Forecasting Stock Closing Prices using Stacked LSTM Models to create Optimal Portfolios

Tajinderpal Toor
ttoor@torontomu.ca
Electrical and Computer Engineering Graduate Student
Toronto Metropolitan University, Toronto, Canada.

***Abstract*—The stock market is a vital part of the economy ,giving individuals the ability to invest in companies in order to help meet their financial goals. Due to numerous factors the market can be extremely volatile and can potentially result in a loss of funds. Therefore it is important to be prepared for fluctuations in the market by having an optimal portfolio. By calculating the performance of selected investments, the potential downside can be estimated. Through the use of Stacked Long Short Term Memory this paper proposes an architecture which will forecast prices and create an optimal portfolio. Stocks will be chosen from four ETFs through the use of two different stock screeners to create two different sets of potential stocks. The prices of these stocks will be forecasted and an optimal portfolio will be created based on the Sharpe Ratio. The portfolio expected returns are then compared to the SP 500.**

***Index Terms*—Long Short Term Memory Network , Time Series Data, Convolutional Neural Networks, Stock price Forecasting**

I. INTRODUCTION

The stock market is a vital part of the world economy. Through the stock market individuals can invest in companies with the hopes of achieving growth. By investing in the market, one can protect themselves against inflation and grow their money till their individual goals are met. Although investing in the market has numerous benefits the risk involved is a huge factor into why some individuals hesitate to invest their savings. Due to the ever shifting economy, political affairs and other factors, the markets tend to be very volatile and one's money can potentially be at risk. In order to protect oneself from the market, they must be educated on the performance of their chosen portfolio. Being able to predict the performance of a portfolio or chosen stock, will allow the customer to choose an investment that can produce maximal results within their risk tolerance. Throughout the years, there has been an increase in the use of deep learning to help predict the performance of companies. Recurrent neural networks, specifically the Long short term memory architecture has been used to process the sequential time series stock data to perform accurate forecasting of historical data. This paper proposes a hybrid method which uses two common architectures: the LSTM and CNN to perform accurate predictions on a pool of investments and create an optimal portfolio which is tested to estimate its potential earnings.

II. LITERATURE REVIEW

Designing and managing a portfolio can be considered an optimization problem where capital is being allocated to a set of assets. The goal of portfolio design is to allocate assets in a way where returns are maximized and risk is minimized. Authors in [1] Stock Portfolio Optimization Using a Deep Learning LSTM Model constructed an optimum portfolio for nine sectors of the Indian economy. Nine sectors were identified and the stocks of the companies that had the most significant impact to each sector were used to create portfolios. The nine sectors were: pharmaceuticals,infrastructure, realty, media, public sector banks, private sector banks, large cap, mid cap and small cap. This paper built a regression model using LSTM architecture to predict the price of each stock. The LSTM model used the past 50 days as the time series input, specifically the model took in daily close price as its input. The model in total had two LSTM layers, with a dropout of thirty percent to prevent overfitting. Through grid search it was found that the optimum epoch was 100 and the optimum batch size was 64.

Researchers at MIT ADT University in India proposed hybrid networks and a stacked LSTM model to predict stock prices. In [2], Stock Market Prediction and Portfolio Optimization researchers discussed using a Hybrid 1D-Convolutional Layer LSTM, GRU-LSTM model and a conventional stacked LSTM. The stock prices of SBI, Indian Bank and Bank of India are all predicted using these methods and compared to see which offers the best performance. The Stacked LSTM model gives good precision but is seen to be computationally expensive. The stacked LSTM is essentially just having multiple LSTM layers and a dense layer as an output. Stacking LSTM layers makes the model more complex and allows it to create more complex features and learn more complex patterns. This model takes in the input and passes it to a LSTM layer with 100 nodes, the output of this layer is fed to another LSTM layer with 50 nodes, which is then fed to another LSTM layer with 150 nodes and finally to an LSTM layer with 100 nodes. Next researchers discussed the Hybrid 1D-Convolutional Layer LSTM. The 1D convolutional layer extracts the most common and essential features. The input is passed through a 1d-convolutional layer with 32 filters then to another 1d convolutional layer with 64 filters then to a LSTM with 150 nodes then to a 1d-convolutional layer with 128 filters then finally through a LSTM with 100 nodes before being flattened and passed to a dense layer. Finally the GRU-LSTM model was introduced. In this model the input is passed to a gru with 256 nodes, then a dropout of 0.4 is applied, passed to LSTM with 256 nodes then to dropout of 0.4 then dense layers(64) then flattened before being passed to another dense layer.

In [3], researchers from various institutions and departments presented the paper, Harnessing a Hybrid CNN-LSTM Model for Portfolio Performance: A Case Study on Stock Selection and Optimization. This paper proposed a method which uses modern portfolio theory and deep learning to create an optimal portfolio. This paper introduced a CNN-LSTM + MV model which would forecast stock prices using a hybrid model and then allocate the stocks in an optimal portfolio using the mean variance model. Twenty-one stocks were selected from the National Stock Exchange of India and their historical data spanning from January 2005 to December 2021. This period features two market crashes, which will allow the hybrid model to extract and learn patterns from these downturns in the market. Convolutional Neural Networks can handle numerous channels of input data, the model can learn complicated patterns from many forms of input data. For example the input can be technical indicators, OHLC prices, or even market sentiment. OHLC prices are the daily open,high,low and close price of a stock. The output of the convolutional layer is passed to a max pooling layer before being flattened and fed to an LSTM to find long term dependencies. The twenty one stocks that were selected are fed to the hybrid model. From here the top K stocks with the highest returns will be selected for a portfolio. Once these stocks are identified, Markowitz's mean variance model is used to find the optimal capital allocation of the stocks. Researchers selected: Open,close,high,low,adjusted close,simple moving average,exponential moving average,relative strength index,rate of change,true range,average true range,momentum index,commodity channel index as the inputs which includes daily stock information and various technical indicators. The presented model had an average annualized cumulative return of 25.62

Researchers from the Department of Computer Engineering at the Istanbul Technical University presented [4], Portfolio Construction with Stock Prices Predicted by LSTM using Enhanced Features. This paper created a portfolio construction pipeline that predicts stock prices and constructs a portfolio based on these predictions. Usually LSTM models use raw time series data, but recent studies showed that features extracted from time series data can increase the capacity of the models. This paper proposes a model which takes in raw stock price data, such as opening price, closing price, highest price and lowest price to create candlestick charts. Through the use of an autoencoder, semantic information can be extracted from these images. The vector produced from this step is concatenated with the closing price to create a vector which is fed to the LSTM. The candlestick charts created had data from the previous twenty days. The LSTM forecasted the closing price at the end of the next day. These forecasted prices were used to calculate returns and make a portfolio featuring stocks with the highest returns. The portfolio created featured the highest performing stocks as per calculated from the proposed LSTM with enhanced feature model. This portfolio was compared to other portfolio strategies such as the 1/N method, Sharpe Ratio and LSTM with raw series and had the greatest daily mean return.

In [5], researchers presented Multivariate Regression Analysis for Stock Market price prediction using Stacked LSTM. This paper presented a model which uses a Stacked LSTM to forecast future stock prices at high accuracy. Researchers tested their model on data from the Dhaka Stock Exchange. This paper proposed a MultiVariate LSTM model which takes in multiple feature variables and shows how a deeper LSTM can handle the variability of the stock market better. The proposed model in this paper outperformed single layer LSTM which illustrated that a deep architecture is more robust to learn extreme variability. The model featured three hidden LSTM layers, where the output of one LSTM layer is the input of the next LSTM layer. Following this there is a dropout of 0.5 and a dense layer. The LSTM layers were fifty units each. The authors of this paper used Open, Close, High, Low as the inputs into the model. To show that the multivariate LSTM architecture performed better than the univariate, authors first used each of the features separately to perform a univariate model and calculated the RMSE. It was shown that as the number of features increases the RMSE value lowers. For example using close as the single feature input, the RMSE value is 58.57, when two input features (Close,Open) were used the RMSE becomes 57.32 and finally when all four features are used the RMSE becomes 52.41.

Researchers from Bangladesh, presented a paper [6], Predicting Stock Market Price: A Logical Strategy using Deep Learning which uses frequently used algorithms to create a prediction model that forecasts the price of a stock. This paper compared the performance of the Long Short Term, Extreme Gradient Boosting (XGBoost), Linear Regression, Moving Average and Last Value Model on twelve months of historical data for the Dhaka Stock Exchange. The historical data contained 236 data points between January 2019 to December 2019. The attributes researchers used to train their model are date, opening price, high, low,closing price and adjacent close. In this architecture the authors use two LSTM layers, where there is a dropout in between each layer and finally the output is passed through a dropout before being fed to the dense classifier. To evaluate the model, Mean Absolute Percentage Error is used where the lower the MAPE the better the result is. The LSTM model had a MAPE of 0.635, which outperformed all the other algorithms.

In [7], Intraday Stock Trading Strategy Based on Analysis Using Bidirectional Long Short-Term Memory Networks, researchers proposed a hybrid model with a Convolutional Neural Network and Bidirectional Long Short-Term Memory to forecast prices and generate stock trading signals. The model was used to predict the prices of 12 Stocks, AMD, APA, DVN, GOOGL, MOS, MRNA, NFLX, NVDA, OXY, SQQQ, TQQQ, and TSLA using technical indicators. This study took historical data from 2019 to 2022 and used the pening price, highest price, lowest price, closing price, and trading volume for each day to calculate the technical indicators. This data was split into training,validation and testing data where the training data was the first three weeks of the month, validation is the following week and the testing data is the year after the validation. The Bidirectional LSTM captures dependencies in both the forward and backward direction, the input sequence is processed in forward and backward direction with two separate hidden layers and outputs from both directions are combined. This paper investigated 4 different configurations: CNN-LSTM, LSTM-CNN ,CNN-BILSTM and BILSTM-CNN. The CNN-LSTM has an input layer, convolutional layer,dropout layer, LSTM layer, flatten layer, batch normalization, dense layer and output layer. The LSTM-CNN has an input layer,lstm layer,dropout layer,convolutional layer,flatten, batch normalization, dense layer and finally an output layer. The BILSTM-CNN has an input layer, BILSTM layer, dropout layer, convolutional layer, flatten layer, batch normalization and dense layer. Finally the CNN-BILSTM has an input layer , convolutional layer, dropout, BILSTM layer ,flatten,batch normalization, dense, and an output layer. The number of filters used in the cnn layer is : 32,64,128,256 and of nodes in lstm: 32,64,128,256 with a batch size of 32.

Wenjian Zheng from China authored a paper [8], Exchange-Traded Fund Price Prediction Based on the Deep Learning Model which investigated using a LSTM model and CNN-BiLSTM model to predict the next day price of an ETF. ETFs or exchange traded funds are a popular investment which allows customers to buy a pool of shares. ETFs are essentially a basket of shares and usually mimic the performance of stock indices. The models will predict the price of the next day and then from there it can be determined if the fund should be bought or not. This study used the opening price, highest, lowest,closing price, change and other technical indicators as features. The models were trained and evaluated on the Shanghai 50 exchange-traded securities investment fund where the daily trade data of 3500 trading days is used as data. The LSTM model has three LSTM layers, each with 128 units and a 0.2 dropout. On the other hand the CNN-BiLSTM–AM architecture had a convolutional layer with 256 filters, max pooling layer, 0.2 dropout, BILSTM with 256 units an attention mechanism and a dense layer. The Attention Mechanism is used to capture the effect of characteristic status on the highest price which increases prediction accuracy.

In paper [9], researchers from the Department of Computer Science and Engineering at the G.H Raisoni College of Engineering, present a method of forecasting stocks through the use of technical indicators and LSTM is used. The objective of this study was to enhance profitability and minimize the potential losses in trading. In this study, Algorithmic Trading Strategy Using Technical Indicators the first step is to collect the historical price of the NIFTY 50 index. This data included the open, high, low, and close prices for each trading day. This data is then used to calculate technical indicators such as supertrend, fibonacci pivot points, average directional index as features. Following this, the study defined the trading strategy. A buy signal is generated when the adx¿20 which represents a strong trend, price above supertrend line and the current price above fibonacci pivot point. On the other hand a sell signal is generated when adx¿20, price crosses below supertrend line and the current price is below fibonacci pivot point. The architecture used in this model had 3 LSTM layers and 1 dense layer. This strategy was used in options trading and generated a profit of 1.5 lakhs in options trading over a year and a half.

In paper [10], Tonghui Li from Tianjin University of Commerce, presented a method to predict the price of bitcoin using a LSTM model and multi-feature LSTM. Being able to accurately predict the price of bitcoin can reduce investors risk and eliminate trader concerns which may better facilitate the circulation of bitcoin in the market. This paper uses the daily open, high,low,close,volume and weighted prices of bitcoin between Jan 7th 2014 and October 17th 2017 as input to its model. Furthermore the model used would predict the weighted price of bitcoin using single features and multiple features. After training the single feature model produced an RSME of 121.333 while the multi feature model had an RMSE of 90.136.

III. Problem Statement

This paper aims to forecast the prices of high growth stocks. A pool of stocks will be chosen from commonly known Exchange traded funds and forecasted through the proposed model. From this pool the stocks will be placed in an optimal portfolio determined by Sharpe Ratio. The performance of this portfolio will be tested agaisnt the performance of the SP 500 year to date performance.

IV. System Model

Long short-term memory (LSTM) is a special kind of recurrent neural network (RNN) that deals with the vanishing gradient problem and excels at learning long term dependencies in sequences. A conventional LSTM unit consists of a cell, input gate, output gate and forget gate. The three gates are used to control the information flow (information stored, written or read) of the cell. The LSTM uses sigmoid and Tanh activation functions. The Tanh is a non linear activation function that regulates the values flowing through the network. It can take on values between -1 and 1. On the other hand, the sigmoid function keeps the value between 0 and 1. This helps the network to either forget or know which data to keep. If the result of multiplying by sigmoid is 0, the information is forgotten, but if the result is 1 the information stays 1. The forget gate will help decide which bits of the cell state are useful given the previous hidden state and new input data. This data is passed through a sigmoid function which will give a value between 0 and1. If the output is close to 0, the component of the input is irrelevant and vice versa if output is closer to 1. This result will then be multiplied with the previous cell state, which will tell us which components to ignore or have less influence. Essentially the forget gate helps decide which components should be given less weight.

The input gate decides what new information should be added (ie. updating the cell) to the cell state given the previous hidden state and new input data.

Using a tanh activation function allows the architecture to combine the previous hidden state and the new input data to generate a vector which tells the architecture how much to update each component of the cell given new data. This vector is called the ‘new memory update vector’ and has values between -1 and 1 where a negative value means the architecture needs to reduce the implant of a certain component. The input gate then helps decide which of the updates are worth keeping through the use of a sigmoid function. The output of the sigmoid is multiplied by the vector and added to the cell state, which will update only the selected components. Finally the output gate decides the new hidden state using the previous hidden state, new input data and newly updated cell state. Through the use of a sigmoid function only the necessary information is outputted. In summary the forget gate helps decide which information is unnecessary, the input gate helps to determine which components should be updated and by how much, finally the output gate helps determine which of the components need to be outputted and which don’t.

V. Data

The stocks selected for forecasting and portfolio creation were chosen from four ETFs (Exchange Traded Funds). An Exchange Traded Fund is essentially a basket of securities which typically tracks a particular index, sector or commodity. Another important aspect of ETFs is that they can be bought or sold on a stock exchange the same way regular stocks can. The four ETFs that were chosen are:

1) VUG - Vanguard Growth ETF
2) VIOG - Vanguard SP Small Cap 600 Growth ETF
3) ARKK - ARK Innovation ETF
4) ARKF - Fintech Innovation ETF

The Vanguard Growth ETF seeks to track the performance of the CRSP US Large Cap Growth index, and provides investors a way to match the performance of the largest growth stocks. This ETF features 221 stocks with its largest holdings being: Apple (Ticker: AAPL), Microsoft (Ticker: MSFT) and Amazon (Ticker: AMZN).

The Vanguard SNP Small-Cap 600 Growth ETF seeks to track the SNP Small-Cap 600 Growth Index, which is an index featuring stocks whose market capitalization fall between 300 million and 2 billion. Small cap growth stocks are generally more volatile but may offer higher returns making it a may more riskier or aggressive ETF. This ETF features 347 stocks with its largest holdings being Comfort Systems USA Inc (Ticker: Fix), Applied Industrial Technologies Inc (Ticker: AIT).

The ARK Innovation ETF invests in primarily companies that provide “disruptive innovation” or technologies that provide a product or service that changes the way the world works. This ETF invests in companies that relate to the following areas: DNA Technologies and the “Genomic Revolution”, Automation, Robotics, and Energy Storage, Artificial Intelligence and the “Next Generation Internet”,etc. This ETF typically holds 35-55 stocks where the largest holdings include Coinbase Global Inc (Ticker: COIN), ROKU Inc (Ticker: ROKU) and UiPath Inc (Ticker: PATH).

Finally the ARK Fintech Innovation ETF invests in primarily stocks that invest in companies related to Financial Technology (‘Fintech’). Fintech can be thought of as a technology product or service that may change the way the financial sector works. This ETF features companies related to the following business platforms: Transaction Innovations, Blockchain Technology, Risk Transformation, etc. This ETF typically holds 35-55 stocks where the largest holdings are Coinbase Global Inc (Ticker: COIN), Shopify Inc (Ticker: SHOP) and DraftKings Inc (Ticker: DKNG).

As you can see these ETFs feature a wide variety of sectors and risk levels giving us an ideal range of stocks to create portfolios from. The diverse selection of stocks will allow us to create portfolios where the holdings have low correlation, reducing the risk of the portfolio. Low correlation stocks can be used to migrate stocks as the performance of one stock is not necessarily the same as another (As one stock decreases the other may increase). In combination these ETFS provide a list of hundreds of stocks, from which only a select few are selected for portfolio creation. The methods in which these stocks were chosen and the concepts related are discussed in the following sections.

VI. Stock Selection

In order to understand the methodology used in screening and selecting the stocks, one must understand a few financial concepts. In order to pick the stocks that were used for forecasting and portfolio generation, two different Stock Screeners were used. A stock screener is a tool that allows investors and traders to sort through lists of individual stocks and choose ones that fit their own methodologies and search criteria.

Stock screeners can be used to separate or identify stocks based on their different fundamental and technical indicators. Fundamental Indicators can be found on the annual reports of stocks or various media properties that provide financial information,news and data. Financial indicators include: EPS(Earnings per Share), P/E (Price to Earnings Ratio), P/B (Price to Book Ratio).

The first stock screener used financial indicators and other attributes in order to identify potential stocks. These attributes include: Number of Institutional Holdings, Forward PE Ratio and annualized returns. The aim of this screener was to potentially identify stocks that would provide a higher return. The number of institutions holding a specific stock was used as an attribute as it shows interest in stocks from large parties. Large Institutions generally buy large amounts of stocks which can be a factor of price increase. Forward Price to Earning Ratio(P/E) represents the market’s optimism for a company’s prospective growth. A company with a higher forward P/E ratio than the industry indicates that the company is likely to experience growth.

In order to implement this screener the stocks from the four ETFs were combined into a single list. This list contained 598 stocks where certain stocks may be repeated due to overlapping holdings in the four ETFs. The entire list was then segmented into the different sectors present. For each sector, the mean of every attribute mentioned above was calculated. This mean would be used as a baseline, where attribute values higher than the mean may be selected. For each sector, the top 20 stocks that have the highest number of institutional holdings and annualized return greater than their specific means were chosen and placed in separate lists. For the Forward P/E, the values of each stock were divided by the sector mean, and the top twenty stocks with the highest PE greater than the mean were chosen. Out of these three lists the common stocks were chosen. If there were no common elements between the three lists, the common elements between the annualized returns and forward P/E ratio lists were chosen. Furthermore, if there were no common elements between these two lists, the annualized returns list was used to identify the top stocks from this sector. Finally if the number of stocks in a given sector was less than twenty, all stocks would be chosen. This process was repeated for every sector, and resulted in a final list of twenty three stocks that were identified.

The second stock screener was based on the Mark Minervini Trend Template. This screener finds stocks that are in an up-trend. This screener uses various technical analysis indicators to identify stocks that are in an uptrend or are approaching an uptrend.

Technical Analysis is a methodology used for analyzing and forecasting the direction of prices and market behavior through the study of past market data such as price. Furthermore, technical analysis is a trading discipline that identifies trading opportunities by analyzing statistical data. Technical analysis believes that prices trend directionally, meaning that the price of a stock will move up,down,sideways (flat) or a combination. Technical analysts also believe that history repeats itself, specifically the behavior of investors. Therefore they believe that predictable patterns can be recognized on charts. Technical indicators are pattern based signals produced by the price or volume of security. These indicators are used to predict future price movements.

The Mark Minervini Trend template used in Stock Screener 2 used eight conditions in order to make a decision. The criteria stocks are identified are: The current price of the security must be greater than the 150 and 200-day simple moving averages. The 150-day simple moving average must be greater than the 200-day simple moving average. The 200-day simple moving average must be trending up for at least 1 month. The 50-day simple moving average must be greater than the 150 simple moving average and the 200 simple moving average. The current price must be greater than the 50-day simple moving average. The current price must be at least 30The current price must be within 25The IBD RS-Rating must be greater than 70 (the higher, the better).

A trend is the overall direction of a market or asset price and can be identified by different technical indicators. Assets and markets can have either an uptrend or downtrend. On the other hand the Simple Moving Average is a technical indicator and is a moving average used to establish the direction the price of a stock is moving based on previous prices. Simple Moving Average can essentially be thought of as the stock's closing price over a period of time.

The RS rating is a metric of a stock's price performance over the last year compared to all other stocks and the overall market. The RS rating is generally a number between 1 and 99, where 1 is the worst RS rating.

The second stock screener took the original list of stocks and identified nine potential stocks. The stocks that were identified by this method are:

1) NU - Nu Holdings Ltd
2) NET - Cloudflare Inc
3) DKNG - DraftKings Inc
4) LNG - Cheniere Energy, Inc.
5) BRO - Brown & Brown, Inc.
6) UBER - Uber Technologies Inc
7) LRCX - Lam Research Corporation
8) INVA - Innoviva Inc
9) FBP - First Bancorp

VII. Features

The first architecture which will be discussed in detail below uses the daily Open, High, Low, Close and Volume of the stock over a set period of time. The second architecture introduces more features, specifically technical indicators. Using the daily close price the following technical indicators are calculated: Simple Moving Average (10,20,50 day period), Exponential Moving Average (10,20,50 day period), Double Exponential Moving Average (10,20,50 day period), Triple Exponential Moving Average (10,20,50 day period), Relative Strength Index, Average True Range, Kaufman Adaptive Moving Average, Momentum, Moving Average Convergence/Divergence and Bollinger Bands.

The exponential moving average (EMA) is a type of moving average that places greater weight and significance on the most recent data points. EMA reacts more significantly to price changes than the Simple Moving Average, mentioned above. Double Exponential Moving Average (DEMA) is a technical indicator used to reduce the lag in the results produced by traditional moving averages. The DEMA can filter "noise" or irrelevant market action which may affect results. The Triple Exponential Moving Average is an indicator designed to smooth price fluctuations which make it easier to identify price trends. The RSI or Relative Strength Index is a momentum indicator that measures the speed and magnitude of a stock's recent price change to evaluate if it is overbought or oversold. RSI is a value between zero and 100, where anything above 70 means the stock is overbought while under 30 means it is underbought. True Range is the maximum of these three conditions: Current high - current low, absolute value of current high minus previous close and absolute value of current low minus previous close. The Average True Range is then a moving average of the True Range. Kaufman Adaptive Moving Average (KAMA) is a moving average designed to account for market noise and volatility. KAMA will closely follow prices when the price swings are small and noise or volatility is low, but will adjust when the price swings are larger and the noise or volatility is higher. This indicator can identify overall trend and time turning points. Moving Average Convergence/Divergence (MACD) is a trend following momentum indicator that shows the relationship between two EMAs. The MACD line is calculated by subtracting the 26 period EMA from the 12 period EMA. By taking the nine period EMA of the MACD line, you get a signal line which is plotted on top of the MACD line. The Moving Average Convergence/Divergence can help traders identify the

Fig. 1. Methodology

strength of a directional move and the warning of a potential price reversal. Finally Bollinger Bands are plotted as two standard deviations both positive and negative away from a simple moving average. This indicator can also help identify overbought or oversold periods. An overbought stock is a stock that is typically overvalued and its price will fall as investors start selling. On the other hand an oversold stock is one that has been trading at a lower price and has the potential for a price increase.

VIII. Methodology/ Architecture

The image shown above represents the general methodology implemented in this paper. The initial list of stocks are processed through a stock screener, where potential stocks are identified based on different parameters. Once stocks are identified and fetched from the initial list, they are preprocessed inorder to be fed into the LSTM model. The first step in the preprocessing part is to standardize the data. The MinMaxScaler is used for this process to transform the features into a range of (0,1). Once the data is standardized, sequences are pulled out from the dataset inorder to be fed into the LSTM. For the models implemented in this paper, we are using data from the past 14 days in order to predict the closing price of the next day. Once the data is processed through the Stacked LSTM model, the output is used to extract the forecasted prices which is then used to create an optimal portfolio which aims to maximize return and minimize risk.

In order to minimize risk and maximize returns, we implement a portfolio optimization based on the Sharpe Ratio. The Sharpe Ratio is calculated by subtracting the Risk Free Rate from the Portfolio Return and then dividing by the standard deviation. The below steps were followed in order to implement the Sharpe Ratio Portfolio Optimization.

1) Calculate the Lognormal Returns from the forecasted close prices and annualize the returns
2) Set the Risk Free Rate. In general the risk free rate could be considered to be 2
3) Define the initial weights. Initially stocks in our chosen list are given equal weights.
4) Calculate the covariance matrix of all the stocks using annualized returns
5) Create a function that calculates the Portfolio risk using the portfolio standard deviation. This function will take in the weights and covariance matrix to calculate the standard deviation.
6) Create a function that calculates the portfolio's expected return using the weights and annualized returns
7) Using the metrics found in steps 5 and 6, create a function that will calculate the sharpe ratio

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_4 (LSTM)               (None, 14, 1000)          4024000

 lstm_5 (LSTM)               (None, 14, 1000)          8004000

 lstm_6 (LSTM)               (None, 14, 50)            210200

 lstm_7 (LSTM)               (None, 50)                20200

 dense_3 (Dense)             (None, 500)               25500

 dense_4 (Dense)             (None, 500)               250500

 dropout_1 (Dropout)         (None, 500)               0

 dense_5 (Dense)             (None, 1)                 501

=================================================================
Total params: 12534901 (47.82 MB)
Trainable params: 12534901 (47.82 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig. 2. Architecture 1

8) Set portfolio constraints(ie. Upper and lower bounds for weights). The constraints essentially will not allow a stock to be weighted less than zero or more than 0.5 (50
9) Create a function that will optimize the sharpe ratio(ie. Find the highest sharpe ratio) and return the optimum weights

As mentioned above, there were two separate architectures created, one for each of the stock screening methods. The architecture used in the first method is shown below.

The second architecture, which took input from the second stock screener, was also a stacked LSTM but was implemented using hyperparameter tuning. Using hyperparameter tuning the number of stacked LSTM layers,nodes in each LSTM and the dropout value were chosen to optimize accuracy and minimize loss. For the input LSTM, the number of nodes was chosen to be a value between 50 and 1000 nodes, with a step size of 50. Furthemore, the "hidden" LSTM layers were optimized to choose the ideal number of layers (1 to 4) and the same node tuning as mentioned above. Finally a learning rate scheduler was used to adjust the learning rate between epochs or iterations as the training progressed. Both architectures used an Adam Optimizer and Mean Squared Error as the loss function. The model for the second method can be seen below.

IX. Results

In order to train the model, 20 years of historical data was used for each stock. This specific time frame was chosen as there have been many significant market movements. Throughout the last twenty years we have had the financial crisis in 2007, COVID-19 and many more movements. These significant price movements are important data points as the models can pick up certain patterns which can be helpful in forecasting future prices.

For the first method, where Stock Screener 1 and the architecture shown in Figure xx was used, the results can be seen below.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 14, 500)           1056000

 lstm_1 (LSTM)               (None, 14, 950)           5513800

 lstm_2 (LSTM)               (None, 14, 50)            200200

 lstm_3 (LSTM)               (None, 14, 50)            20200

 lstm_4 (LSTM)               (None, 14, 50)            20200

 lstm_5 (LSTM)               (None, 6)                 1368

 dropout (Dropout)           (None, 6)                 0

 dense (Dense)               (None, 6)                 42

 dropout_1 (Dropout)         (None, 6)                 0

 dense_1 (Dense)             (None, 1)                 7

=================================================================
Total params: 6811817 (25.99 MB)
Trainable params: 6811817 (25.99 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig. 3. Architecture 2

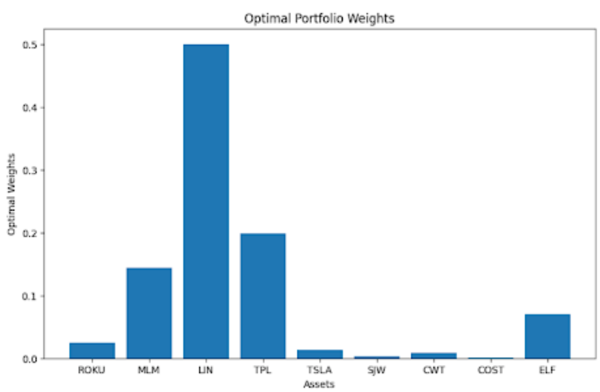
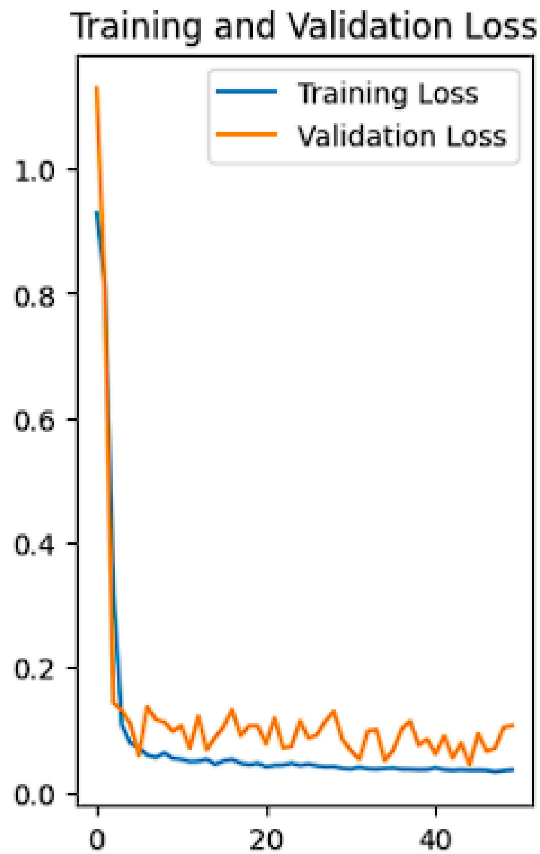


Fig. 4. Enter Caption

Using this approach, optimal portfolio created was as follows,:

1) SHOP - 0.0693
2) TPL - 0.2973
3) MELI - 0.2111
4) UTL - 0.2752
5) ELF - 0.1471

The portfolio created using the above Stocks would result in an expected annual return of 0.3896 or 38.96%. Typically the SNP 500 index is used as a benchmark for performance and over the last year has had a growth of 23.61%. Therefore the chosen portfolio has performed better than the SNP 500, by roughly 12.35%. In order to validate the results the method forecasted closing prices for the range of December 4th 2023 to March 2nd 2024. Using the stocks chosen in the optimal portfolio and their chosen weights, we created a portfolio for December 4th to December 15th 2023. Using real market closing prices, the portfolio had an expected annualized return of 0.4668 or 46.68%. The sharpe ratio calculated was 2.03 which is considered a very good sharpe ratio. The training



and validation loss plots for UTL and ELF can be seen below.

1) DKNG - 0.5
2) LRCX - 0.5

For the second method, where Stock Screener 2 and the architecture shown in Figure xx was used, the results can be seen below.The portfolio chosen for this method, consisted of only two stocks both equally weighted. The portfolio would have an expected annualized return of 0.1473 or 14.73%. Using real market closing prices for December 4th to December 15th 2023, an expected annualized return of 1.1587 or 115.87% with an expected volatility of 0.1981 or 19.81% was seen. This is well above the current year SNP 500 performance.

X. Future Work

LSTMs or long term short memory are powerful architectures for analyzing time series data and pattern recognition. Forecasting stock prices is a very challenging problem due to the uncertainty of the market and multitude of factors that affect stock prices. Instead, one can use the pattern recognition aspect of LSTMs to try and identify patterns that occur in the market. This can help to identify trends and buying or selling points of different stocks. By identifying buying and selling points, one can continuously update their portfolio, to ultimately try and achieve a return that beats the overall market.

Due to the numerous factors that affect market prices, more features can be added. Some other common features that can

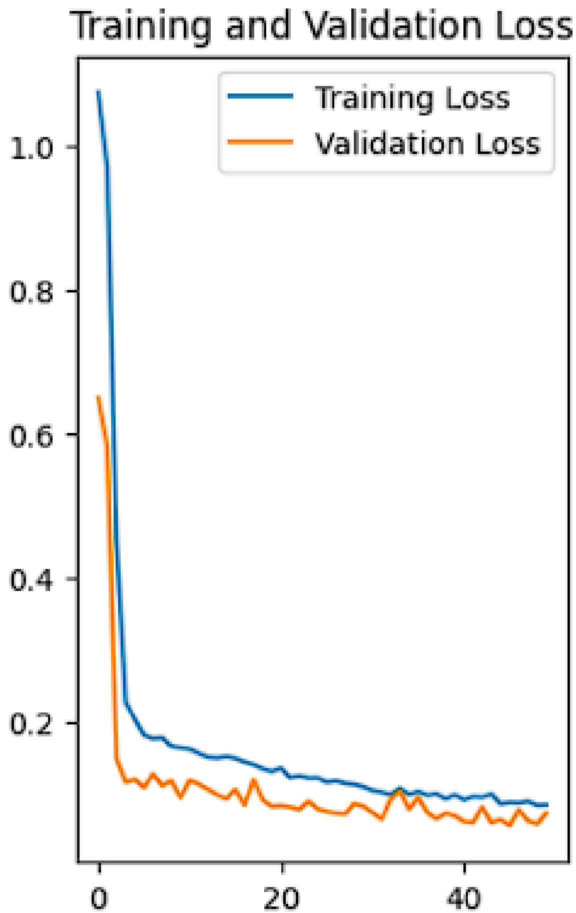
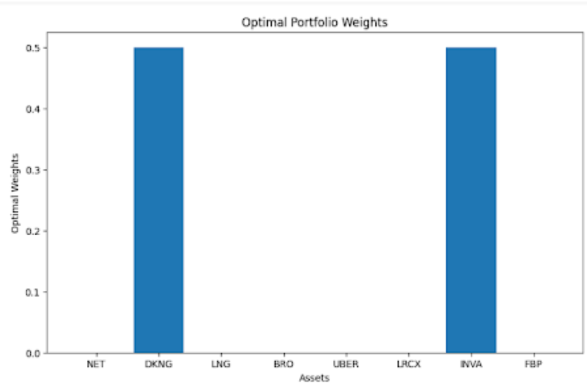


Fig. 5. Enter Caption



have a huge impact on prices are: current political status, twitter/social media sentiment , institution options positions and company news(ie. Layoffs, quarterly results, new contracts, etc. All these can have an adverse effect on prices and should be taken into account.

REFERENCES

[1] . Sen, A. Dutta and S. Mehtab, "Stock Portfolio Optimization Using a Deep Learning LSTM Model," 2021 IEEE Mysore Sub Section International Conference (MysuruCon), Hassan, India, 2021, pp. 263-271, doi: 10.1109/MysuruCon52639.2021.9641662.

[2] . Gondkar, J. Thukrul, R. Bang, S. Rakshe and S. Sarode, "Stock Market Prediction and Portfolio Optimization," 2021 2nd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2021, pp. 1-10, doi: 10.1109/GCAT52182.2021.9587659.

[3] . Singh, M. Jha, M. Sharaf, M. A. El-Meligy and T. R. Gadekallu, "Harnessing a Hybrid CNN-LSTM Model for Portfolio Performance: A Case Study on Stock Selection and Optimization," in IEEE Access, vol. 11, pp. 104000-104015, 2023, doi: 10.1109/ACCESS.2023.3317953.

[4] . L. Özbilen and Y. Yaslan, "Portfolio Construction with Stock Prices Predicted by LSTM using Enhanced Features," 2021 6th International Conference on Computer Science and Engineering (UBMK), Ankara, Turkey, 2021, pp. 639-643, doi: 10.1109/UBMK52708.2021.9558889.

[5] . Uddin, F. I. Alam, A. Das and S. Sharmin, "Multi-Variate Regression Analysis for Stock Market price prediction using Stacked LSTM," 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), Chittagong, Bangladesh, 2022, pp. 474-479, doi: 10.1109/ICISET54810.2022.9775911.

[6] . Biswas, A. Shome, M. A. Islam, A. J. Nova and S. Ahmed, "Predicting Stock Market Price: A Logical Strategy using Deep Learning," 2021 IEEE 11th IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE), Penang, Malaysia, 2021, pp. 218-223, doi: 10.1109/ISCAIE51753.2021.9431817.

[7] . Pholsri and P. Kantavat, "Intraday Stock Trading Strategy Based on Analysis Using Bidirectional Long Short-Term Memory Networks," 2023 6th International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 2023, pp. 572-578, doi: 10.1109/ICAIBD57115.2023.10206361.

[8] . Zheng, "Exchange-Traded Fund Price Prediction Based on the Deep Learning Model," 2021 China Automation Congress (CAC), Beijing, China, 2021, pp. 7429-7434, doi: 10.1109/CAC53003.2021.9727762.

[9] . Kumbhare, L. Kolhe, S. Dani, P. Fandade and D. Theng, "Algorithmic Trading Strategy Using Technical Indicators," 2023 11th International Conference on Emerging Trends in Engineering Technology - Signal and Information Processing (ICETET - SIP), Nagpur, India, 2023, pp. 1-6, doi: 10.1109/ICETET-SIP58143.2023.10151614.

[10] . Li, "Prediction of Bitcoin Price Based on LSTM," 2022 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE), Guangzhou, China, 2022, pp. 19-23, doi: 10.1109/MLISE57402.2022.00012.