customizations. The difficulty of updating every weight with LSTMs is reduced to O(1), close to that of Back Propagation Through Time (BPTT), which is a benefit.

The actual LSTM algorithm used a custom-built estimated approximation of gradients that permitted the weights to be modified after each phase move [70]. Instead, the maximum gradient can be measured over time with backpropagation. One challenge with complete gradient training for LSTM is that the components often become too high, directing to mathematical issues. To avoid this, all the analyses in the research cropped the network input losses derivative to the LSTM layers (before the sigmoid and tanh functions are put) within a preset scale.

LSTM networks are a customized variant of RNNs that efficiently recall long-term dependencies in input. The problem of vanishing gradient is encountered by RNNs, but it is overcome by LSTM networks. The memory unit (or cell) at the heart of an LSTM network is seen in Fig. 4.1. A cell is made up of one tanh and three sigmoid layers that create three gates that organize data within and outside the cell. The memory unit's input and output signals are controlled by the input and output gates, correspondingly. With a sigmoid function, the memory unit can be restored by forget gate.

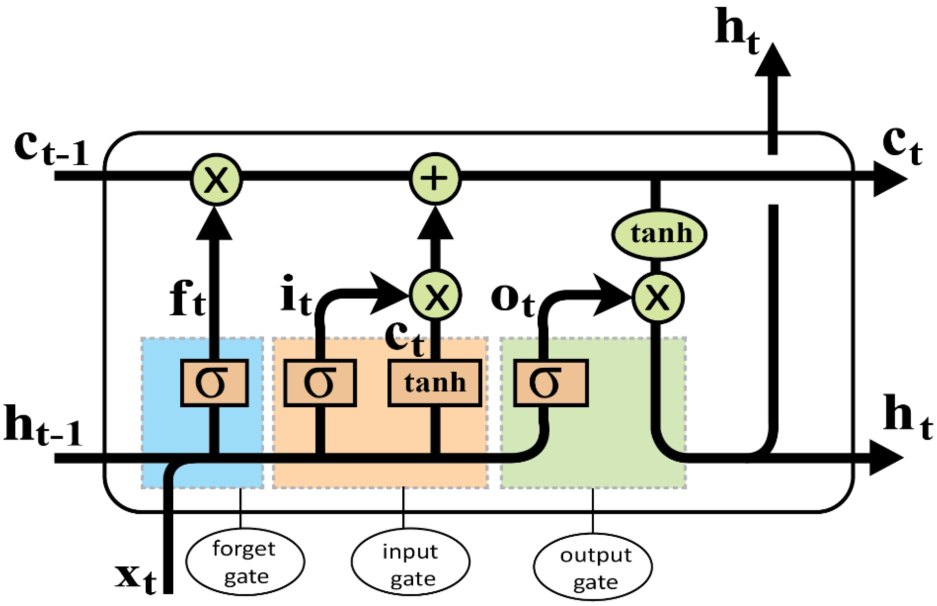

Figure 4. 2: LSTM Structure [70]

Given the information xt, the information flow inside an LSTM cell may be expressed as follows:

$$\sigma\ [bf + (ht\text{-}1, xt)\ Wf] = Ft \tag{4.5}$$

$$\sigma\ [bi + (ht\text{-}1, xt)\ Wi] = it \tag{4.6}$$

$$[bc + (ht\text{-}1, xt)\ Wc]\ \tanh = \tilde{C}t \tag{4.7}$$

$$ft * Ct\text{-}1 + \tilde{C}t * it = Ct \tag{4.8}$$

$$[bo + (ht\text{-}1, xt)Wo]\ \sigma = Ot \tag{4.9}$$

$$(Ct)\ \tanh * Ot = Ht \tag{4.10}$$

where ft is forgotten and it is an input g, and ot is output at time t. A cell state vector denoted by Ct, updated in (5), and ht indicates the current hidden state and hyperbolic tangent function is tanh at time t, (5) and (7) represents a point-wise multiplication operator.

The fact that LSTMs are capable of supporting sequences in their native form is a significant advantage. The internal state representation of an LSTM model, in contrast to that of the CNN, is built up by reading through each individual time step of the sequence and can be used as a learned context for making a prediction in the next time step. The internal state representation of a CNN is not built up by reading through each individual time step of the sequence.

4.2.4 Exploding and Vanishing Gradient

The primary objective during a network's learning phase is to reduce the effects (in regards to deviation or expenses) found in the performance when the training samples have been sent through it. The gradient is measured, such that, losses with regard to the selected set of weights, change the weights appropriately and continue this procedure until an ideal group of weights is obtained whose loss is minimal. This is the Backtracking principle. Occasionally, the gradient is relatively small. It should be remembered that the gradient of a layer in the successive layers relies on certain elements. If any of those elements are smaller (less than 1), the data generated will be even smaller, that is the gradient. It is referred to as the scaling effect. When this gradient is multiplied by the learning rate, which is a small value ranging from 0.1-0.001 in itself, it results in a smaller value. As a result, the weight changes are relatively minimal, generating about the same production as before. Likewise, the weights are modified to a value above the optimum value when the gradients are very high in value due to the large values of the components. These are recognized as gradient explosion problems [69]. The neural network unit was re-built to avoid this scaling effect in such a manner that the scaling factor was attached to just one. Several gating units then enriched the cell, and it was called LSTM.

It is tough for a simple RNN to develop long-term stability when the problem of vanishing gradients and exploding gradients is very common. To overcome this problem, a specific type of recurrent network named LSTM was introduced [69] and successfully extended for translation tasks and sequence generation [70]

4.2.5 ConvLSTM

Conv-LSTM is a recurrent neural network model for the spatiotemporal forecast with convolutional layers in both state transitions. It predicts the future state of a specific grid cell based on the inputs and historical states of its nearby neighbors. This is readily accomplished by employing a convolution operator in state transitions. Like the LSTM, it is a Recurrent layer, but inner matrix multiplications are replaced with convolution procedures. As a result, rather than being a 1D vector containing features, the data that flows through the Conv-LSTM cells retains the input dimension.

In Conv-LSTM the input is passed through the convolutions layers and the output is a set flattened to a 1D array containing the acquired features. This procedure is repeated for all inputs in the time frame, the outcome is a set of features across time, which is the LSTM layer input.

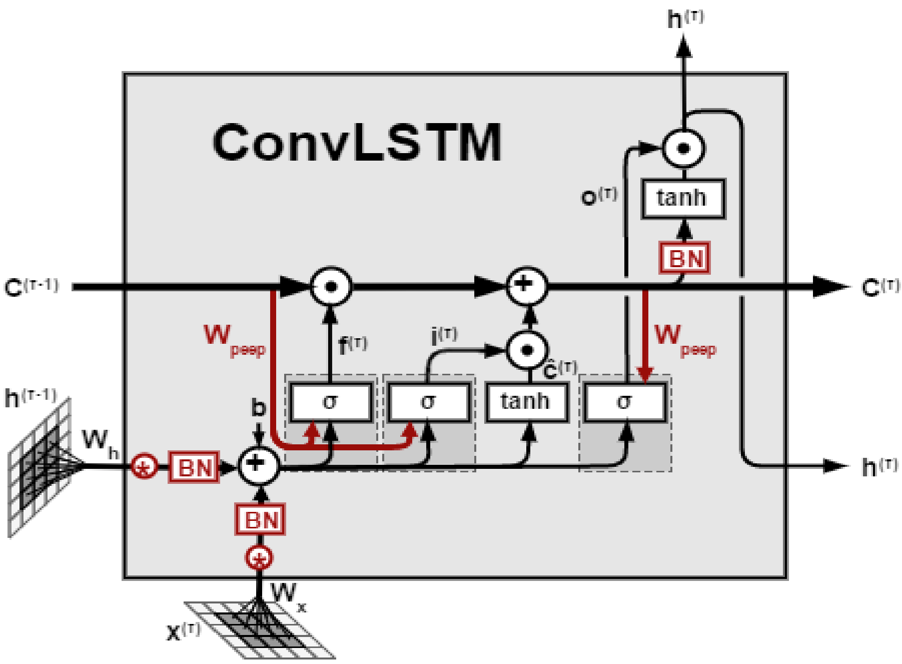

Figure 4. 3: ConvLSTM Structure [70]

4.2.6 Encoder-Decoder Model

When it comes to forecasting unpredictable length output production, the Encoder-Decoder LSTM is a model that has been specifically developed for that purpose. It is referred to as such. In order to solve sequence-to-sequence problems (also known as seq2seq problems), the model was developed for problems involving both input and output sequences. Examples of such problems include translating text from one language to another or predicting the outcome of a video game. With the help of this model, it is possible to forecast time series data in multiple steps at the same time.

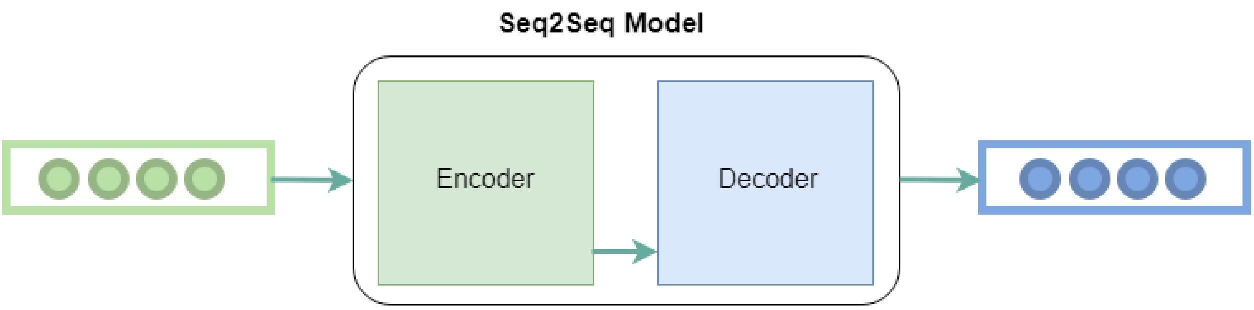

Figure 4. 4: Sequence to Sequence Encoder Decoder [71]

The model is subdivided into two sub-models, which are referred to as the encoder and the decoder, respectively, in order to simplify the description. The encoder is a model that is in charge of reading and understanding the sequence of data that has been received as input from the encoder. Following the operation of the encoder, a fixed size vector corresponding to the model's analysis of the sequence is produced as a result. Other encoder models, such as stacking, bidirectionality, and CNNs, can be used in addition to the traditional Vanilla LSTM model, as can other decoding techniques [71].

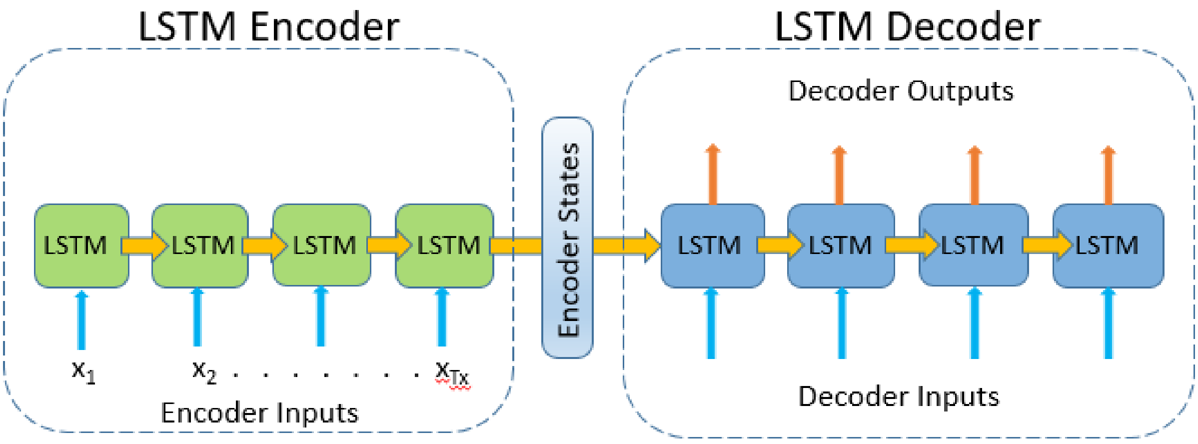

Figure 4. 5: LSTM Encoder Decoder [71]

As a result, the encoder's output can be used as an input by the decoder in both directions, and the reverse is also true. As part of the proper initialization and testing of the encoder, it is necessary to repeat the encoder's fixed-length output a number of times, once for separately time step in the output order that is necessary. After that, an LSTM decoder model is fed into the sequence, which decodes the data and feeds it back into the sequence. When an output model generates values for respectively values in the output time step, and to be decoded and understood by a single output model, the model must conform to the following requirements: LSTM models require that the data be reshaped into a three dimensions shape of [samples, the time steps, and features], as is the case with other types of models, prior to their application to the data. As a result of this requirement for similarity of shape, the output ('y' part) of a training dataset for the Encoder-Decoder model must be of the same shape as the training dataset. Due to the fact that the model will forecast a specific quantity of time steps with a specific amount of features for all the input sample, the amount of features in the input sample will be a role of the number of features in the input sample.

4.3 Proposed Methodology

Tthe Encoder-Decoder Conv-LSTM model for forecasting stock market values and framework structure is demonstrated in fig 4.6. It has been proposed that the design of the proposed model, depicted in Fig. 4.6, is being constructed by obtaining the feature vectors from the stock values collection, based on the data normalization and standardization. During the partition step, with an 80/20 training/testing ratio, the dataset is divided into two parts. As the input vector (xt) of our proposed model, we employed five basic variables from previous time series data. The open, high, low, adj near, and close value make up the vector's elements. The anticipated close value of a stock, represented by y(t+1), is presented by:

$$y_{(t+1)} = F(x_t.y_t) \qquad (4.11)$$

Where F is the plotting function of the proposed model and yt+1 is the predicted values and yt is the original values and xt is the input values of the financial data. The encoder network's input has the structure (n steps, 1, n length, n features) and is multidimensional with time steps of 10. Different sorts of features are processed differently while creating these values. These are the features that change throughout time, such as open, close, low, high, and adj close. The ConvLSTM is fed with an input sequence with these features.

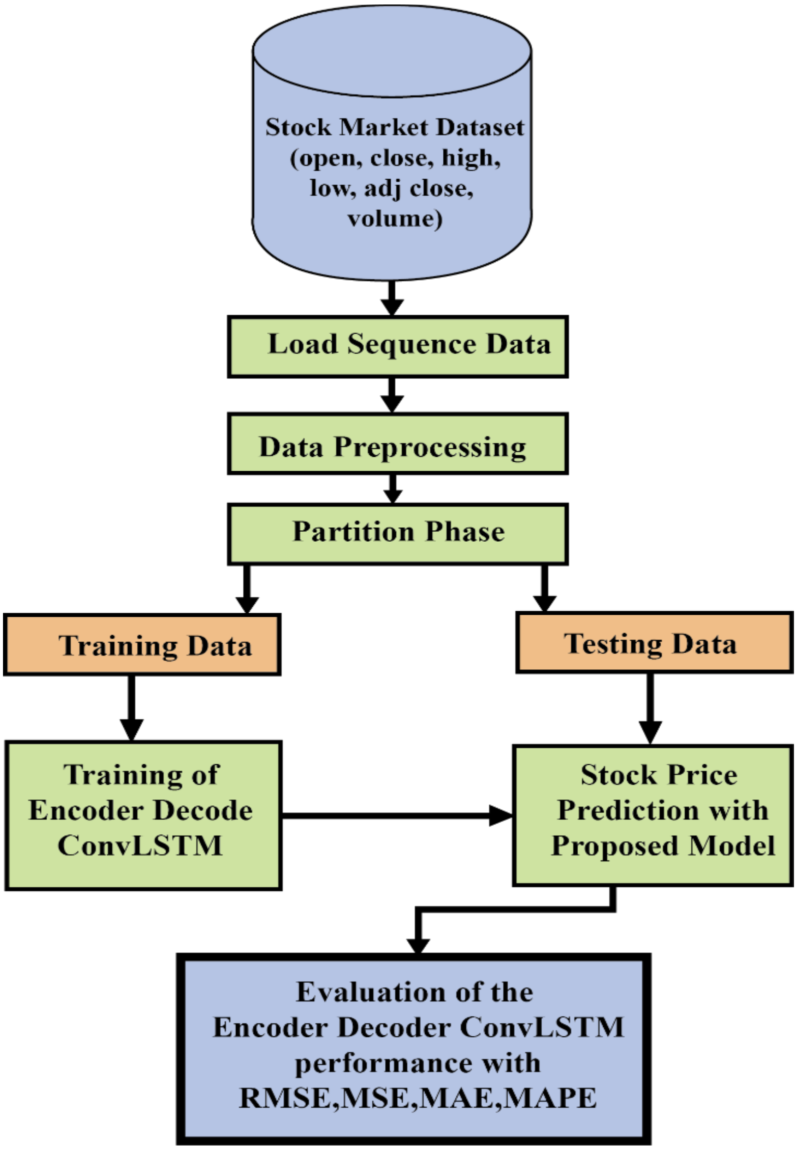

Figure 4. 6: The framework structure of the study

.

4.4 Proposed Model Architecture

In an Encoder-decoder ConvLSTM, ConvLSTM works as an encoder, and LSTM is used as a decoder. The ConvLSTM sometimes doesn't support series data; rather, a 1D ConvLSTM can read over series data and automatically extracts salient features. These would be then normally decoded by an LSTM decoder. The ConvLSTM and the LSTM models both require the same three-dimensional structure in the input data, even though several features are perceived as distinct channels that have the same impact in the end.

We propose that the architecture of our proposed model is depicted in fig. 4.7, be constructed by first extracting feature vectors from the individual power consumption dataset followed by the construction of our proposed model. We forecast the dataset into four different time resolutions like hourly, and daily basis. First, we load the dataset to further process. The individual stock market data is a multivariate dataset. we used several time series variables to forecast the time series data. There were 2,515 values in the S&P dataset and 3,113 in SBP dataset. The dataset is then normalized to avoid the overfitting of the model. The normalized dataset is then divided into a ratio of 80/20, training set, and testing set respectively.

We designed a simple but effective ConvLSTM model for the encoder consisting of two convolution layers, the outputs of which are then flattened. The following are the details for each layer of the model:

- Input Layer: receive time series data.
- First Convolutional Layer: The very first convolutional layers (*ConvLSTM2D*) read the inputs sequences and project the output sequence on feature maps. We read the input sequences using a kernel size of two time-steps and 128 feature mappings per convolutional layer by using "*Tanh*" as an activation function.
- Second Convolutional Layer: The next convolutional layer (*ConvLSTM2D*) does the same procedure on the feature maps, "Relu" as an activation, generated by the first pooling layer, trying to magnify any salient features.
- Flatten Layer: The distilled feature maps are flattened into a single long vector after the pooling layer, which may be used as input to the decoding procedure. The decoder receives the encoder's output as an input.

- Repeat-Vector Layer: For each timestep in the output sequence, the encoder's fixed-length outputs are first repeated once. After that, the sequence is passed into an LSTM decoding model. Throughout the output time step, the model must generate a result for each value.
- LSTM Layer: The decoder is designed as a hidden layer of 64 units. An activation function of the LSTM layer is "tanh". The decoder will then return the entire series with each of every 64 units providing a value to forecast what is going on in the output sequence.
- Fully Connected Layer: A fully connected layer is used before the final output layer to interpret the output vector each time. In particular, the output layer predicts the output sequence. This implies that identical layers will be deployed to each stage of the output sequence. This indicates that the decoder will function with an identical layer and output layer at each stage. The result was a time distribution wrapper that was employed for every step from the decoder by encapsulating interpret and output layers.
- Output Layer: output layer will forecast the next day of time-series data

  Figure 4.7 shows the structure of our proposed model. We have developed the proposed model parameters as given in table 4.1 and 4.2. These tables illustrates the filters for all convolution operations, the layer size and stride number, the pooling layer kernel, and the number of parameters for all layers of the model.

As we previously stated that we also forecast the S&P dataset in daily basis and SBP dataset in hourly basis. Therefore, we used the different architectures of the proposed model to forecast different time resolution datasets. 10 days timestep has been used for the daily forecasting dataset. It means 10 days time series has been passed to the model to forecast the next day closing value of the stock market. Similarly, 10 hours' time-series input sequence passed to forecast the next 01 hour in the SBP hourly forecasting model. Table 4.1 and 4.2 shows the architecture of the model respectively.

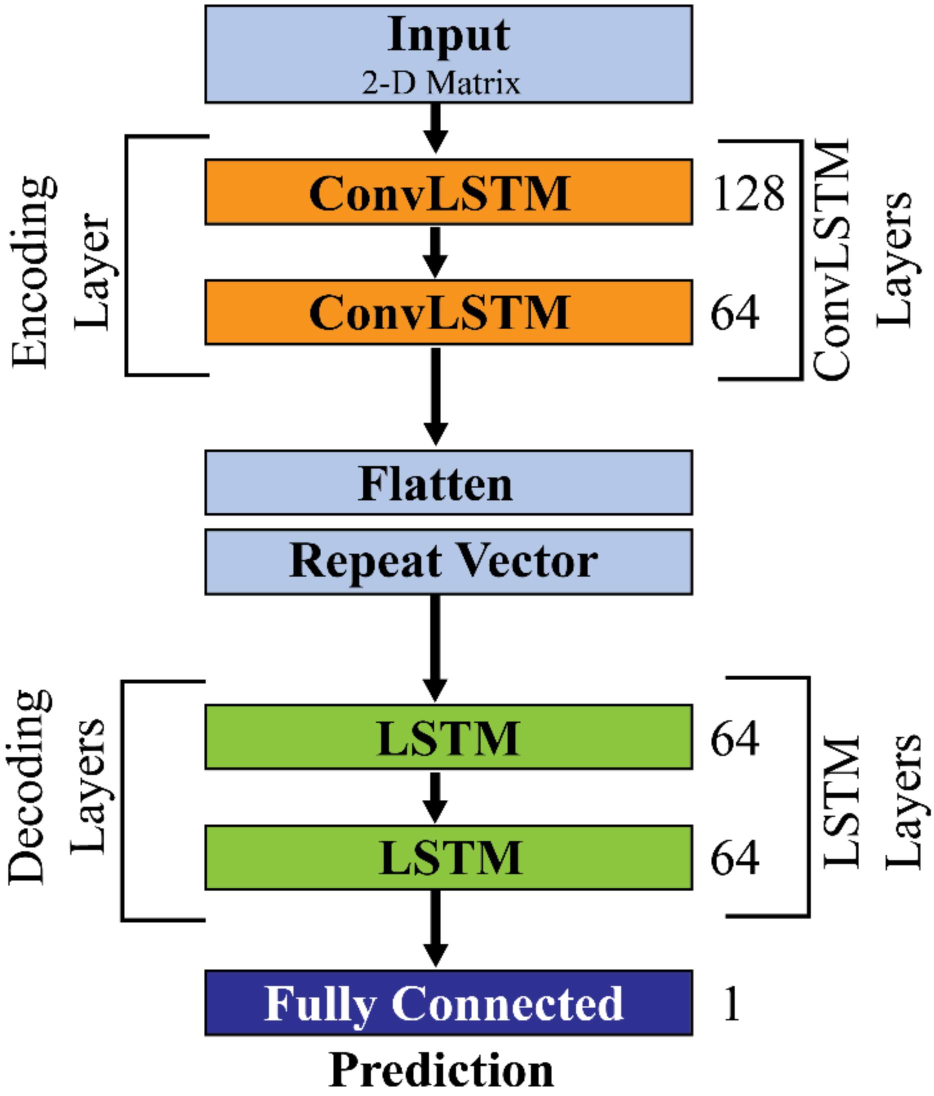

Figure 4. 7: Structure of Encoder-Decoder ConvLSTM

Table 4. 1 The Proposed Model Architecture of Encoder Module

Type	#Filter	Output Shape	Kernel Size
ConvLSTM2D	128	(None, 2, 1, 3, 128)	(1,3)
Activation (tanh)	-		-
ConvLSTM2D	64	(None, 1, 1, 64)	(1,3)
Activation (ReLu)	-		-
Flatten	-	(None, 64)	-

Table 4. 2 The Proposed Model Architecture of Decoder Module

Type	**#Filter**	**Output Shape**	**Kernel Size**
Repeat_Vector	-	(None, 1, 64)	-
LSTM(64)	-	(None, 1, 64)	-
Activation (tanh)	-		-
LSTM(64)	-	(None, 1, 64)	-
Activation (tanh)	-		-
Dense(100)	-	(None, 1, 100)	-
Dense(1)	-	(None, 1, 1)	-
Total Number of Parameters			**425,161**

4.5 Summary of Proposed Model

As a result, two ConvLSTM layers were employed as an encoder in economic time series information to learn the attributes of the variations between time steps as well as the internal properties of each time step. Tanh as an activation function has been used with the filter size of 128 for the first ConvLSTM layer and the second layer filter size set as 64 with relu as an activation function. The vector is received by the decoder from the encoder; however, unlike the encoder, which uses a ConvLSTM, the decoder is created by two LSTM layers. Each decoder cell's forecast is supplied as an input to the next decoder cell. Each decoder cell is composed of an LSTM with a filter size of 64 for each layer and tanh as an activation function, the output of which is fed into a fully connected layer, which produces the prediction. Our proposed model gives better results than other models because we combine various architectures to develop the proposed model. The detailed discussion of results is in chapter 5.

4.6 Forecast Metrics

Due to the fact that these results must be compared across various datasets and provide an accurate assessment of the quality of the forecast, the selection of forecast metrics is extremely important. In the benchmarking tool, we will compute multiple performance metrics at the same time, but we will only display the few that can be compared across all datasets. A few important features to consider for all forecast metrics is comparability across datasets, sensitivity to data transformation, and outliers.

4.6.1 The Mean Absolute Error (MAE)

Mean Absolute Error (MAE) computes the usual magnitude of inaccuracies in a sequence of forecasts without taking trends into account. It is the mean of the absolute inaccuracies among anticipated and observed values over the test sample, with all deviations given equal weight. The MAE formula is as follows:

$$\frac{1}{n}\sum_{i=1}^{n}|\tilde{\varphi}_i - \varphi_i| = \text{MAE} \qquad (4.12)$$

Properties:

- It determines the average absolute divergence between projected values and previous values when comparing forecasted and original values.
- In addition, it is referred to as the Mean Absolute Deviation (MAD) of a distribution (MAD).
- Extreme forecast errors are not penalized by the MAE because they are unaffected by any transformation of the data that occurs during the forecast process.
- The MAE should be as close to zero as possible to produce the best possible forecast.

4.6.2 The Mean Absolute Percentage Error (MAPE)

MAPE is a numerical quantity used to calculate the mean absolute proportion error function of a prediction. This common metric displays the proportion of the typical unquestionable mistake that happened. MAPE has a relatively small range of values. It is unable to establish the error's direction. The optimum MAPE value is near zero. The formula for evaluating the MAPE is as follows:

$$\frac{1}{n}\sum_{i=1}^{n}\left|\frac{\tilde{\varphi}_i - \varphi_i}{\varphi_i}\right| \text{x } 100 = \text{MAPE} \qquad (4.13)$$

Properties:

- Forecasting professionals use this metric to determine the ratio of average absolute error that happened throughout the process.
- Data transformation has an impact because it is unbiased of the level of measurement (e.g. MinMaxScaling), despite the fact that it is self-sufficient of the scale of size (e.g. MinMaxScaling and Log Scaling)
- Extreme forecasting errors are not subject to any kind of penalty under any circumstances.

- Produce a metric that is unaffected by the values that were observed in the series of observations that were used to create it.
- When MAPE is applied to data sets in which the original measures contain zeros, the results are ineffective because the original measures are not replicated (MAPE).
- Someone who does not have a statistical background will have no difficulty understanding and interpreting the findings.

4.6.3 The Mean Squared Error (MSE)

MSE is a measure for measuring system accomplishment. MSE calculates network accomplishment centered on the average forecast miscalculation. In predictive and regression analysis, this function is used to validate research findings. The MSE formula is as follows:

$$\frac{1}{n}\sum_{i}^{n}(\tilde{\varphi}_i - \varphi_i)^2 = \text{MSE} \qquad (4.14)$$

Properties:

- When comparing two consecutive periods, it is defined as the average squared deviation between the forecasted and actual values.
- This system penalizes forecasters who make significant errors in their forecasting while they are in the process of forecasting.
- MSE specifically emphasizes that large individual forecast errors have a significant impact on the overall forecast error and that large individual forecast errors are particularly problematic (Figure 1). (outliers)
- In addition, it becomes more sensitive with each increase in data scale and each data transformation that takes place.
- Because the value of this metric is dependent on both the dataset and the values, it is difficult to interpret. Another disadvantage of this metric is that it is not comparable across all datasets, which is a further drawback.

4.6.4 Root mean square error (RMSE).

One of the most frequent measures employed for regression error measurements is RMSE. It is equivalent to the MSE squared. The RMSE measures how evenly distributed these variations are. This measurement of error can tell you how much data is centered on the best-fitting line. According to fundamental principles, the optimum RMSE value is near zero. The RMSE formulae are as follows:

$$\sqrt{\frac{1}{n}\sum_{i}^{n}(\tilde{\varphi}_i - \varphi_i)^2} = \text{RMSE} \qquad (4.15)$$

4.7 Summary

In this chapter, we gave details of three prediction models for the stock market. The development of LSTMs for the time series forecasting has been discussed as well as the advantages and the disadvantages of the recurrent neural network. The architecture of the LSTM and ConvLSTM model is explained in detail and the vanishing gradient problem has been overviewed. In order to solve sequence-to-sequence problems we discuss the encoder decoder model and examined the functionality of the encoder and decoder specifically. After that the proposed methodology has been explained and the architecture of the proposed model is explained which includes the summary of both encoder and the decoder module. We compared the performances of the algorithms on the training and testing datasets and found that Encoder Decoder ConvLSTM show good performance as compared to the performance of traditional LSTM and ConvLSTM. So, we can conclude that the hybrid model has good performance for stock market prediction. In the end the performance metrices along with their properties are explained which have been used the measure the performance of the models used in this study.

CHAPTER 5: EXPERIMENTAL RESULTS

5.1 Experimental Setup

We ran a number of tests to evaluate the suggested models performance, evaluating it to existing created models and using metrics such as root mean square error (RMSE), mean absolute percentage error (MAPE), mean square error (MSE) and mean absolute error (MAE). To illustrate the value and utility of the system different models have been evaluated such as LSTM, ConvLSTM, and Encoder-Decoder ConvLSTM. First of all, we take the publicly available datasets which is Standard and Poor (S&P) and tested the results for all three above mentioned models. After testing the dataset successfully for all three models the next step is to compare the results so we can find the best model from these. The same dataset has been used for all three models so we can easily compare which model is performing best used the above-mentioned evaluation matrices. The second dataset that we used in this thesis is the dataset of State Bank of Pakistan which is not publicly available and the results of the three models are also evaluated using the SBP dataset.

5.2 Comparison of Models

The results of the proposed work which is Encoder Decoder ConvLSTM have been compared with the previous work done with different Neural Network Models such as LSTM and ConvLSTM using the same dataset. This comparison is based on the evaluation matrices MSE, MAE, RMSE and MAPE are shown in Table 5.1. In comparison to the other two methods (LSTM and ConvLSTM), the suggested framework has performed well as it can be perceived that the RMSE, MSE, MAPE, and MAE values for LSTM model are 6.620, 37.148, 0.226 and 2.925 respectively. Similarly, the evaluation matrices values of Conv-LSTM model are 6.716, 45.117, 0.447 and 5.753 which shows that the LSTM model performed well out of these two models in the past. Similarly, the values of RMSE, MSE, MAPE and MAE for the proposed model was 4.471, 19.993, 0.282 and 2.662.

Table 5. 1: Comparison of Proposed model with traditional models

Prediction performing on S&P 500				
Model	**RMSE**	**MSE**	**MAPE**	**MAE**
LSTM	6.620	37.148	0.226	2.925
CONV-LSTM	6.7169	45.117	0.447	5.753
ED Conv-LSTM (Proposed Model)	4.471	19.993	0.282	2.662

This shows that the proposed model has performed well out of all these three models and predicted the best possible result. This table only depicts the comparison of models and predictive tests were carried out to forecast the following day's concluding price movements in S&P 500 dataset. The graphical representation can be found in Figure 5.1.

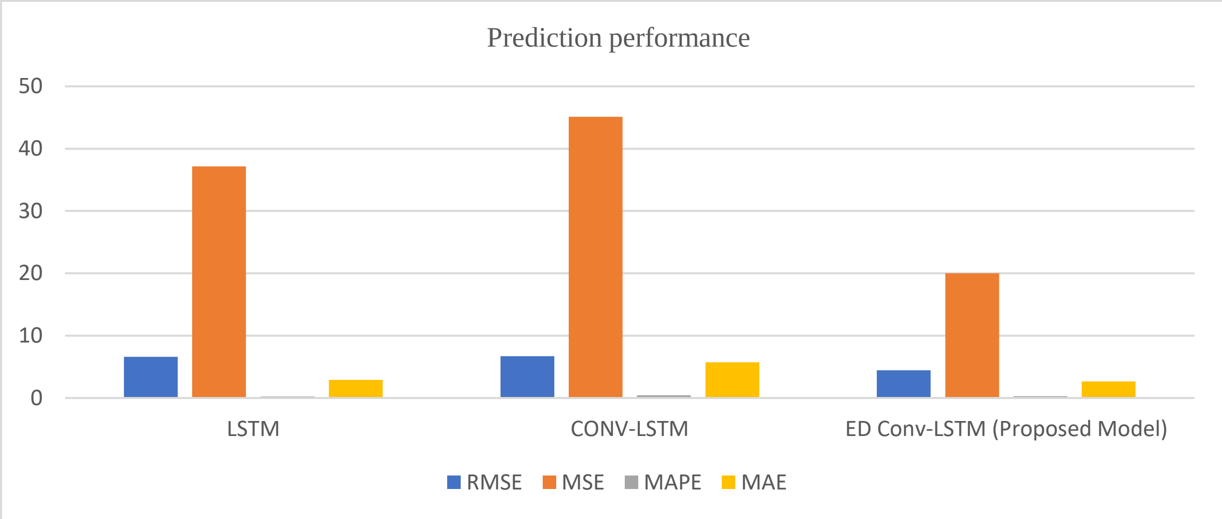

Figure 5. 1: Comparison of Proposed model with traditional models

5.3 Results for S&P Dataset:

5.3.1 Results with Traditional LSTM Model:

Different researches have been held out for stock market projection using the S&P 500 dataset as we already compared the proposed model with the previous model and it can be noticed that the suggested model has performed well. The dataset used in the proposed framework has been tested on traditional model LSTM as well and compared the results with the previous results. From Table 5.2, it can be noticed that the LSTM model has been used to foresee the stock market closing price for the following days which shows the approximately same results with very small differences as compared to previous research has been done. When compared to the results of [25] in which the author has used the same dataset the best results for the RMSE was 8.41 while when we applied the LSTM model using the different parameters results approved up to 6.63 for RMSE, Similarly, the MSE, MAPE and MAE are 37.1, 0.23 and 2.93 respectively for 400 epochs.

Table 5. 2: Results of traditional LSTM model for S&P Dataset

Sr.No	Epochs	RMSE	MSE	MAPE	MAE
1	100	9.06	82	0.67	8.73

2	200	8.26	68.3	0.59	7.66
3	300	12.4	54	0.9	11.4
4	**400**	**6.63**	**37.1**	**0.23**	**2.93**
5	500	9.71	94.4	0.7	9.01

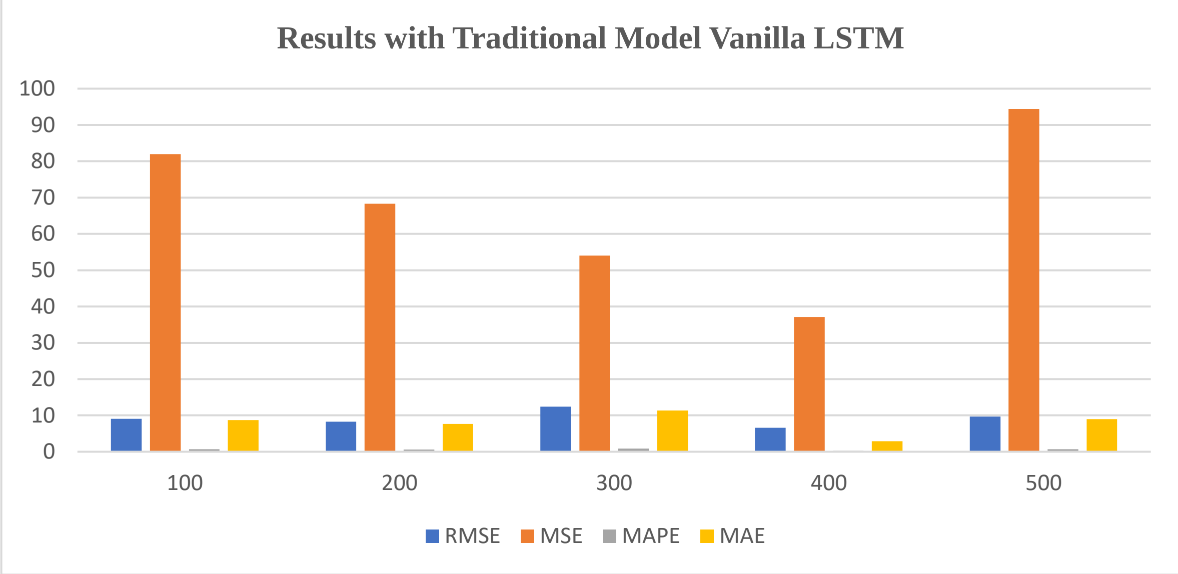

Figure 5. 2: Results of traditional LSTM model for S&P Dataset

In the Figure 5.2 the comparison using the different parameters has been shown as 1st plotting represent the 100 epochs and similarly the numbers 2,3,4 and 5 represent the 200, 300, 400 and 500 epochs respectively. It can be seen in Figure 5.2 that the number 4 represents the 400 epochs where the LSTM model-generated best results with other iterations tried to generate the best possible results.

5.3.2 Results with ConvLSTM Model:

The dataset used in the proposed framework has been tested on traditional model LSTM as well and compared the results with the previous results. From Table 5.3, it can be noticed that the LSTM model has been used to foresee the stock market closing price for the following days which shows the approximately same results. When compared to the other results in which the author has used the same dataset the best results are 6.71 for RMSE, Similarly, the MSE, MAPE and MAE are 45.1, 0.44 and 5.75 respectively for 400 epochs.

Table 5. 3: Results of ConvLSTM model for S&P Dataset

Sr.No	**Epochs**	**RMSE**	**MSE**	**MAPE**	**MAE**
1	100	10.77	115.9	0.64	8.21

2	200	9.17	84.0	0.55	7.51
3	300	8.44	71.8	0.51	9.95
4	400	6.83	46.6	0.43	3.03
5	**500**	**6.71**	**45.1**	**0.44**	**5.75**

5.3.3 Results with Encoder Decoder ConvLSTM:

After evaluating the results from traditional LSTM model the next step was to evaluate the results using the same dataset with Encoder Decoder ConvLSTM model. Up to a certain point, the proposed model was able to predict with greater precision using the same dataset as before, and the best results were obtained using the same dataset. The best possible outcomes have been obtained by varying various parameters in an experimental setting.

5.3.4 Results Using Batch size 32:

The batch sizes were one of the parameters that had an impact on the results, a comparison of different batch sizes was performed to determine which was the most effective. A certain number of samples must be processed in order for the internal model parameters to be updated, and this number is specified by this hyperparameter. A batch, if you think about it, is a for-loop that iterates over a set of samples and makes predictions about the outcomes. A comparison between the predicted output variables and the expected output variables is performed at the conclusion of the batch, and an error is calculated as a result of both of the comparisons. Based on this error, it is possible to improve the model, for example, by decreasing the error gradient as the model improves, as shown in the figure 5.3 .

This model was tested on the S&P dataset with batch sizes 32 , as evidenced by the results shown in Table 5.4. It can be perceived that the nest results achieved using 500 epochs with batch size of 33 for the evaluation metrices RMSE, MSE, MAPE, and MAE values for LSTM model are 5.31, 28.237, 0.319 and 4.0931 respectively. The comparisons of the results using the batch size 32 has been shown in the Figure 5.3.

Table 5. 4: Results Using Batch size 32

Sr.No	**Epochs**	**RMSE**	**MSE**	**MAPE**	**MAE**
1	50	15.385	236.691	0.956	12.217
2	300	12.901	166.443	0.943	12.316

3	**500**	**5.3138**	**28.237**	**0.319**	**4.0931**
4	700	17.517	306.847	1.306	16.961

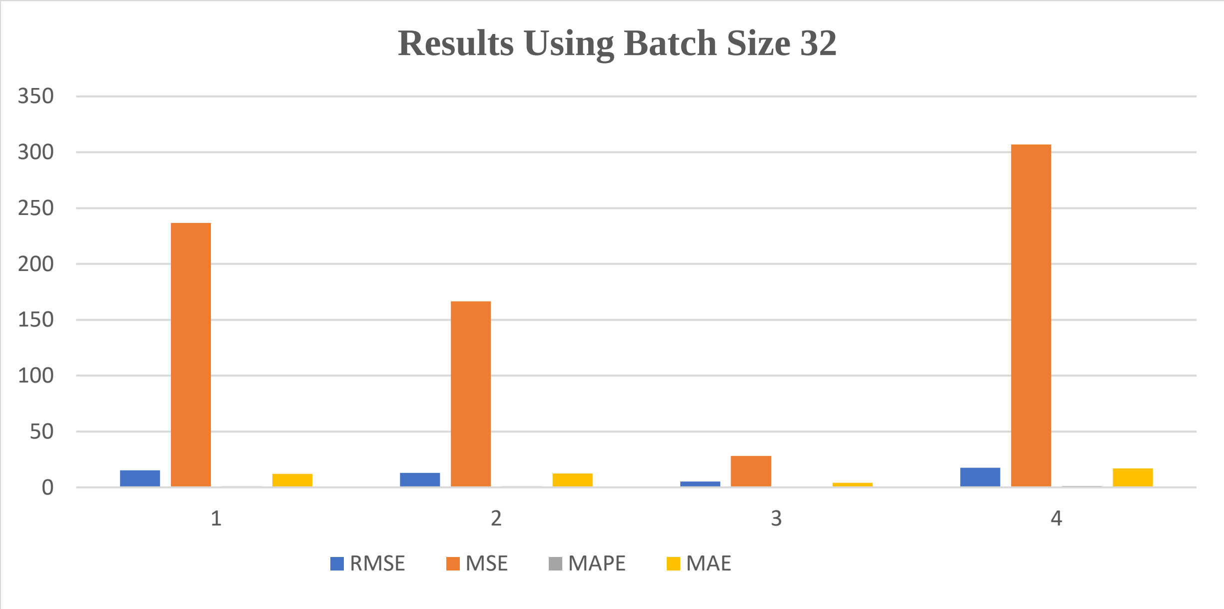

Figure 5. 3: Results Using Batch size 32

5.3.4 Results Using Batch size 16:

This model was tested on the S&P dataset with batch sizes 32 and 16, as evidenced by the results shown in the following table, which show that the model performed admirably. This model was tested on the S&P dataset with batch sizes 16 , as evidenced by the results shown in Table 5.4.

Table 5. 5: Results Using Batch size 16

Sr.No	Epochs	RMSE	MSE	MAPE	MAE
1	50	10.865	118.054	0.645	8.278
2	100	6.716	45.117	0.447	5.753
3	**150**	**4.471**	**19.993**	**0.282**	**3.662**
4	200	7.038	49.541	0.451	5.883

From the above table , it can be found that the best results of the stock market calculation using the S&P 500 dataset by using the Encoder Decoder ConvLSTM model have been generated having batch size of 16 in the model. The best results that have been generated after using the different parameters were 4.471, 1.993, 0.282 and 3.662 for RMSE, MSE , MAPE and MAE. Different epochs have been applied and it has been observed that at 150 epochs model generated the best results after increasing the epochs from 150 to 200 it can be seen that errors have been increased again. The comparisons of the results using the batch size 16 has been shown in the Figure 5.4.

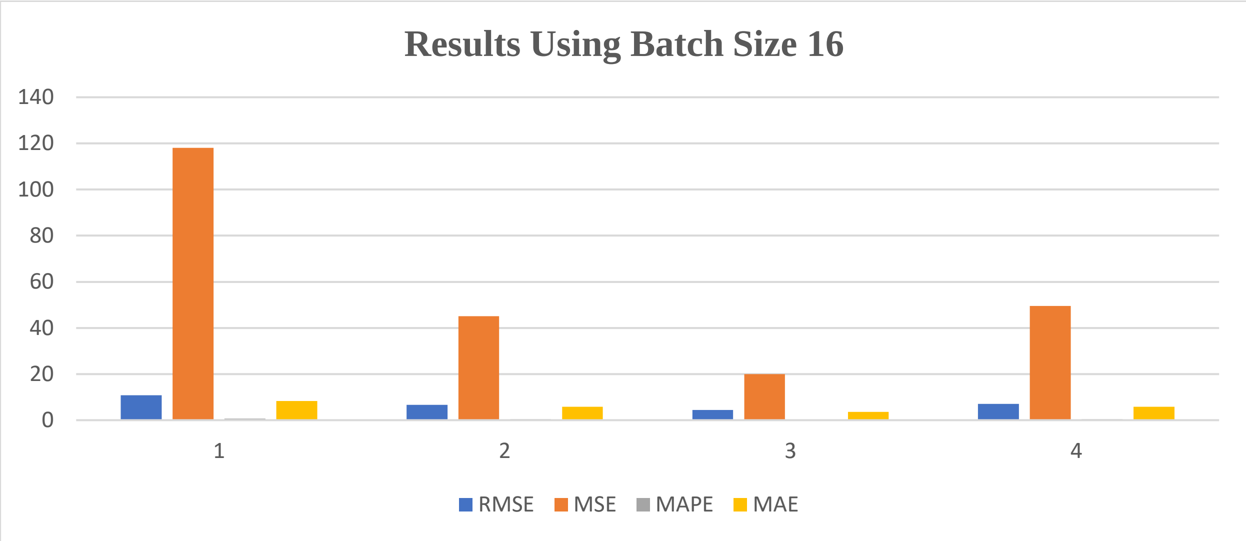

Figure 5. 4: Results Using Batch size 16

After evaluating that the best result can be achieved using the batch size 16 we have plotted the testing data for all the epochs applied to predict the next day closing price using the proposed model encoder decoder ConvLSTM. The figure 5.5 represent the plotting of testing data using 50 epochs in which blue line representing the original test data and orange line represent the test predicted data using the proposed model.

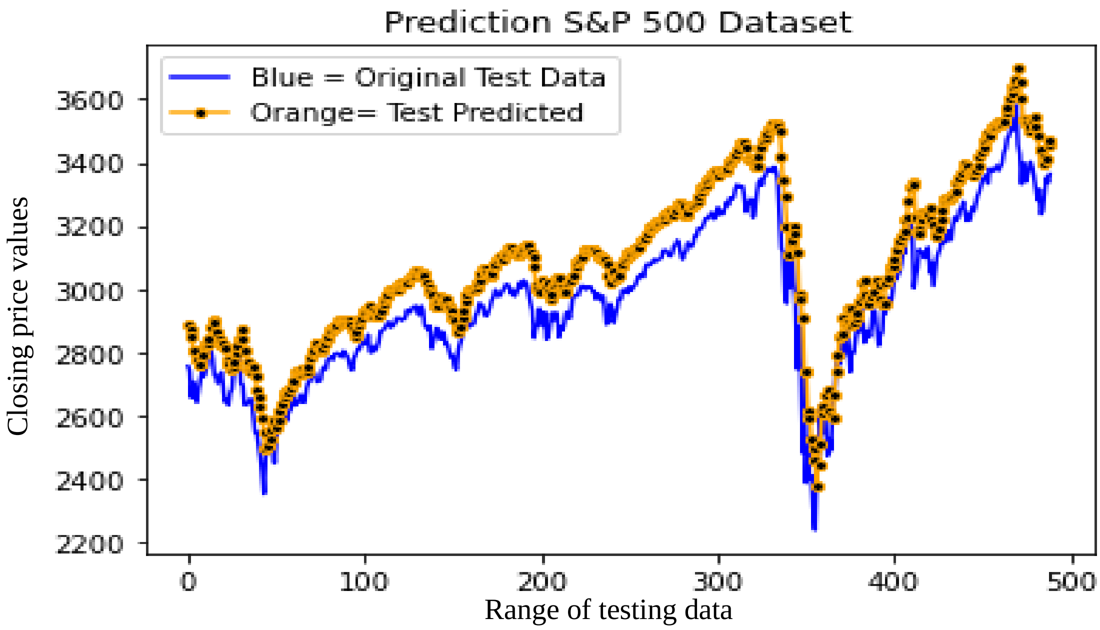

Figure 5. 5: Prediction with 50 epochs

Similarly, different parameters have been used to predict the best possible result and the figure 5.6 represent the plotting of testing data using 100 epochs and the figure 5.7 represent the testing data using the 150 epochs which is the best predicted result using the proposed model. in which blue line representing the original test data and orange line represent the test predicted data using the proposed model.

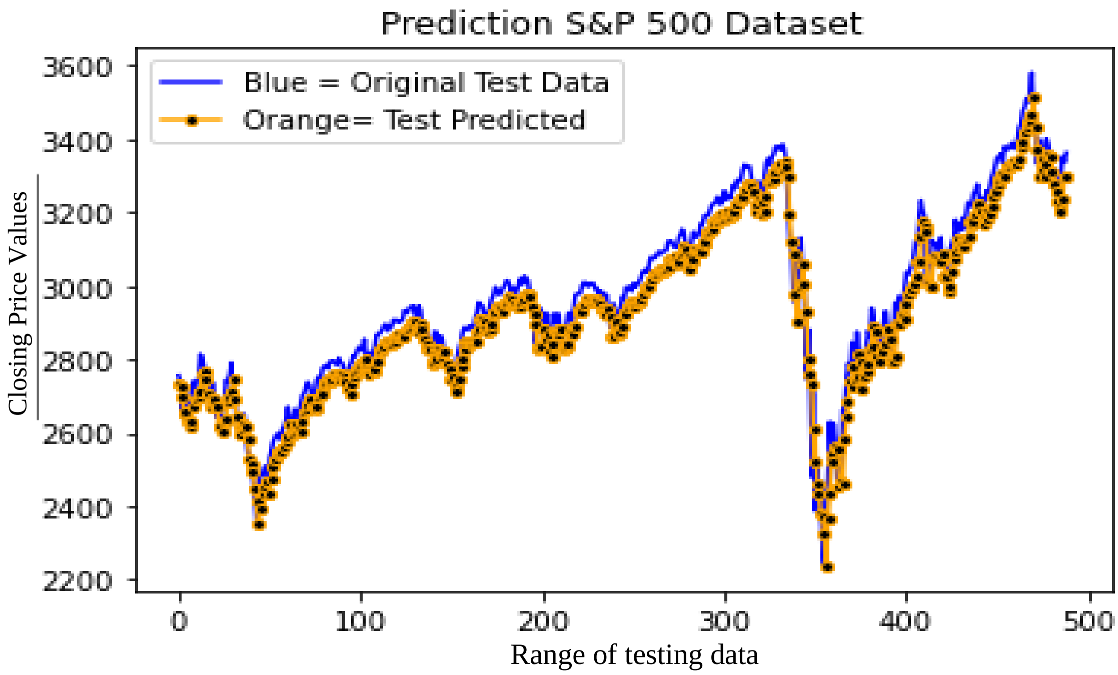

Figure 5. 6: Testing prediction with 100 epochs

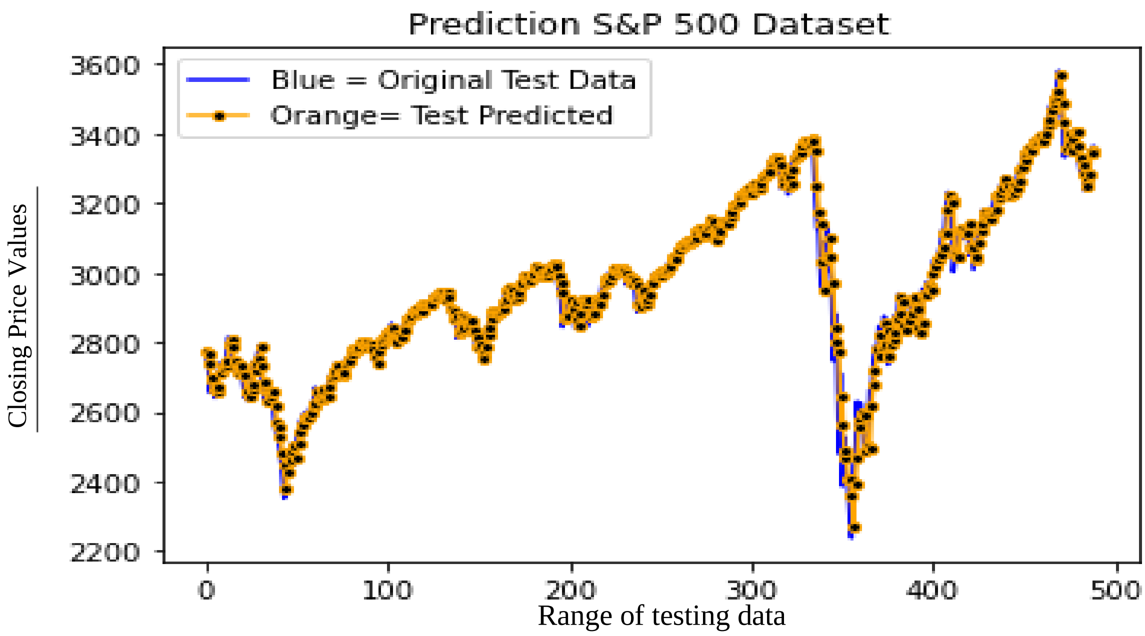

Figure 5. 7: Testing prediction with 150 epochs

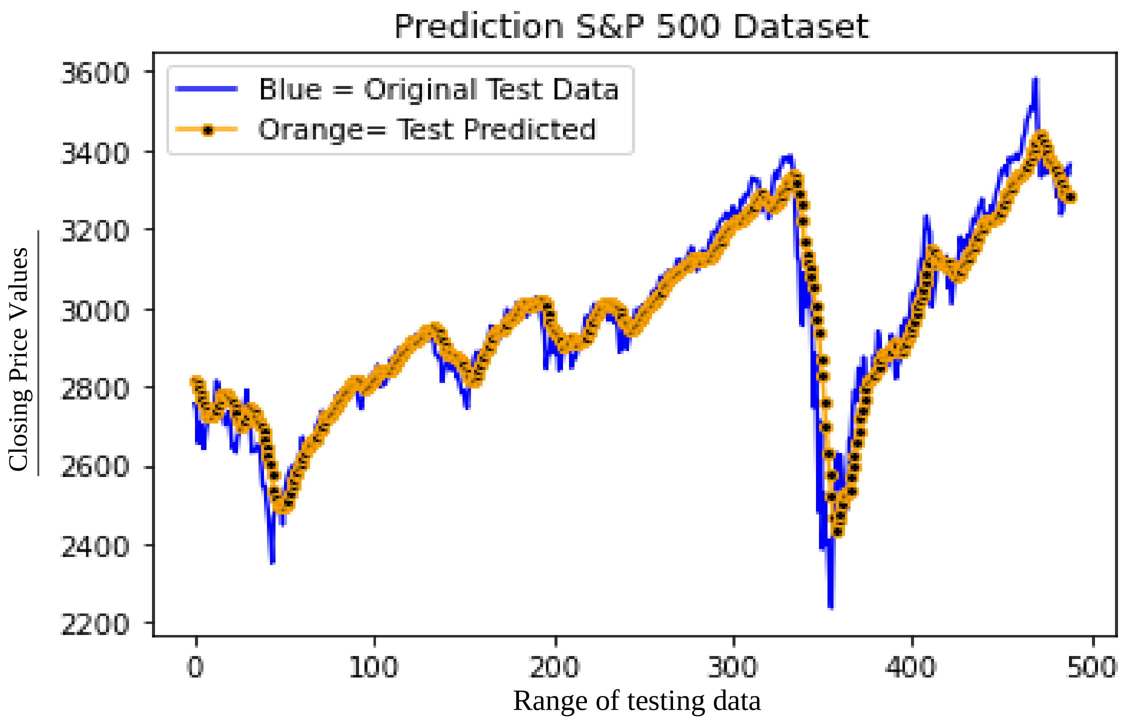

Figure 5. 8: Testing prediction with 200 epochs

Following the identification of the optimal model by utilizing the various parameters discussed in the preceding section, the plotting of training and testing data is depicted in the following figures. As previously stated, it can be seen in Table 4 that as the number of iterations increases from Figure 5.6 to Figure 5.8, the prediction performance continues to improve, with the best prediction of data being discovered using the 150 epochs. Figure 5.8 shows the plotting of testing data using 200 epochs after increasing the number of epochs even further to 200. The prediction performance of the model has begun to deteriorate once more after this increase. As a result, it has been discovered that the proposed model provides the best results for next day prediction on closing price when using 150 epochs with a batch size of 16 for the closing price.

5.4 Loss Functions:

Calculating the difference among the output of our algorithms and the objective value that we have specified in our specifications is accomplished through the use of loss functions. It is critical to understand that the distance between our computed output and the true value is measured in terms of distance from the true value, not in terms of distance from the mark, as defined by the loss function. Loss functions are an excellent tool to have in your toolbox if you want to save money on your next purchase. When performing regression, the use of loss functions can assist you in determining which line of fit is the most appropriate for your situation. They are used in order to find the best line of fit by minimizing the overall loss of all the points with the predicted value from the line, which is determined by the line's prediction, which is determined by the line's prediction, which is determined by the line's prediction, which is determined by the line's prediction. In order to influence the way that the weights of perceptron's and neural networks are updated during the training and application phases of their respective algorithms, loss functions must be used in conjunction with them. As the magnitude of the loss increases, so does the magnitude of the corresponding correction, which is inversely proportional. As the margin of error is reduced to the smallest possible value, the accuracy of the model improves as a result. When developing machine learning applications, it is necessary to weigh up the tradeoff between the amount of data that must be updated and the amount of data loss that must be avoided at all costs. When viewed in conjunction with the models predictive performance, the loss functions of the various epochs applied follow the pattern depicted in the Figure 5.9 , with the loss functions of the various epochs applied continuing to decrease from part a to part C before beginning to rise again.

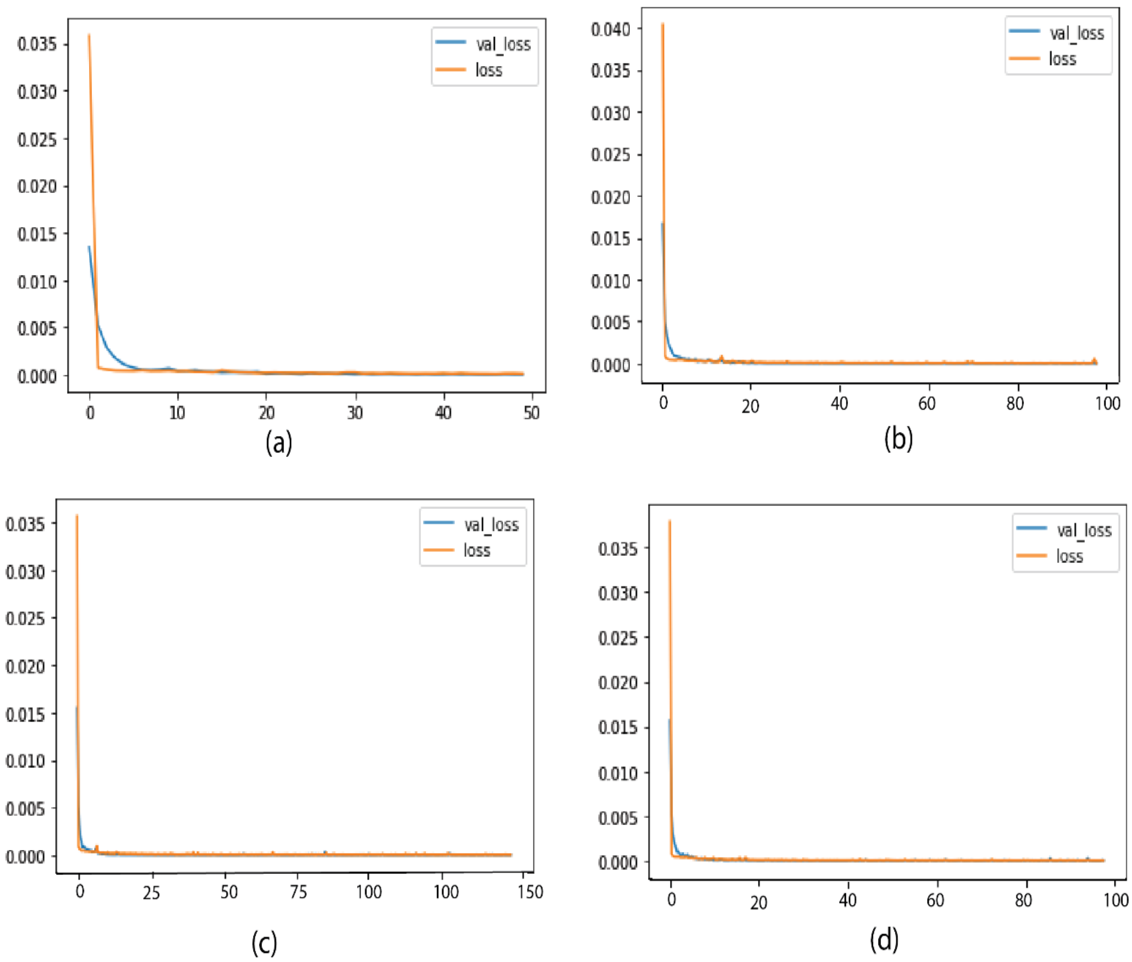

Figure 5.9: Loss function of all epochs tested for S&P dataset

5.5 Results for SBP Dataset:

After Applying different epochs for the standard and poor 500 index dataset and finding the best possible model for stock market prediction the next step was to apply the same model on the State Bank of Pakistan's dataset and predict the next day closing price. Details about the dataset have already been discussed in the above chapter and the results after applying different iterations have been shown in the following tables.

5.5.1 Results with Traditional LSTM Model:

The dataset used in the proposed framework has been tested on traditional model LSTM as well and compared the results with the previous results. From Table 5.6, it can be noticed that the LSTM model has been used to foresee the stock market closing price for the following days which shows the next day closing best prediction price was using 200 epochs which is 0.004 for RMSE, Similarly, the MSE, MAPE and MAE are 4.645, 0.99 and 0.005 respectively.

Table 5. 6: Results of traditional LSTM model for SBP Dataset

Sr.No	**Epochs**	**RMSE**	**MSE**	**MAPE**	**MAE**
1	10	0.044	0.0019	1.726	0.009
2	50	0.020	0.0004	1.017	0.009
3	100	0.017	0.0002	0.003	0.007
4	150	0.006	4.455	0.69	0.004
5	**200**	**0.004**	**4.645**	**0.9945**	**0.005**
6	400	0.006	4.754	0.68	0.004

5.3.2 Results with ConvLSTM Model:

The dataset used in the proposed framework has been tested on model ConvLSTM as well and compared the results with the previous results. From Table 5.7, it can be noticed that the ConvLSTM model has been used to foresee the stock market closing price for the following days which shows best results are 0.006 for RMSE, Similarly, the MSE, MAPE and MAE are 2.326, 0.45 and 0.375 respectively for 200 epochs.

Table 5. 7: Results of ConvLSTM model for SBP Dataset

Sr.No	**Epochs**	**RMSE**	**MSE**	**MAPE**	**MAE**
1	10	0.079	0.00624	1.852	0.034

2	50	0.030	0.00093	1.630	0.007
3	100	0.010	0.00015	1.477	0.008
4	150	0.007	2.645	0.575	0.051
5	**200**	**0.006**	**2.326**	**0.375**	**0.045**
6	400	0.007	2.587	0.557	0.068

5.5.1 Results with Encoder Decoder Conv LSTM Model:

Different epochs have been applied to the SBP dataset using the Encode Decoder ConvLSTM model as well and results are shown in Table 5.8. This dataset is not publicly available that's the reason that no previous research work has been done on this dataset. So, this was the main problem that no work has been done before on this dataset and we were not able to compare the results of proposed model results on this dataset. For this reason, the model has been applied to the publicly available dataset (Standard and Poor 500 index) Which has been explained in the above section. It has been find out that the results showed gradual increment upto 400 epochs as the results for evaluation metrices RMSE, MSE, MAPE and MAE are 0.005, 2.603, 0.557 and 0.003. After increasing the epochs to 500 a very small changes in results have been noticed. The graphical representation of the comparisons of the results of different parameters are shown in Figure 5.10.

Table 5. 8: Results with Encoder Decoder Conv LSTM Model for SBP dataset

Sr. No	**Epochs**	**RMSE**	**MSE**	**MAPE**	**MAE**
1	10	0.054	0.0037	1.949	0.011
2	50	0.012	0.0015	1.522	0.009
3	100	0.0109	0.0001	1.436	0.008
4	150	0.007	6.185	0.9945	0.005
5	200	0.006	3.631	0.69	0.004
6	**400**	**0.005**	**2.603**	**0.557**	**0.003**
7	500	0.005	2.524	0.555	0.003

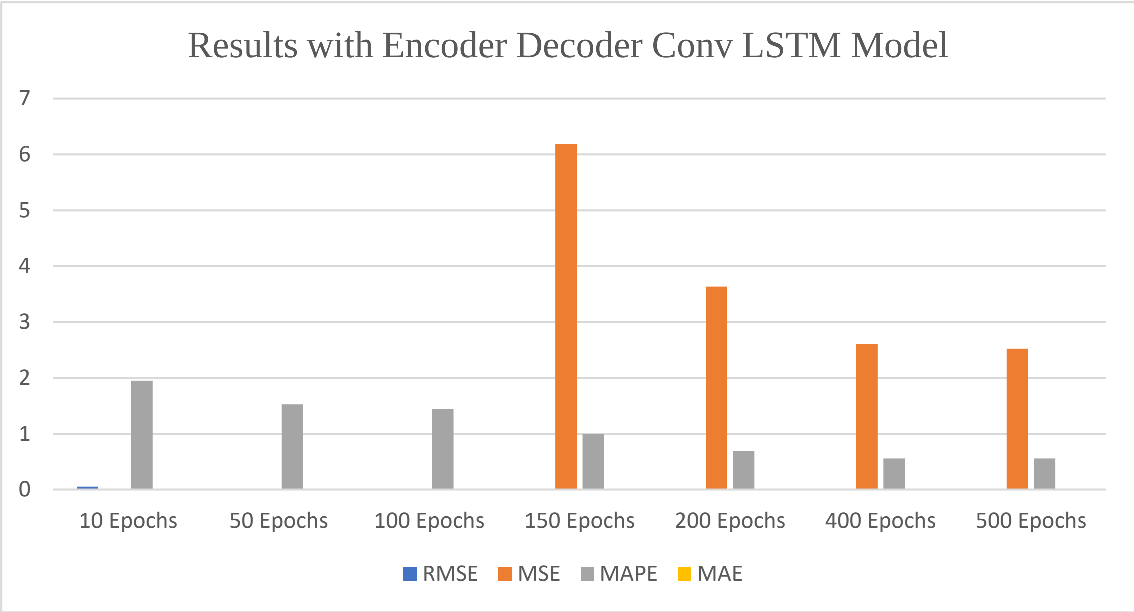

Figure 5. 10: Results with Encoder Decoder Conv LSTM Model for SBP (AUDUSD) dataset

After evaluating that the best result can be achieved using the proposed model encoder decoder ConvLSTM we have plotted the testing data for all the epochs applied to predict the next day closing price The figure 5.11 represent the Complete plotting of all data using 50 epochs in which blue line representing the original test data and orange line represent the test predicted data using the proposed model.

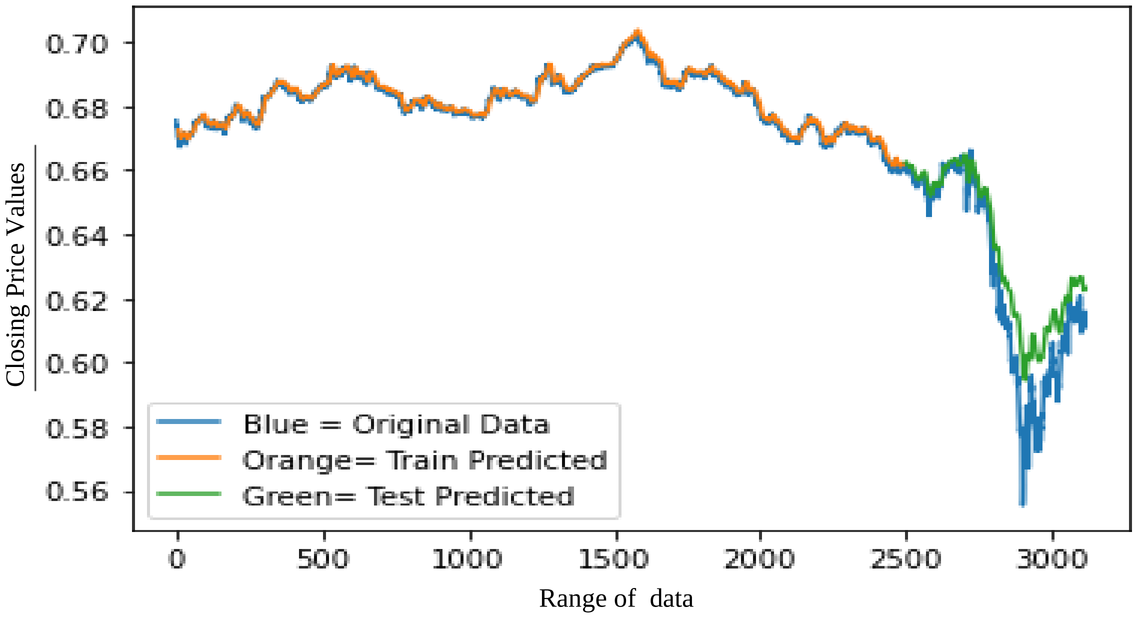

Figure 5. 11: Complete plotting with testing and training prediction of SBP (AUDUSD) dataset with 50 epochs

After plotting the complete dataset using the proposed model for the 50 epochs in the above figure different parameters have been applied to find the best possible results for the SBP dataset . Figure 5.12 represent the plotting of testing data using 50 epochs and the figure 5.13 represent the testing data using the 100 epochs for the SBP dataset result using the proposed model. in which blue line representing the original test data and orange line represent the test predicted data using the proposed model.

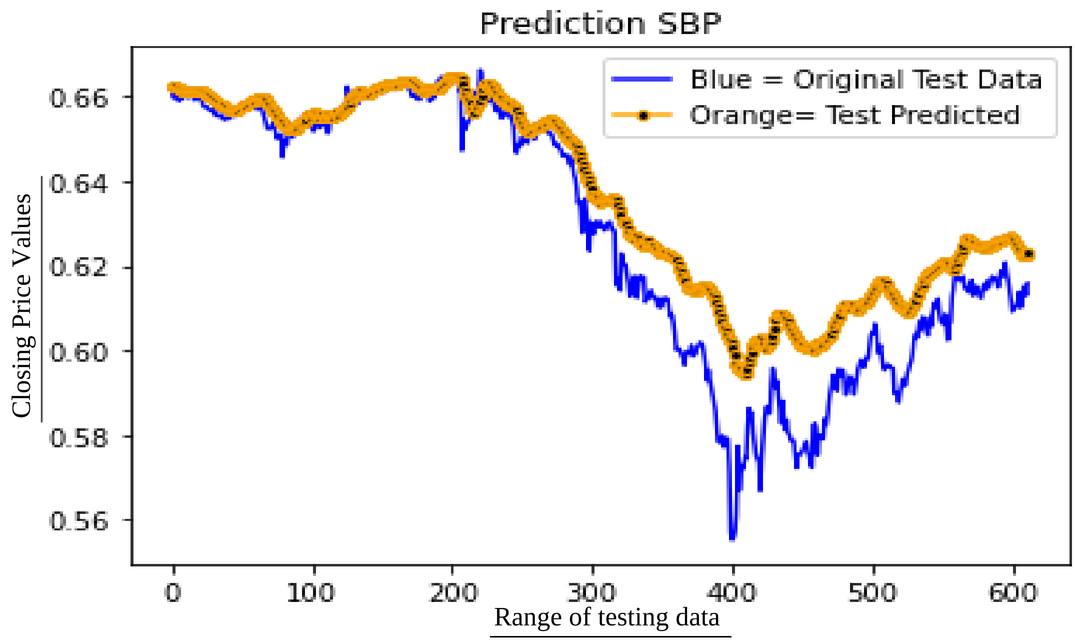

Figure 5. 12: Plotting of testing prediction of SBP (AUDUSD) dataset with 50 epochs

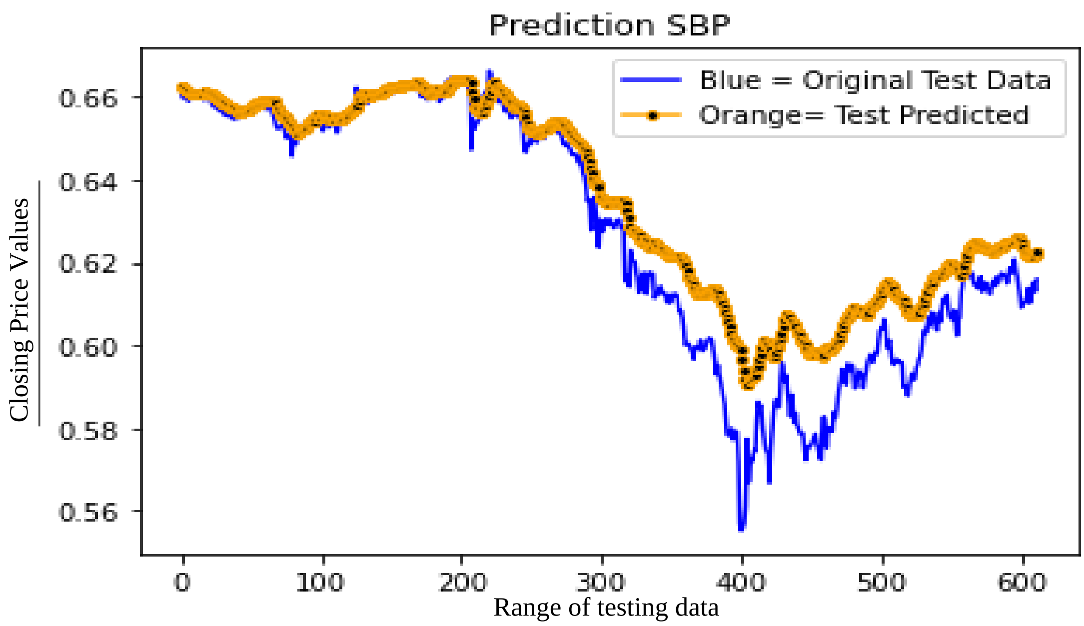

Figure 5. 13: Plotting of testing prediction of SBP (AUDUSD) dataset with 100 epochs

From Figure. 5.14 and Figure. 5.15 represent the plotting of testing data using 200 epochs and 500 epochs. The prediction performance keeps on increasing and after 400 epochs the prediction performance gradual increment has been stopped as when epochs increased from 400 to 500 a very little changings have been noticed for all the evaluation matrices as compared to the above performance. So, it has been observed from Figure 5.15 that the proposed model predicted the next day stock prices well by using the 500 epochs for the State Bank of Pakistan's dataset.

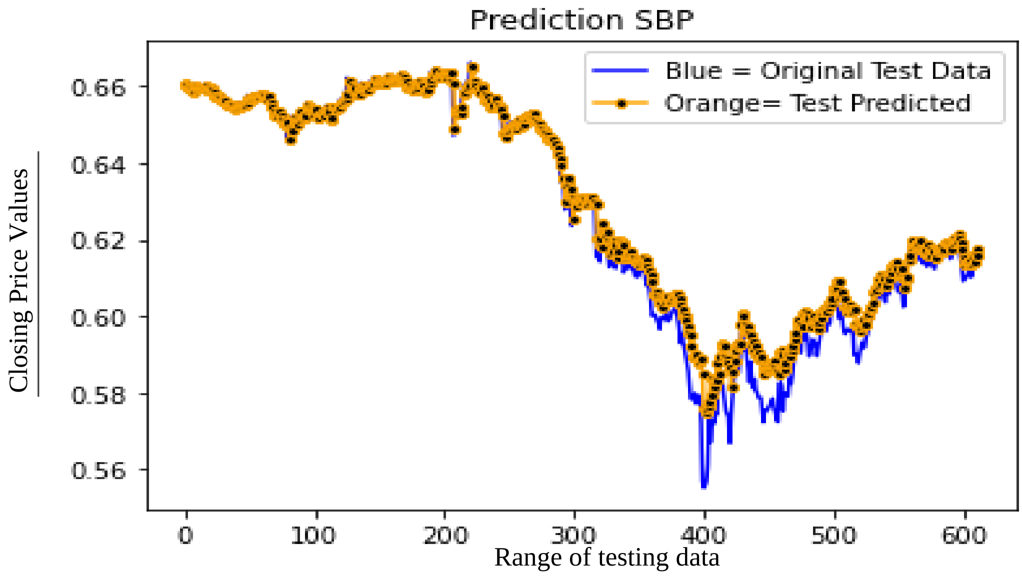

Figure 5. 14: Plotting of testing prediction of SBP (AUDUSD) dataset with 200 epochs

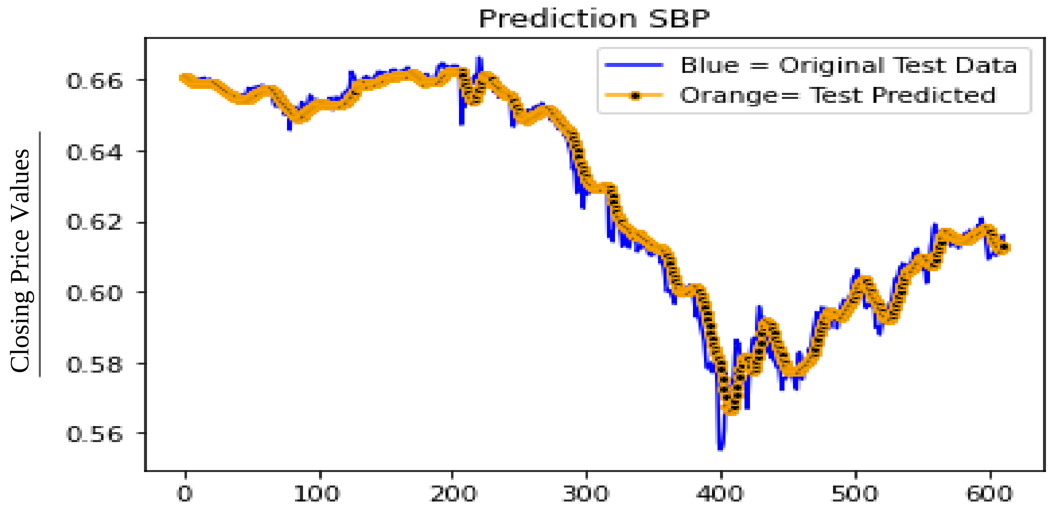

Figure 5. 15: Plotting of testing data of SBP (AUDUSD) with 500 epochs

5.5.2 Loss Functions for SBP dataset:

The loss functions for the SBP dataset for all the iterations applied are shown in figure 5.16 and it can be seen that by increasing the epochs the loss functions keep on decreasing upto some extent and in parts e and f it can be seen that the loss function for both 400 epochs and

500 epochs are almost same which means that the best prediction with the lowest lost function is at 500 epochs and after that no effect has been seen in the loss functions so we stopped the testing of the model at 500 epochs.

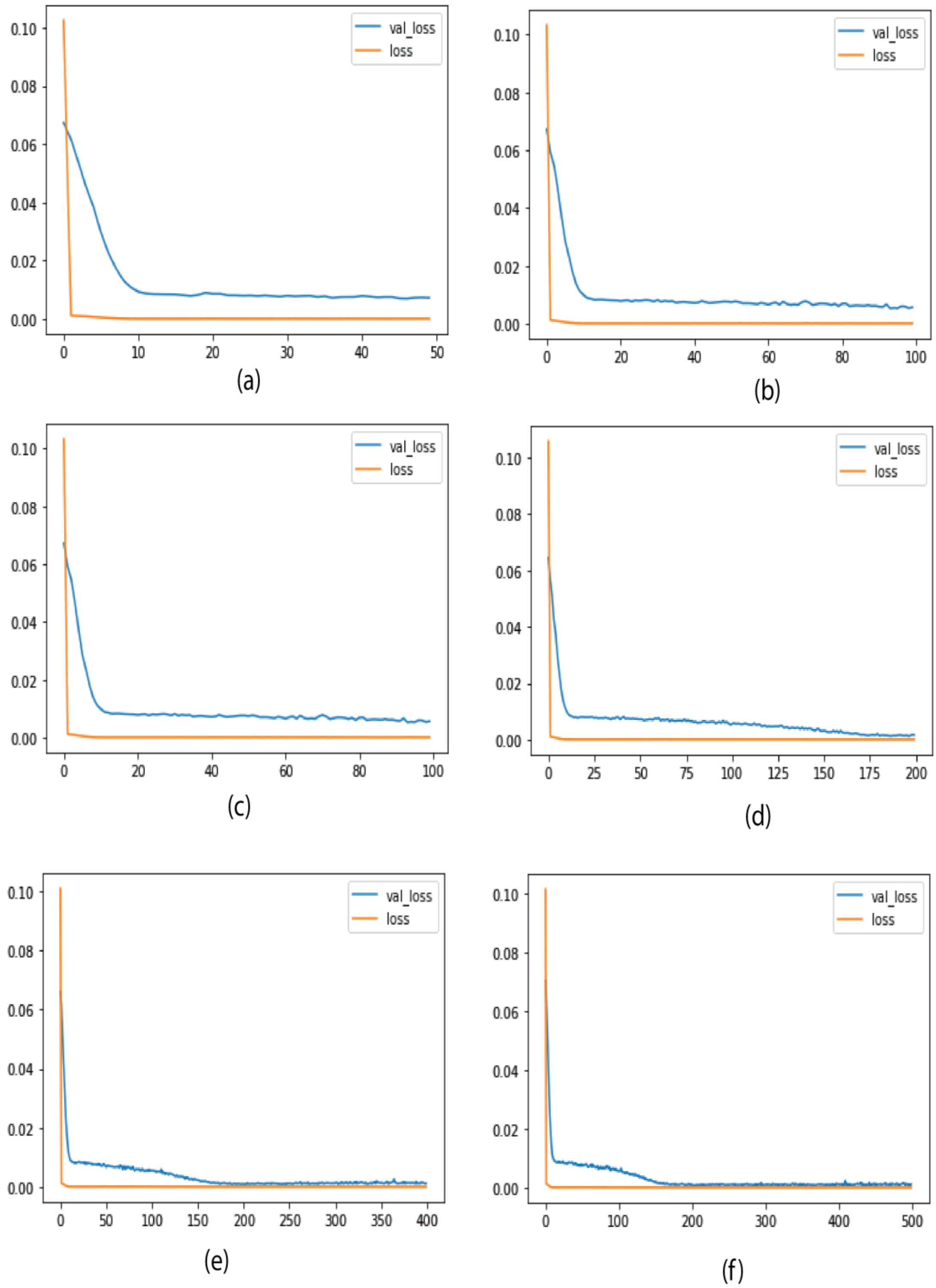

Figure 5.16: Loss functions of SBP (AUDUSD) dataset with all epochs applied

5.6 Comparison of Proposed model with previous research done

When compared to other reference model's accuracy findings, the testing findings on the S & P 500 indicator in Table 5.9 demonstrate that our Encoder-Decoder ConvLSTM model has the best performance metrics results. It can be seen that the best result of RMSE in [25] and [24] is 8.416 and 16.471 while the proposed model's best result for RMSE is 4.471 and similarly the proposed model has performed well in the case of MAE and MAPE when compared to the other model. The graphical representation of the comparisons can be seen in Figure 5.21 in which blue line representing the RMSE, orange line representing the MAE and grey representing the MAPE respectively.

Table 5. 9: Comparison of Proposed model with previous research done

Model	**Performance Metrics**			**Reference**
	RMSE	**MAE**	**MAPE**	
Deep LSTM	8.416	-	0.143	[25]
LSTM	16.471	11.40	0.53	[24]
ED Conv-LSTM (Proposed Model)	**4.471**	**2.662**	**0.282**	**Proposed**

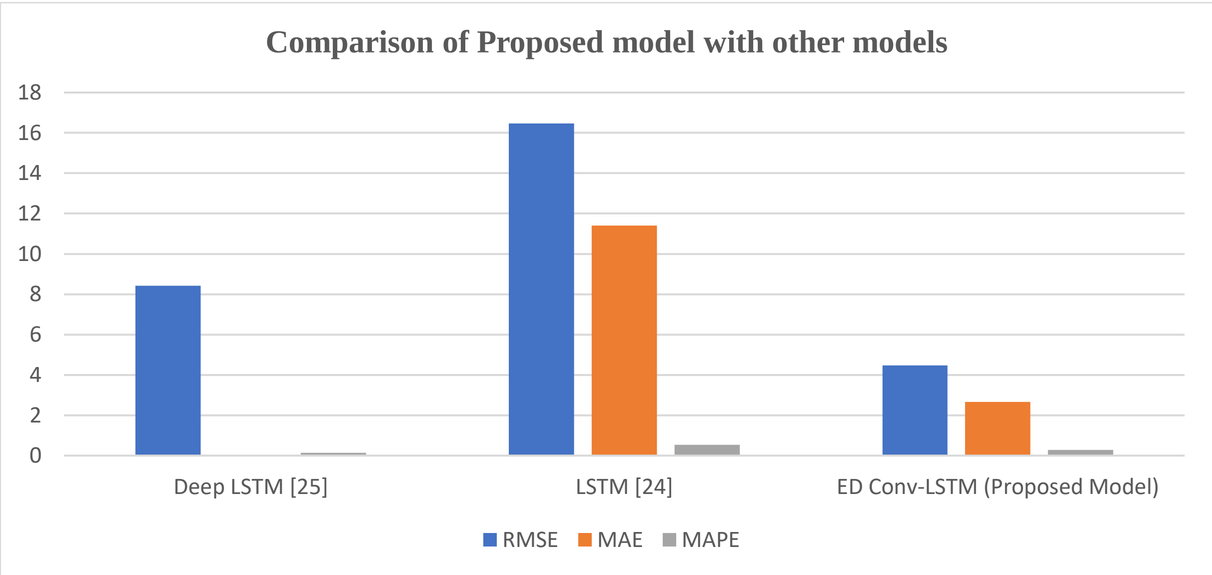

Figure 5.17: Comparison of Proposed model with previous research done

CHAPTER 6: CONCLUSION

6.1 Conclusion

In this thesis document, the time series analysis and time series forecasting has been investigated using novel deep learning methods, which are traditional LSTM models and the proposed model encoder decoder ConvLSTM. We chose historical data rather than real-time data for testing purposes not only because it was more time efficient, but also because it allowed us to compare models and trading strategies using the same testing data, which was advantageous. For Stock Data Prediction, cutting-edge deep learning has been presented for the prediction of the next day closing price of the stock. The performance of the model was evaluated in this study in comparison to that of traditional models such as the LSTM and the ConvLSTM models, among others. First, the proposed model has been compared with the other models using the same dataset to check the efficiency of the model. After evaluating that the proposed model was generating good results as compared to the other models, the collected datasets have been tested on the model using different parameters to find the best possible result. It was decided to use the performance metrics RMSE, MAE, MAPE, and MAE for this comparison. It is evident from the results that for the next day prediction, the proposed model results beat the LSTM models and ConvLSTM model, and We were able to develop a method for successfully predicting the stock market with the help of historical data, which we then combined with a sound trading strategy to generate profits from stock trading. The proposed model is a hybrid machine learning model consisting of two models ConvLSTM which are functioning as an encoder and the two LSTM layers that are functioning as the decoder. Two datasets have been utilized for this work which is the standard and Poor 500 Index dataset which has collected from Yahoo Finance and it was ten years of the dataset from 2010 to 2020. Similarly, the proposed model has also been tested on another dataset that belongs to the State Bank of Pakistan, which is the six months dataset. This dataset was not publicly available and no work has been done before on the dataset like these. That was the main reason that we have also tested our model on the publicly available dataset so that we can find out the efficiency of our model by comparing it with already published work. The evaluation metrics results of the proposed framework using the S&P 500 dataset generated the best results when compared to the previous work done for stock market prediction using the same dataset.

6.2 Contributions

We propose a combination of Recurrent Neural Network in developing the stock prediction model. Improvement in performance of prediction models is observed when the Time Series data's are considered while predicting the stock market. We find such stock markets that are hard to predict for the traders. We identify such stock markets that are more affected by dynamic behavior of data. We propose the usage of deep learning techniques and show improvement in the prediction accuracy. The contrast of the suggested model with the traditional LSTM models is also presented in this thesis. The review and comparison of recent developments and models for the stock market prediction have also been discussed in this thesis.

6.3 Future Work

We must continue to test the current methods with additional data as the data continues to flow into the system. Eventually, we will need to test our model with even more data to ensure that it is applicable in other situations as time goes on.

It is possible to use this model in conjunction with other time-series datasets, and Reinforcement learning can be used to generate long-term predictions by modifying the model as needed. Reinforcement learning has emerged as one of the most favorable research areas in the field of machine learning, particularly in the context of robotics. RL is considered to be one of the most profitable ways to make money due to the perfection with which the stock market can be modeled with the software. Despite initial positive outcomes from the study, much more work is needed to ensure that this model can be put into practice. We decided to postpone Reinforcement Learning until a later date primarily because of time and computing power constraints.

This research study used only stock market ticker symbols to collect time series data. For collecting more stock relevant data, keywords relevant to stock markets can be used. It will result in collecting more stock-relevant data and hence will improve the quality of predictions. We have focused on companies in the technology sector. The research can be extended to companies in other sectors like energy, healthcare, etc. Predictions can be performed for more than 01 day to evaluate the effectiveness of the LSTM and external media on the stock market prediction.

REFERENCES:

[1] S. Makridakis, Wheelwright. and McGEE, FORECASTING: Methods and Applications. John Wiley and Sons, Inc, second ed, 1997.

[2] J.Lin, K. Keogh, M. Eamonn; Lonardi, Stefano; Chiu, Bill (2003). "A symbolic representation of time series, with implications for streaming algorithms". Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. New York: ACM Press. pp. 2– 11. CiteSeerX 10.1.1.14.5597. doi:10.1145/882082.882086. S2CID 608 4733

[3] D. Nelson, A. M. Pereira, and A. A. Oliveira, “Stock Market’s Price Movement Prediction with LSTM Neural Networks,” in International Joint Conference on Neural Networks (IJCNN 2017), 2017, pp. 1–10, doi: 10.1109/IJCNN.2017.7966019.

[4] D. K. Kılıc¸ and O. U ¨ gur, “Multiresolution analysis of s&p500 time ˘ series,” Annals of Operations Research, vol. 260, no. 1-2, pp. 197–216, 2018.

[5] P. Li, C. Jing, T. Liang, M. Liu, Z. Chen, and L. Guo, “Autoregressive moving average modeling in the financial sector,” in 2015 2nd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE). IEEE, 2015, pp. 68–71.

[6] S. A. Verma, G. Thampi, and M. Rao, “Inter-comparison of artificial neural network algorithms for time series forecasting: Predicting indian financial markets, International Journal of Computer Applications, vol. 162, no. 2, 2017.

[7] M. S. Babu, N. Geethanjali, and B. Satyanarayana, “Clustering approach to stock market prediction,” International Journal of Advanced Networking and Applications, vol. 3, no. 4, p. 1281, 2012

[8] M. Usmani, S. H. Adil, K. Raza, and S. S. A. Ali, “Stock Market Prediction Using Encoder Decoder ConvLSTM,” in 2016 3rd International Conference on Computer and Information Sciences (ICCOINS). IEEE, 2016, pp. 322–327.

[9] B. G. Malkiel, A Random Walk Down Wall Street. W.W. Nortin and Company, 1999.

[10] A. E. Biondo, A. Pluchino, A. Rapisarda, and D. Helbing, “Are Random Trading Strategies More Successful than Technical Ones?,” PLoS One, vol. 8 (July), no. 7, pp. 1–14, 2013, doi: 10.1371/journal.pone.0068344.

[11] S. Chen and H. He, “Stock prediction using convolutional neural network,” IOP Conference Series: Materials Science and Engineering, vol. 435, no. 1, pp. 1–9, 2018.

[12] X. Ding, Y. Zhang, T. Liu, and J. Duan, “Deep learning for eventdriven stock prediction,” in Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, pp. 2327– 2333, 2015.

[13] R. Singh and S. Srivastava, “Stock prediction using deep learning,” Multimedia Tools and Applications, vol. 76, no. 18, pp. 18 569–18 584, 2017.

[14] Prediction with LSTM. Neural Computation, 2000, 12(10):2451.

[15] H. Jia, Investigation Into The Effectiveness Of Long Short Term Memory Networks For Stock Price Prediction. 2016.

[16] P. Oncharoen and P. Vateekul, “Deep Learning for Stock Market Prediction Using Event Embedding and Technical Indicators,” in 5th International Conference on

Advanced Informatics: Concepts Theory and Applications (ICAICTA 2018), 2018, pp. 19–24, doi: 10.1109/ICAICTA.2018.8541310
[17] Gu, H. Hsieh, C. Wu, R. Chang, and J. Ho, “A Fund Selection RoboAdvisor with Deep-learning Driven Market Prediction.,” in IEEE International Conference on Systems, Man and Cybernetics (SMC 2019), 2019, pp. 2845–2850.
[18] B. Yang, Z.-J. Gong, and W. Yang, “Stock market index prediction using deep neural network ensemble,” in 2017 36th Chinese Control Conference (CCC). IEEE, 2017, pp. 3882–3887.
[19] T. Gao and Y. Chai, “Improving stock closing price prediction using recurrent neural network and technical indicators,” Neural Computation, vol. 30, no. 10, pp. 2833–2854, 2018.
[20] Y. Hu, X. Sun, X. Nie, Y. Li, and L. Liu, “An enhanced LSTM for trend following of time series,” IEEE Access, vol. 7, pp. 34 020–34 030, 2019.
[21] Z. Y. Xu, L. N. Tang, and C. P. Tian, “Prediction of stock trend based on deep belief networks,” Applied Mechanics and Materials, vol. 644, pp. 5538–5541, 2014.
[22] F. Shen, J. Chao, and J. Zhao, “Forecasting exchange rate using deep belief networks and conjugate gradient method,” Neurocomputing, vol. 167, pp. 243–253, 2015.
[23] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, “Deep direct reinforcement learning for financial signal representation and trading,” IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 3, pp. 653–664, 2016.
[24] K. Zhang, G. Zhong, J. Dong, S. Wang, and Y. Wang, “Stock market prediction based on generative adversarial network,” Procedia computer science, vol. 147, pp. 400–406, 2019
[25] S.Haykin, 2nd ed. “Neural networks: a comprehensive foundation” Upper Saddle River, NJ: Prentice Hall, 1999.
[26] K. Hagan, M.T., Demuth, H.B., and Beale, M., Boston, MA:” Neural network design” PWS Publishing Company, 1996.
[27] Yahoo’s financial, “Financial Stock Prices,” 2020. https://finance.yahoo.com/ (accessed Sep. 10, 2020)
[28] Human in Machine world, https://medium.com/human-in-a-machineworld/mae-and-rmse-which-metric-is-better-e60ac3bde13d/ (accessed on Feb. 15, 2021)
[29] T. Gao and Y. Chai, & Y. Liu, (2017). Applying long short term momory neural networks for predicting stock closing price. 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS).
[30] T. Damrongsakmethee, V. Neagoe,(2020). Stock Market Prediction Using a Deep Learning Approach. 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)
[31] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta, “Singlelayer & multi-layer long short-term memory (lstm) model with inter-mediate variables for weather forecasting,” Procedia Computer Science,vol. 135, pp. 89–98, 2018.
[32] L. Di Persio and O. Honchar, “Artificial neural networks architectures for stock price prediction: Comparisons and applications,” International journal of circuits, systems and signal processing, vol. 10, no. 2016, pp.403–413, 2016.

[33] K. Cho, B. Van Merri{\"e}nboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.
[34] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," PloS one, vol. 14, no. 2, p. e0212320, April 2019.
[35] R. Bisoi, S. Ranjeeta, K. Dash. "A hybrid evolutionary dynamic neural network for stock market trend analysis and prediction using unscented Kalman filter." *Applied Soft Computing* 19 (2014): 41-56.
[36] P. Robert, and C. Hsinchun. "Textual analysis of stock market prediction using breaking financial news: the Azfin text system." *ACM Trans Inf Syst* 27, no. 2 (2009): 1139-1141.
[37] J. Clarke, T. Jonathan, J. Tomas, and M. Gershon. "The efficient markets hypothesis." *Expert financial planning: Advice from industry leaders* 7, no. 3/4 (2001): 126-141.
[38] M. Gorecki, T. Łuczak, (2013), ”Using derivatives in time series classification”, ´ Data Mining and Knowledge Discovery, Vol. 26, No. 2, pp. 310-331, Adam Mickiewicz University in Poznan (Poland), Koszalin University of Technology (Poland)
[39] M. Dixon, D. Matthew, K. Diego, "Classification-based financial markets prediction using deep neural networks." *Algorithmic Finance* 6, no. 3-4 (2017): 67-77.
[40] R. Schumaker, P. Robert, and C. Hsinchun. "Textual analysis of stock market prediction using breaking financial news: The AZFin text system." *ACM Transactions on Information Systems (TOIS)* 27, no. 2 (2009): 1-19.
[41] X. Ding, Z. Yue, L. Ting, and D. Junwen. "Deep learning for event-driven stock prediction." In *Twenty-fourth international joint conference on artificial intelligence*. 2015.
[42] G. Gehrig, S. Thomas, and M. Lukas. "Extended evidence on the use of technical analysis in foreign exchange." *International Journal of Finance & Economics* 11, no. 4 (2006): 327-338.
[43] W. Andrew, M. Harry, and W. Jiang. "Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation." *The journal of finance* 55, no. 4 (2000): 1705-1765.
[44] I. Mierswa, Z. Ingo. "Automatic feature extraction from large time series." In *Classification—the Ubiquitous Challenge*, pp. 600-607. Springer, Berlin, Heidelberg, 2005.
[45] F. Glorot, X. Yoshua, G. Bengio. "Understanding the difficulty of training deep feedforward neural networks." In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249-256. JMLR Workshop and Conference Proceedings, 2010.
[46] M. Velay and F. Daniel, “Stock chart pattern recognition with deep learning,” arXiv preprint arXiv:1808.00418, 2018.

[47] R. Dash and P. K. Dash, "A hybrid stock trading framework integrating technical analysis with machine learning techniques," The Journal of Finance and Data Science, vol. 2, no. 1, pp. 42–57, 2016.

[48] M. U. Gudelek, S. A. Boluk, and A. M. Ozbayoglu, "A deep learning based stock trading model with 2-d cnn trend detection," in 2017 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2017, pp. 1–8.

[49] T. Chen, L. Tai, and Y. Feng. 2016. An intelligent pattern recognition model for supporting investment decisions in stock market. Information Sciences 346: 261–74.

[50] P. Parracho, R. Paulo, N. Rui, and H. Nuno. 2010. Trading in Financial Markets Using Pattern Recognition Optimized by Genetic Algorithms. Paper presented at the 12th Annual Conference Companion on Genetic and Evolutionary Computation, Portland, OR, USA, July 7–11

[51] M. Velay, F. Marc, and D. Fabrice. 2018. Stock Chart Pattern recognition with Deep Learning. arXiv

[52] S. Nesbitt, V. Keith V., and Stephen Barrass. 2004. Finding trading patterns in stock market data. IEEE Computer Graphics and Applications 24: 45–55. [CrossRef] [PubMed]

[53] H. Lee, S. Mihai, M. Bill, and J. Dan. 2014. On the Importance of Text Analysis for Stock Price Prediction. Paper presented at the 9th International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26–31.

[54] C. Park, H. Cheol-Ho, and H. Scott. 2007. What do we know about the profitability of technical analysis? Journal of Economic Surveys 21: 786–826.

[55] M. Hiransha, E. Gopalakrishnan, K.M. Vijay, and K. Soman. 2018. NSE stock market prediction using deep-learning models. Procedia Computer Science 132: 1351–62

[56] T. Chen, L. Tai, and Y. Feng. 2016. An intelligent pattern recognition model for supporting investment decisions in stock market. Information Sciences 346: 261–74.

[57] B. Bernal, F. Armando, F. Sam, and P. Rohit. 2012. Financial Market Time Series Prediction with Recurrent Neural Networks. State College: Citeseer

[58] Edureka Blog. [Online] https://www.edureka.co/blog/classification-in-machine-learning/ Accesses on May, 2021

[59] Builtin Blog. [Online] https://builtin.com/data-science/regression-machine-learning Accessed on May, 2021

[60] Investopedia Blog [Online] https://www.investopedia.com/ask/answers/040215/what-does-sp-500-index-measure-and-how-it-calculated.asp Accessed on June, 2021

[61] Analyzing Alpha Blog. [Online] https://analyzingalpha.com/open-high-low-close-stocks Accessed on July, 2021

[62] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, ''Short-term residential load forecasting based on resident behaviour learning,'' IEEE Trans. Power Syst., vol. 33, no. 1, pp. 1087–1088, Jan. 2017.

[63] C. Li, Z. Ding, D. Zhao, & G. Zhang, (2017). Building energy consumption prediction: An extreme deep learning approach. Energies, 10(10), 1525.

[64] T. Kim, T.-Y.; Cho, S.-B. Predicting Residential Energy Consumption using CNN-LSTM Neural Networks. Energy 2019

[65] F.U.M. Ullah, A. Ullah, I.U. Haq, S. Rho, S.W.J.I.A. Baik,. Short-Term Prediction of Residential Power Energy Consumption via CNN and Multilayer Bi-directional LSTM Networks; IEEE: Piscataway, NJ, USA, 2019

[66] Z. A. Khan, T. Hussain, A. Ullah, S. Rho, M. Lee, & S. W. Baik, (2020). Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid CNN with a LSTM-AE based framework. Sensors, 20(5), 1399.

[67] Predicting the Household Power Consumption Using CNN-LSTM Hybrid Networks

[68] Improving Electric Energy Consumption Prediction Using CNN and Bi-LSTM

[69] D. C. Thomas, J. Prakash, K. Harigovind, & Sen, D. (2018, July). Lung Consolidation Detection through Analysis of Vocal Resonance Signals. In 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 957-960). IEEE.

[70] Radiopaedia. [Online], www.radiopaedia.org, accessed on December, 2017

[71] Toward Data Science Blog. [Online] https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346 accesses on August 2021