ARTIFICIAL INTELLIGENCE
(CSE3013)

Topic: Intelligent Stock Trading

J COMPONENT

Vrishab V Srivatsa (18BCI0074)

Pratyush Kumar (19BCE0506)

Vishnu Shetty B (19BCE2116)

SLOT: C1+TC1

Submitted to

SANTHI K (SCOPE)

Objective:
The task of making an accurate prediction is an intricate as well as a highly difficult task. At the same time, the highly rewarding nature of stock prediction makes it an ideal challenge for AI Models. Our objective is to provide an AI-assisted platform with a highly accurate stock predicting system that combines multiple prediction models.

Abstract:
In recent times the stock market has proved to be an extremely volatile but high-rewarding field. The enormous amount of past data of each stock makes it an ideal environment to test and train our AI models. Our project aims to deliver an intelligent stock trading assistant which can give suggestions to the user regarding which stocks to buy and sell, so as to make the maximum profit. In order to maximize the profit, our AI Algorithms should give accurate predictions. In our project, we will be making use of a combination of multiple AI models to come up with an accurate prediction. This prediction will be displayed in the form of a simple percentage, thereby understood by a common man. By using multiple algorithms, the results provided by each will make up for the deficiencies of each other.

Literature Survey:

S. No.	Title of the Paper and year	Algorithms used	Dataset used	Performance measures	Gap identified	Scope for future work
1.	Machine Learning Approach in Stock Market Prediction (2017)	Support vector Machine (SVM) along with RBF Kernel Algorithm	Live data from Yahoo finance Url and stored data of 2014-2016	Graph is plotted showing prediction accuracy for four different feature lists. Results showed that the accuracy upto 89% is achieved	Feature list can be further expanded and improvised with different classifiers	Use of unsupervised preprocessor along with the supervise classifier
2.	A Machine Learning Model for Stock Market Prediction (2013)	Least Square Support Vector Machine (LS-SVM), Particle Swarm Optimization Algorithm	Datasets form Jan 2009 to Jan 2012 taken from Yahoo finance. All datasets are divided into training part (70%) and testing	Indicators such as relative strength index, money flow index, exponential moving average, stochastic	Better algorithms can be used	Implementation of various other algorithms to gain more accurate results.

			part (30%)	oscillator and moving average convergence/ divergence.		
3.	A Time Series Analysis-Based Stock Price Prediction Using Machine Learning and Deep Learning Models (2020)	Logistic Regression, K-Nearest neighbor, Decision tree, Bagging, Boosting, Random Forest, ANN, SVM, Multivariate Regression	Stock data of Godrej Consumer Products Ltd. using Metastock tool for collecting data on the short-term price movement of stocks.	Sensitivity, Specificity, PPV, NPV, CA, F1 Score for Classification methods. Correlation coefficient, RMSE/ Mean of Absolute Values of Actuals, Percentage of Mismatched Cases for Regression Models	Deep learning models have a much higher capability of extracting and learning the features from a time series data than their machine learning counterpart. However, in order to exploit the power of deep learning models, the volume of data should be very large	Use of generalized adversarial networks (GAN) in forecasting price movements and values. An integrated approach to building deep learning models that combines the power of LSTM, CNN and GAN.
4.	Stock Market Prediction Using Machine Learning (ML) Algorithms (2019)	Linear Regression, 3-month average, Exponential Smoothing, Time Series Forecasting ,	Data obtained from Yahoo finance for Amazon, Google and AAPL	Close Price, CMA(3), Forecasting	The stock prediction is only for a month and therefore cannot be used for long-term analysis	Various other Machine Learning Algorithms can be used, such as, LSTM and ARIMA
5.	Integrated Long-Term Stock Selection Models Based on Feature Selection and Machine Learning	SVM, Random Forest (RF) and ANN for Stock Price Prediction. SVM-RFE (recursive elimination feature),	The data set is from January 1, 2010 to January 1, 2018	Total Return, Annualized Return,Sharpe Ratio, Max Drawdown	No test in overseas markets such as the US and UK. The feature selection algorithm still needs to be optimized, such as how	Continually explore more new features which have more predictability

	Algorithms for China Stock Market (2020)	Feature selection based on RF for feature selection.			to determine the number of features selected.	
6.	Predict Stock Market Behavior: Role of Machine Learning Algorithms (2017)	Models: Six Sigma (SS) and Time-series based Mathematical models which include ARIMA and Exponential Smoothing	This research paper talks about the various models and ML Algorithms that can be used for Stock Market prediction and hence, there is no actual dataset used.	No actual calculations, instead, comparison different ML Algorithms.	No actual prediction or analysis is carried.	Design of the functional prototype, the design of EML algorithms, comparative analysis of EML algorithms on the common benchmark platform based on the certain technical indicators and performance evaluation
7.	Stock Market Prediction using Machine Learning (2017)	Different Regression models: Linear, Logistic, Polynomial, Stepwise, Ridge, Lasso and Elastic Net Different Classification Models: Support Vector Machine (SVM), Bayesian's Classifier and Decision Tree	Dataset is provided by Bombay Stock Exchange (BSE)	This paper summarizes the different ML models and Algorithms, and the tool for implementation. No actual calculation has been done. In general, results are given on the basis of accuracy and estimation.	No graphs or observation tables to support the conclusion.	Implementation of Linear and logistic regression to carry out stock market analysis and SBM for accurate results
8.	Stock Market	SVM and ANN	Data fetched	Adjacent Open,	Outputs and results can be	Implementation of other Machine

	Prediction Using Machine Learning (2020)		from quandl	Adjacent High, Adjacent Low, Adjacent Close, Adjacent Volume, HL_PCT, PCT_Change,	given in graph or tabular form for better comparison.	learning models and extending the time period of analysis.
9.	Stock Market Analysis using Supervised Machine Learning (2019)	Linear Regression	Dataset provided by quandl.com (WIKI/GOOGL)	Adjacent Open, Adjacent High, Adjacent Low, Adjacent Close, Adjacent Volume, HL_PCT, PCT_Change.	Paper is limited to only supervised machine learning, and tries to explain only the fundamentals of this complex process.	Shifting to SVM, trying and testing different models, looking for new and improved features, changing the entire data model to suit the model entirely etc.
10.	Stock market prediction using machine learning techniques (2016)	SLP, MLP, RBF, SVM	NEWS and twitter data was available in the form of feed which was processed using text mining techniques. The library OpinionFinder was used for this purpose.	Covariance, Correlation	Lack of resources and unavailability of data for the market.	Implementing the used ML Algorithms on various other countries' stock exchanges and for an extended time-period.

S.No.	**Title of the Paper And year**	**Algorithm s used**	**Data set being used**	**Performance measures**	**Gap identified**	**Scope for future work**

11	Forecasting stock prices with long-short term memory neural network based on attention mechanism(2019)	Long short-term memory neural networks (LSTM)	Stock indices(stock price and volume)	Mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R2). The smaller the MSE, RMSE, and MAE, the closer the predicted value to the true value; the closer the coefficient R2to 1, the better the fit of the model.	Due to the nonstationary, nonlinear, high-noise characteristics of financial time series,[7] traditional statistical models have difficulty predicting them with high precision. A time-weighted function was added to an LSTM neural network, and the results surpassed those of other models.	Our work has found that an attention-based LSTM has more predictive outcomes for price prediction than other methods. However, simply considering the impact of historical data on price trends is too singular and may not be able to fully and accurately forecast the price on a given day. Therefore, we can add data predictions related to stock-related news and basic information, so as to enhance the stability and accuracy of the model in the case of a major event.
12	An innovative neural network approach for stock market prediction (2018)	Deep long short-term memory neural network (LSTM) with embedded layer	Stock-related data tables	Accuracy between trained data and tested data	Traditional neural network algorithms may incorrectly predict the stock market, since the initial weight of the random selection problem can be easily prone to incorrect predictions	The text information in the stock market such as news is not fully utilized. So in the next step, we will consider adding text information factor to the model to further improve the performance.
13	Adaptive Stock Trading Strategies with Deep Reinforcement Learning Method (2020)	Reinforcement learning	Price open, close, high, low and volume of stocks	Technical indicator	Previous studies did not fully exploit the properties of financial data and had unstable returns in a volatile market. The financial data contains a large amount of noise, jump, and movement leading to the highly non stationary results	They haven't included the idea of portfolio management. Investment in a single stock has a limited profit margin and comparatively high risk. A good quantitative stock trading strategy needs to build portfolios of multiple stocks. Hence, our future work will focus on utilizing deep reinforcement learning methods in the portfolio management

14	Applying Long Short Term Memory Neural Networks for Predicting Stock Closing Price (2018)	Recurrent Neural Networks (RNN),Long short-term memory neural networks (LSTM)	Stock market historical trading data	Mean absolute error (MAE),root mean square error (RMSE), mean absolute percentage error (MAPE) , average mean absolute percentage error (AMAPE)	Mining stock market trend is generally considered to be a both interesting and challenging task due to its uncertainty, nonlinear, dynamic, nonparametric, inherent noisy environment and nonlinear characteristics.	To achieve higher accuracy by using multiple ai models
15	A graph-based convolutional neural network stock price prediction with leading indicators (2020)	Stock sequence array convolutional neural network. (SSACNN)	Historical data of prices and two leading indexes, futures and options of stock	Plotting accuracy graphs and comparing results with other algorithms	Making an accurate prediction becomes a highly difficult task. A new convolutional novel neural network that can improve the prediction accuracy	The LSTM model might have more potential to get better performance than the CNN model.
16	A New Approach to Neural Network Based Stock Trading Strategy(2019)	Multilayer Perceptrons Neural Network(MLP NN)	Price open, close, high, low and volume of stocks	Technical indicator,as well as the profit earned.	In our approach we try to take into account the best features of all the three sources (machine learning, technical analysis and human factor) together to build a robust trading system.	Even the best model is only as good as the data it is trained on. In stock price prediction selection of the right input variables is a much more difficult problem than building and training the predictive model itself.
17	A Deep Neural-Network Based Stock Trading System Based on Evolutionary Optimized Technical Analysis Parameters	Multilayer Perceptron (MLP)	Stock price between a given period	Moving Average Convergence and Divergence (MACD), Relative Strength Index (RSI), , Simple Moving Average (SMA)	Combining evolutionary optimized technical analysis indicators as features for a neural network based stock trading model has not been studied extensively.	We plan on combining more technical parameters and utilize Convolutional Neural Networks (CNN) or other deep neural network models.

	(2017)					
18	An integrated framework of genetic network programming and multi-layer perceptron neural network for prediction of daily stock return (2019)	GNP (Genetic network programming)	Daily Final price, the maximum price, minimum price, and trading Volume	Technical analysis indicators	To increase the accuracy of prediction	Using a combination of technical indicators with more effective configuration
19	An Empirical Study on Importance of Modeling Parameters and Trading Volume-Based Features in Daily Stock Trading Using Neural Networks(2018)	Multilayer Perceptron (MLP)	Price and volume information	Accuracy of prediction and the trading profit.	Studies attempted to extract input features mostly from the price information with little focus on the trading volume information. We generated input variables by considering both price and volume information with even weight.	Investigation of more complicated indicators that can be derived from either price of volume data. In addition, we will apply our method to the prediction of other financial markets such as interest rate, exchange rate and cryptocurrency .
20	A Comparative Study of A Recurrent Neural Network and Support Vector Machine for Predicting Price Movements of Stocks of Different Volatilities (2019)	Long short-term memory (LSTM) and support vector machines (SVM)	Basic stock trading information, i.e. daily open, close, high, low prices, volume, amount and daily change	Accuracy between trained data and tested data	Existing systems focus on improving the ai modesl.In this paper various data pre-processing techniques including the principal component analysis are utilized to enhance their overall performance.	LSTM has proven to be more accurate than SVM.Scope for future work would be o improve upon the existing LSTM model

S.No	**Title of the Paper and year**	**Algorithms Used**	**Data set being used**	**Performances Measure**	**Gap Identified**	**Scope for future work**

21	The application research of Neural Network and BP Algorithm in Stock Price Pattern Classification and Prediction (2020)	BP algorithm neural network	Test data (stock price data for 5 consecutive days)	Accuracy of the stock price under the prediction of deep learning fuzzy algorithm and under the prediction of the BP algorithm neural network	We can often see that there is a big deviation between the stock price and the price, which is mainly affected by the relationship between supply and demand.	In the next step, research of more stock technical indicators are improved and find out the relationship between indicators and prediction accuracy. Further, improve the prediction accuracy of the law. In the process of forecasting the stock price with the help of a neural network system, this study is only in an ideal state. It does not consider other external factors such as economic development momentum, government policy factors, other emergencies and so on. In fact, in a certain period, these external factors have a great impact on the stock price
22	A Q-learning agent for automated trading in equity stock markets (2020)	Q-learning algorithm of Reinforcement Learning to find optimal dynamic trading strategies	real stock market data from the Indian and American stock markets.	Average Annual Return (%) Accumulated Return Average Daily Return Maximum Drawdown (%) Standard Deviation (%) Sharpe Ratio (%)	This study is based on the Indian and the American Equity stock market, which permits both long buy and short sell trading using the Securities Lending & Borrowing (SLB) mechanism. The experiment performed on daily data.	In future, these models can use for other frequency datasets such as hourly dataset.

23	Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction (2018)	a hybrid approach integrating long short-term memory (LSTM) network and genetic algorithm (GA)	Korea Stock Price Index (KOSPI) data.	The derived result of the GA-optimized LSTM network is measured by computing the mean squared error (MSE), mean absolute error (MAE), and the mean absolute percentage error (MAPE) of the actual closing price of stock market, and the output of the proposed hybrid model. One form of statistical verification, called t-test has been conducted to investigate whether GA–LSTM outperforms the benchmark significantly	First, we did not take into consideration the trading commission in the analysis, and only forecasted the value of stock index and prices. However, in real-world investment environments, it is necessary to consider the trading commissions for higher returns, which can be a good topic for further discussion. Second, this study is conducted using only Korean stock market data. Therefore, further research can include data from various stock markets. Third, as mentioned earlier, the decision output of LSTM network is not easy to comprehend	To prevent these problems, learning parameters of the neural network should be properly selected, thus, further research can be conducted to optimize various other hyperparameters in LSTM networks. In addition, when it also comes to setting control parameters of GA, like the crossover rate and mutation rate, many suitable combinations can be derived that can improve the performance of research

24	Pattern Matching Trading System Based on the Dynamic Time Warping Algorithm (2018)	a pattern matching trading system (PMTS) based on a dynamic time warping algorithm that recognizes patterns of market data movement in the morning and determines the afternoon's clearing strategy	uses the KOSPI 200 index futures time series data.	A self-developed program was used for the analysis in Phase 2 with daily 10-min time series data. For pattern matching of daily market data by the dynamic time warping algorithm, two sets of 27 fixed patterns and 13 fixed patterns are used as input data	When financial market investors make less efficient investment strategies with the PMTS, the financial markets are less likely to be efficient.	A future study can be enriched by the studies presented in this paper. An interesting extension to the current study would include empirical studies using a more sophisticated DWP algorithm, such as the deepening dynamic time warping (DDTW) algorithm or the segmented dynamic time warping (SDTW) algorithm or the cluster generative statistical dynamic time warping (CSDTW) algorithm, from which better results a

25	Predict Stock Market Behavior: Role of Machine Learning Algorithms (2019)	ML algorithms (LR, LogR, DT etc)	Data from the NYSE realtime	Time series-based mathematical models: In this model the correlation between datasets can be used with the help of statistical tools. Time series models are built to study how a time series moves over time and application of volatility forecasting models to predict changes in volatility	The problem here is that prediction of price movement is not enough for successful application in trading. ML prediction model tells us the move will be +1% or −1% tomorrow. Because this 1% might mean −10% with some probability or +20% with some other much smaller probability. SPX is always going up almost all the time for long-term investments, but every 1% of precision improvement in prediction is very difficult to predict	The analysis part can be used to design EML which is a correct combination of all predictive models in order to make a good decision better. This forms the basis for subsequent research. Further, research stages would be the design of the functional prototype, the design of EML algorithms, comparative analysis of EML algorithms on the common benchmark platform based on the certain technical indicators and performance evaluation, which can be done in near future.
26	Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms (2019)	ANN,SVM,RF, DL, LSTM	The data used in this study include the close price of daily data of iShares MSCI United Kingdom from January 2015 to June 2018.	The closing price of iShares MSCI United Kingdom was analysed from January 2015 to June 2018 using four methods. Based on the Jarque–Bera test, these data do not have a normal distribution	Due to the use of time series data in this study, the role of other influential factors on stock price prediction were not investigated;	Researchers are recommended to consider the role of other factors in future studies and compare the results obtained with the results of this study.

27	Predicting the Unpredictable: An Application of Machine Learning Algorithms in Indian Stock Market (2019)	time series analysis and machine learning algorithms such as the artificial neural network (ANN)	Various approaches are compared based on methodologies, datasets, and efficiency with the help of visualisation.	We study fundamental and technical analysis and compare them to get the results. In technical analysis, analysis starts with stock charts, as all relevant information is included in stock prices In fundamental analysis, analysis starts with the company's financial statements like income, balance sheet, and cash flow statement	A comparative analysis of various different algorithms used for predicting future stock market prices and found that Long short Memory Neural network (LSTM NN) producing better results as compared to other techniques.	In the future, this research would help to improve the efficiency and accuracy of prediction. Deep learning classifiers would also be analysed in the future to predict and gain the maximum benefit from the stock market.

28	Automated trading systems statistical and machine learning methods and hardware implementation: a survey (2018)	trading systems built using various methods and empirically evaluate the methods by grouping them into three types: technical analyses, textual analyses and high-frequency trading.	These raw data may include fine-grained data that contain extremely detailed information about orders being handled by the exchange (Kearns and Nevmyvaka 2013), or these data may include less-detailed data, such as opening, closing, highest, and lowest prices and transaction volumes over a certain time window.	studies of technical analysis-based trading systems, and we consider classical methods based on recent findings and highlight the application of machine learning algorithms. studies of fundamental analysis-based trading systems, and we emphasize the systems that focus on textual analysis since most of the influential news information is transmitted in the form of a text data stream.	As financial market is heavily correlated with various sources of information, a research direction lies on the system fusion for multi-information processing. An integrated EIS is aimed at fulfilling the intelligent decision-making with as much as possible useful information taken into consideration. In order to meet the real-time trade demand, dedicated hardware is to be implemented to ensure accelerated computation and secure communication in a real-time manner.	In future, for the purpose of data safety, techniques such as network encryption security should therefore be further explored and exploited.

29	Evaluation of Pattern Based Customized Approach for Stock Market Trend Prediction With Big Data and Machine Learning Techniques (2019)	There are two common methods to predict stock market prices, first one is technical analysis and the second one is fundamental value analysis. Two models are introduced as part of the research. First is daily prediction model considers both sentiment and historical data which forecasts the trend for next day. The second model is monthly prediction model considers only historical data (Nayak et al., 2016).	This article learns the model from Indian National Stock Exchange (NSE) data obtained from Yahoo API to forecast stock prices and targets to make a profit over time.	First is for Next-Day model in which 50% accuracy was obtained and second was a Long-term model in which 79% accuracy was obtained overall. Logistic Regression, SVM, Gaussian Discriminant Analysis, Quadratic Discriminant Analysis were applied and 70% of the data was used as a training data and remaining data was testing data. Overall, the highest accuracy was obtained by SVM.	it can be easily seen that fewer, more appropriate and stable indicators are obtained, and they produce better prediction accuracy than the full feature space (Pehliyanit et al., 2016). By considering various patterns like continuous up/down, volume traded per day and also including sentiment of the company a model has been built and tested with different stock market data available open source as shown in (Nayak et al., 2016).	It is concluded that stock technical indicators are very effective and efficient features without any sentiment data in predicting short-term stock trend as per trend Deterministic Data Preparation Layer proposed by (Patel et al., 2015) paper exploits inherent opinion of each of the technical indicators About stock price movement.

30	Dynamic Weighting Multi Factor Stock Selection Strategy Based on XGboost Machine Learning Algorithm (2018)	A dynamic weighting multi-factor stock selection strategy based on XGBoost model is constructed. XGboost machine learning method is used to predict the IC coefficients of factors	The input features of the XGBoost model are historical IC coefficients of factors and market variables. In this paper, the alternative stock pools are constituent stocks of CSI 300 Index and CSI 500 Index. The in-sample training data is the historical data from January 2007 to December 2013, and the out-of-sample test data is the market data from January 2014 to August 2018.	Representative factors are selected in seven categories: profitability, quality, size, growth, liquidity, valuation, momentum and reversal. Illiquidity factor is average value of the absolute value of the daily price change divided by the amount of the trading during a period of time. ILLIQ reflects the volatility of the securities price under the unit turnover. If the ILLIQ is small, the impact of securities trading on the price is small, and the liquidity of stock is good. On the contrary, if the ILLIQ is large, the liquidity is worse.	According to the predicted IC coefficients, the weights of factors are adjusted dynamically. A higher weight will be given to the effective style factor, and the invalid factor will be given a lower weight.	The empirical results prove that XGBoost model is effective in predicting IC coefficients and the dynamic weighting based on XGBoost model can improve the performance of multi-factor stock selection strategy.

Existing Systems:

The accuracy of most existing systems is around 60 percent. The existing systems that are used to predict and forecast the stock market rely solely on one Machine-Learning Algorithm. The same algorithm is used for short-term and long-term prediction. The existing systems only cater to those who have good in-depth knowledge about the working of the stock market. They display the predictions in terms of technical indicators, which are not understood by a common man. Most of the existing systems are paid.

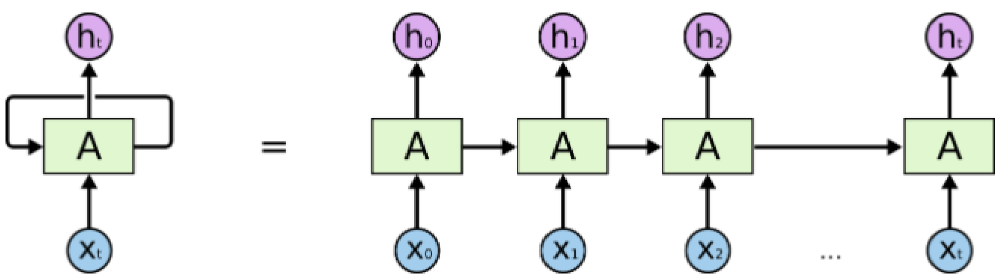
<u>Gap identified</u>:

1. The accuracy of most existing systems is around 60 percent. Our project aims to increase this accuracy by using an aggregation of multiple algorithms.
2. Relying on a single algorithm can give wrong results depending on the scenario. This occurs as each algorithm has its own pros and cons. Our project relies on multiple algorithms, thereby making up for the deficiencies of each other.
3. Most of the existing works have used the same algorithm for all scenarios (both long-term and short-term). This is a big flaw as short-term prediction requires you to see the recent trends, and long-term prediction requires you to see the past trends. Thus our project will be using different algorithms for long-term and short-term.
4. Existing systems cater only to those who have a good knowledge about the working of the stock market, by using technical indicators. Our project can be used by anyone as it gives a simple percentage of success of a stock, which can be understood by anyone, thereby being very user-friendly.
5. The need for an easy to use, free and AI assisted stock trading platform to increase successful trades, in both long-term and short-term perspectives.

<u>Analysis of models</u>:

In our project, we will be using multiple AI models, so that the predictions are more accurate and less reliant on one single algorithm. We will be using different algorithms depending on whether the customer would prefer making long-term or short-term investments.

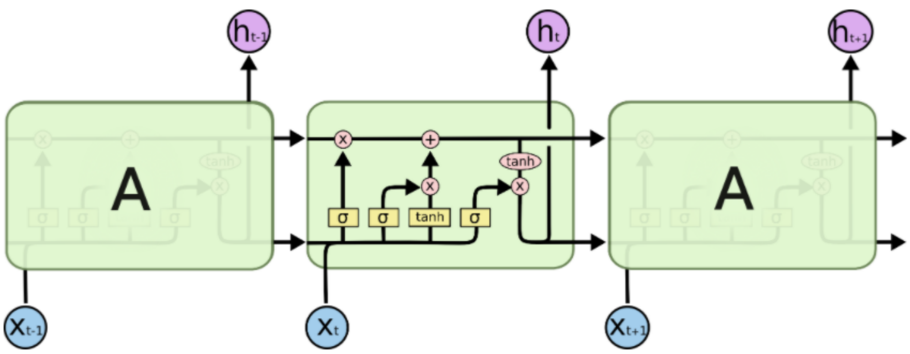
1. **LSTM:**
   Neural network: information processing paradigm inspired by biological nervous systems, such as our brain.
   Humans don’t start their thinking from scratch every second.Your thoughts have persistence.Traditional neural networks can’t do this.
   Recurrent neural networks address this issue. They are networks with loops in them, allowing information to persist.



An unrolled recurrent neural network.

One of the appeals of RNNs is the idea that they might be able to connect previous information to the present task.But although RNNs are able to use past information,they are incapable of handling “long-term dependencies.”RNN is very useful for short term dependencies.

Long Short Term Memory networks:
LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is their default behavior.
LSTMs also have this chain-like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.



The repeating module in an LSTM contains four interacting layers.

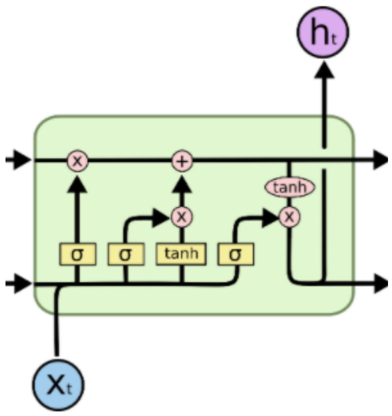
The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates.Thus we are able to select the information that is required for the particular scenario.

The first step in our LSTM is to decide what information we’re going to throw away from the cell state. This decision is made by a sigmoid layer called the “forget gate layer.” It looks at ht−1 and xt, and outputs a number between 0 and 1 for each number in the cell state Ct−1. A 1 represents “completely keep this” while a 0 represents “completely get rid of this.”

The next step is to decide what new information we’re going to store in the cell state. This has two parts. First, a sigmoid layer called the “input gate layer” decides which values we’ll update. Next, a tanh layer creates a vector of new candidate values, C~t, that could be added to the state. In the next step, we’ll combine these two to create an update to the state

We multiply the old state by ft, forgetting the things we decided to forget earlier. Then we add it*C~t. This is the new candidate values, scaled by how much we decided to update each state value.

Finally, we need to decide what we’re going to output. This output will be based on our cell state, but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state we’re going to output. Then, we put the cell state through tanh (to push the values to be between −1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

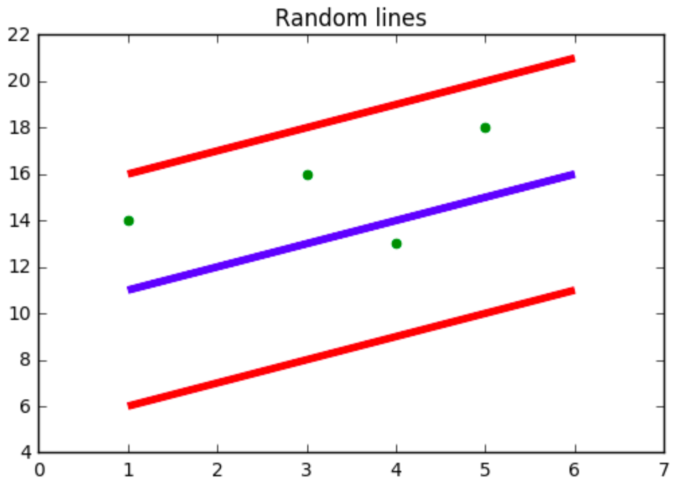


2. **SVR:**

Support vector Regression (SVR) is the combination of a Support Vector Machine and regression. SVR is used for working with continuous values instead of Classification which is used in SVMs.

Key terms used in the implementation of SVR:

a. **Kernel**: The function used to map a lower dimensional data into a higher dimensional data.
a. **Hyper Plane**: In SVM this is basically the separation line between the data classes. Although in SVR we are going to define it as the line that will will help us predict the continuous value or target value
b. **Boundary line**: In SVM there are two lines other than Hyper Plane which creates a margin . The support vectors can be on the Boundary lines or outside it. This boundary line separates the two classes. In SVR the concept is the same.
c. **Support vectors**: These are the data points which are closest to the boundary. The distance of the points is minimum or least.



blue line->hyperplane, red lines-> boundary lines

Let's suppose the red lines are at a distance +epsilon and -epsilon (+e and -e for ease).

Equation of hyperplane is wx+b=f(x)

where w->weight vector, c->bias, f(x)->SVR function

Equations of the boundary lines are:

wx+b=(+)e + y

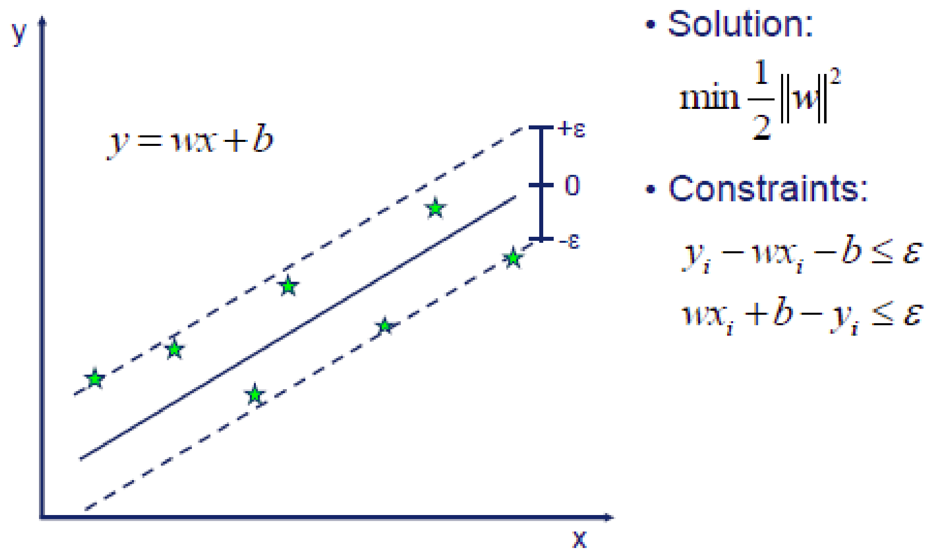
wx+b=(-)e + y

which gives us the constraints

y-wx-b≲e

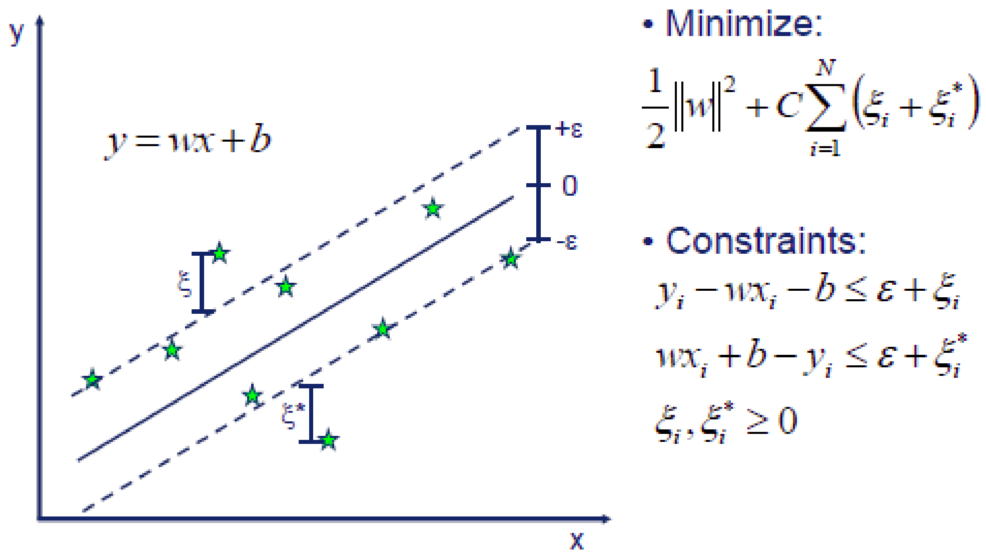
wx+b-y≲e

Here, y is the target value



This is possible only when |f(x)-y|≲e

But, when the training data is not linearly separable, slack variables ξ+ and ξ-.



Thus the decision boundary is our Margin of tolerance that is We are going to take only those points who are within this boundary.

Or in simple terms that we are going to take only those points which have the least error rate. Thus giving us a better fitting model.

Proposed Methodology:

In the scenario of stocks, trends of the stocks can repeat but the stock price may never repeat itself. In our algorithm, rather than predicting the stock price itself based on past values, we take into consideration the stock trends of the past and use it to predict the future. This stock trend is obtained by using the delta function. We take the delta of everyday and input it into our data-set. Therefore, the output given by our algorithm is also in terms of delta. To obtain the final result we add this delta value to the existing stock price of the previous day.

We have found this method to be far more superior than taking previous stock values as input to predict future values. The reason being, the future stock values depend only on the trends and not the past value itself. With this innovation we were able to increase the accuracy of our prediction significantly.

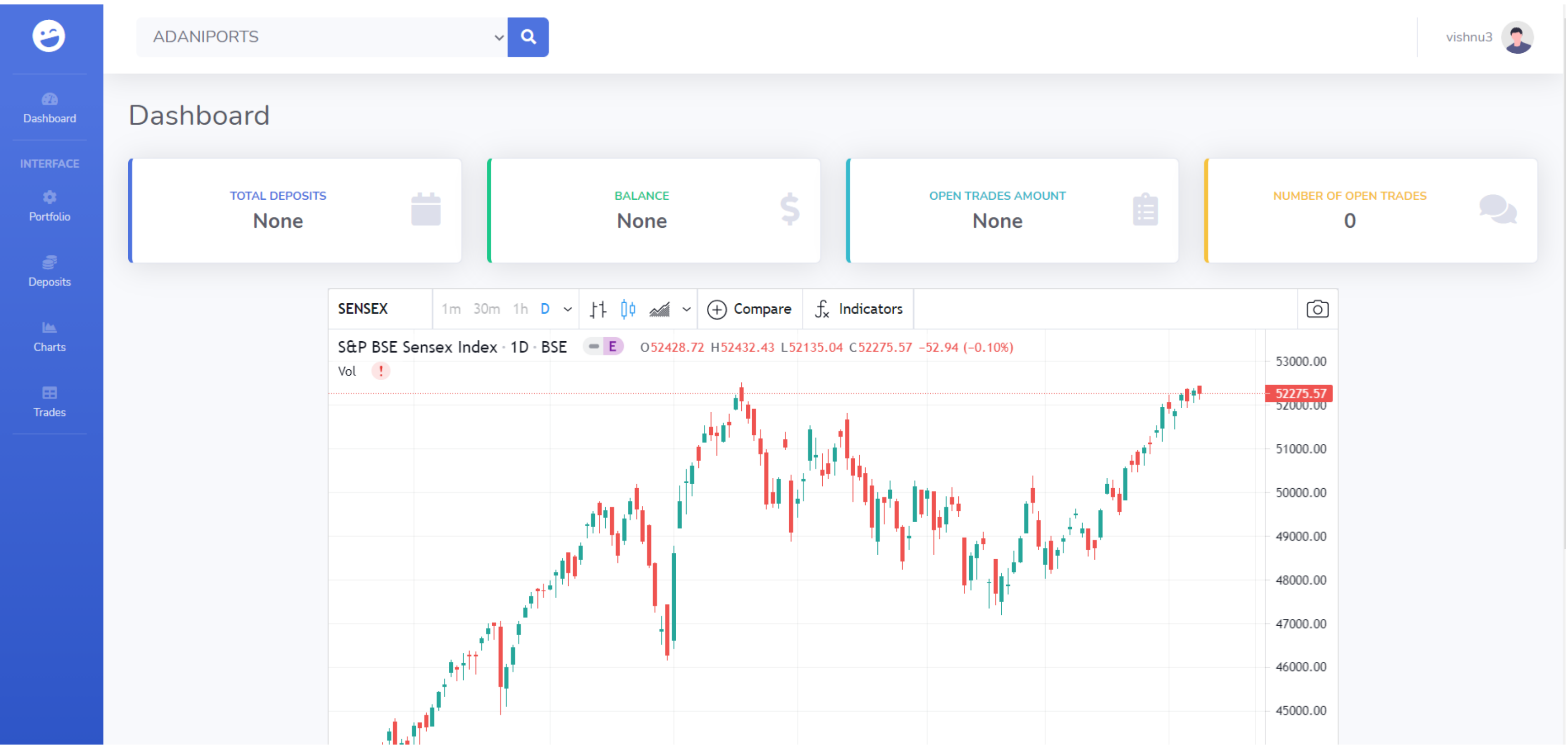
```
for i in range(len(stk_data)):
    if(i>0):
        stk_data['delta'][i] = (stk_data['VWAP'][i] - stk_data['VWAP'][i-1])/stk_data['VWAP'][i-1]
    else:
        stk_data['delta'][i] = 0
```

In our project, after taking the delta as inputs, we train two different models: SVR and LSTM. By training two different models, we get results based on two different approaches which further enhances the likeliness of an outcome. The drawbacks of one model are made up by the other. Thus, we don't rely heavily on a single model, which increases the safety of our predictions.

The results of multiple researchers show that nonlinear systems performed better than linear ones, and two-model systems performed better than single-model ones. Generally, models that relied on predictions from more than one algorithm had better accuracy in predicting the future stock prices. Hence, we have come up with a two-model prediction system for predicting stock price movements.

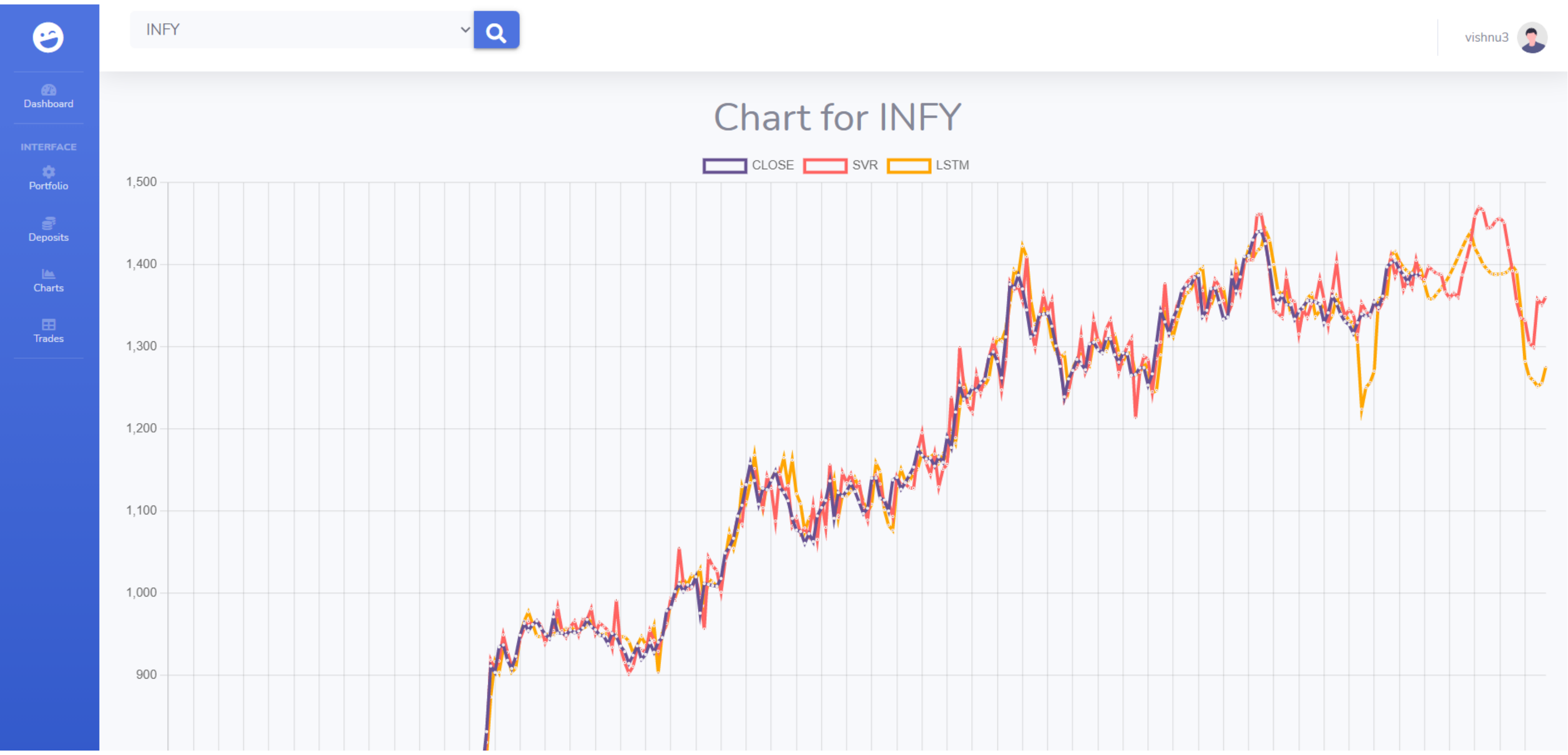
Screenshots:

HOME PAGE

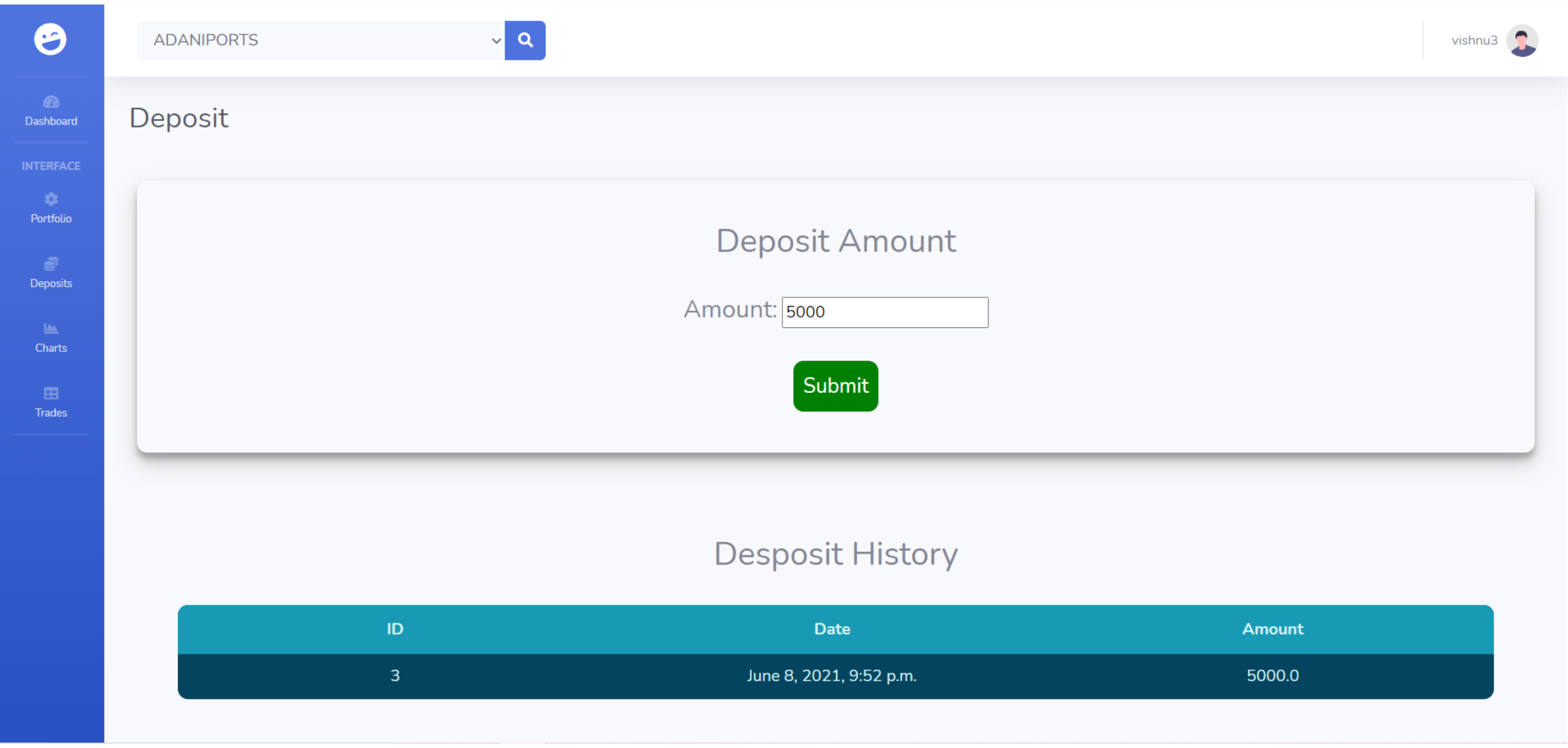


TCS STOCK PREDICTION:

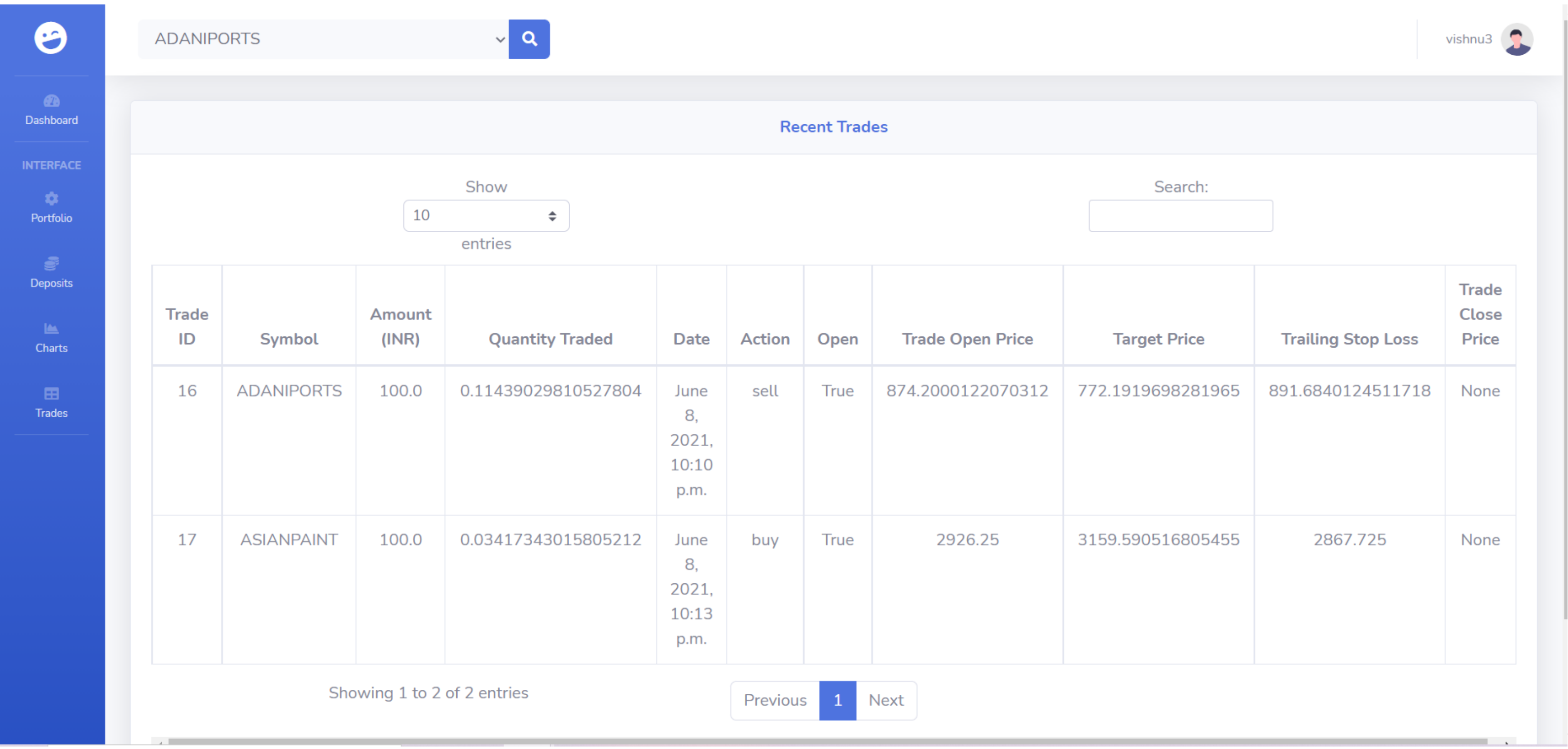
INFY STOCK PREDICTION:



DEPOSIT PAGE:



TRADES PAGE:

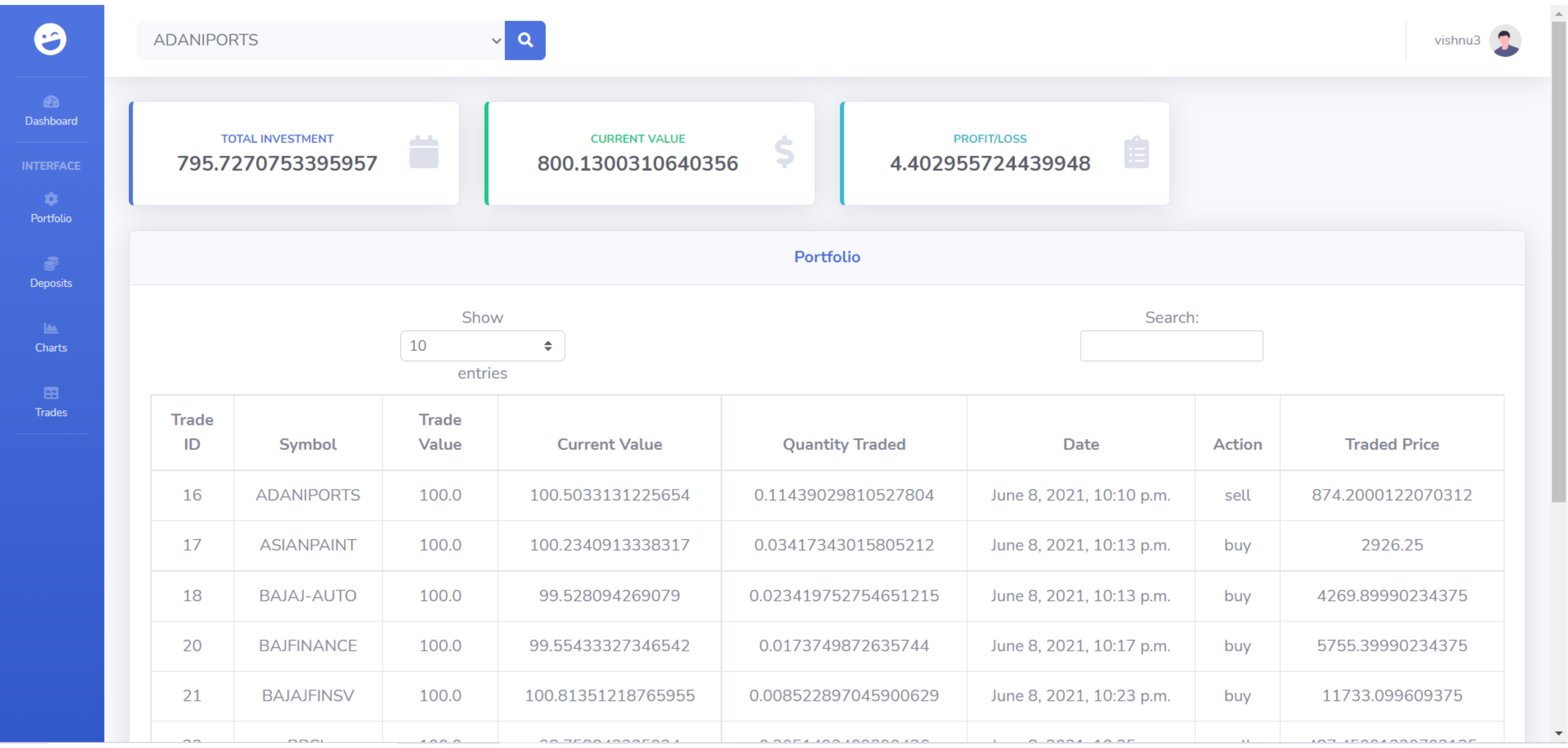


Trade ID	Symbol	Amount (INR)	Quantity Traded	Date	Action	Open	Trade Open Price	Target Price	Trailing Stop Loss	Trade Close Price
16	ADANIPORTS	100.0	0.11439029810527804	June 8, 2021, 10:10 p.m.	sell	True	874.2000122070312	772.1919698281965	891.6840124511718	None
17	ASIANPAINT	100.0	0.03417343015805212	June 8, 2021, 10:13 p.m.	buy	True	2926.25	3159.590516805455	2867.725	None

Showing 1 to 2 of 2 entries

Previous 1 Next

PORTFOLIO PAGE:



Trade ID	Symbol	Trade Value	Current Value	Quantity Traded	Date	Action	Traded Price
16	ADANIPORTS	100.0	100.5033131225654	0.11439029810527804	June 8, 2021, 10:10 p.m.	sell	874.2000122070312
17	ASIANPAINT	100.0	100.2340913338317	0.03417343015805212	June 8, 2021, 10:13 p.m.	buy	2926.25
18	BAJAJ-AUTO	100.0	99.528094269079	0.023419752754651215	June 8, 2021, 10:13 p.m.	buy	4269.89990234375
20	BAJFINANCE	100.0	99.55433327346542	0.0173749872635744	June 8, 2021, 10:17 p.m.	buy	5755.39990234375
21	BAJAJFINSV	100.0	100.81351218765955	0.008522897045900629	June 8, 2021, 10:23 p.m.	buy	11733.099609375

Code:

LSTM MODEL:

```
sc = StandardScaler()
training_set_scaled = sc.fit_transform(train_set)
X_train = []
y_train = []
for i in range(input_days, len(training_set_scaled)):
   X_train.append(training_set_scaled[i-input_days:i, 0])
   y_train.append(training_set_scaled[i, 0])
X_train, y_train = np.array(X_train), np.array(y_train)
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))

   #Defining the LSTM Recurrent Model
regressor = Sequential()
regressor.add(LSTM(units = 50, return_sequences = True, input_shape =
(X_train.shape[1], 1)))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))
regressor.add(Dense(units = 1))

checkpoint_filepath = 'LSTM_Pickled/checkpoint/'+symbol + '/'+symbol +'_lstm'
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error',
metrics=['mse'])
model_checkpoint_callback = tf.keras.callbacks.ModelCheckpoint(
   filepath=checkpoint_filepath,
   save_weights_only=True,
   monitor='loss',
   mode='min',
   save_best_only=True)

regressor.fit(X_train, y_train, epochs =
200,callbacks=[model_checkpoint_callback])
print("Modelf it")
regressor.load_weights(checkpoint_filepath)
print("Model save")
regressor.save('LSTM_Pickled/'+symbol + '_lstm')
```

SVR MODEL:

```
sc = StandardScaler()
training_set_scaled = sc.fit_transform(train_set)
X_train = []
y_train = []

for i in range(input_days, len(training_set_scaled)):
   X_train.append(training_set_scaled[i - input_days:i, 0])
   y_train.append(training_set_scaled[i, 0])
X_train, y_train = np.array(X_train), np.array(y_train)

# Defining the LSTM Recurrent Model
regressor = svm.SVR(kernel='rbf', C=1000.0, gamma=0.1)  # svm.SVR()
regressor.fit(X_train, y_train)
filename = 'finalized_model_' + symbol + '.sav'
pickle.dump(regressor, open(filename, 'wb'))
```

AUTOMATED ASSISTANT:

```
def get_ltp(symbol):
   ticker = yf.Ticker(symbol + ".NS")
   todays_data = ticker.history(period='1d')
   return todays_data['Close'][0]

def execute_trade(user, stock, amount, quantity, type, is_open, open_price,
target, trailing_sl):
   trade = Trade(user = user, stock = stock, amount = amount, quantity = quantity,
type = type, is_open = is_open, open_price = open_price, target = target,
trailing_sl = trailing_sl, date = datetime.now())
   trade.save()
   return True

def create_trade_for_users(symbol, type, cmp, target, trailing_sl):
   user_ids = User.objects.values_list('id', flat=True)
   for user in user_ids:
       user_instance = User.objects.filter(id=user).first()
       open_trade=
Trade.objects.filter(user=user_instance).filter(stock=symbol).filter(is_open=True)
.count()
       wallet = Wallet.objects.filter(user=user).first()
       if wallet is not None and wallet.balance >= 100 and not open_trade:
           execute_trade(user_instance, symbol, 100, 100/cmp, type, True, cmp,
target, trailing_sl)
           wallet.balance = wallet.balance - 100
```

```python
            wallet.save()

def buy_or_sell(cmp, max1, max2, min1, min2, symbol):
   max_of_both = max(max1, max2)
   min_of_both = min(min1, min2)
   max_percentage = (max_of_both - cmp)/cmp
   max_percentage *= 100
   min_percentage = (cmp - min_of_both)/cmp
   min_percentage *= 100
   if(max_percentage >= 8 and min(max1, max2) >= 1.02*cmp): #Worthy of buy
       create_trade_for_users(symbol, "buy", cmp, (max1 + max2)/2, 0.98*cmp)
   elif(min_percentage >= 8 and max(min1, min2) <= 0.98*cmp): #Worthy of sell
       create_trade_for_users(symbol, "sell", cmp, (min1 + min2)/2, 1.02*cmp)

def perform_predictions():
   for symbol in symbol_list:
       print(symbol)
       cmp = get_ltp(symbol)
       vals = main_predict.predict_data(symbol, is_agent=True)
       svr_pred = vals[1][-30:]
       lstm_pred = vals[2][-30:]
       svr_max = max(svr_pred)
       svr_min = min(svr_pred)
       lstm_max = max(lstm_pred)
       lstm_min = min(lstm_pred)
       buy_or_sell(cmp, svr_max[0], lstm_max[0], svr_min[0], lstm_min[0], symbol)

def update_trades():
   for symbol in symbol_list:
       print("Updating " + symbol + " - all trades")
       trades = Trade.objects.filter(stock = symbol).filter(is_open = True)
       cmp = get_ltp(symbol)
       stock = Stock.objects.filter(stock = symbol)
       if stock.first() is not None:
           stock = stock.first()
           old_ltp = stock.ltp
           stock.ltp = cmp
           stock.last_updated = datetime.now()
       stock.save()
       if trades.first() is not None:
           for trade in trades:
               if trade.type == "buy":
                   if cmp >= trade.target or cmp <= trade.trailing_sl:
                       trade.close_price = cmp
                       trade.is_open = False
```

```
            elif cmp >= old_ltp:
                trade.trailing_sl += cmp - trade.open_price
        else:
            if cmp <= trade.target or cmp >= trade.trailing_sl:
                trade.close_price = cmp
                trade.is_open = False
            elif cmp <= old_ltp:
                trade.trailing_sl -= trade.open_price - cmp
        trade.last_updated = datetime.now()
        trade.save()
```

Dataset Description and Sample Data:

In our project, we will be fetching data real-time from APIs such as NSE. Our dataset will include the description of each stock like symbol, date, open, high, low, close, volume and volume weighted average price (VWAP). For prediction purposes, we are only using VWAP.

Sample code to fetch the data:

```
from datetime import date
from nsepy import get_history
reliance_data = get_history(symbol='RELIANCE',
                    start=date(2021,3,1),
                    end=date(2021,3,10))

print(reliance_data)
```

Sample Data:

Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume
2021-03-01	RELIANCE	EQ	2085.80	2110.20	2112.00	2062.50	2103.00	2101.70	2092.87	8159670
2021-03-02	RELIANCE	EQ	2101.70	2122.00	2130.00	2089.10	2108.00	2106.00	2107.78	7915073
2021-03-03	RELIANCE	EQ	2106.00	2121.05	2219.90	2107.20	2207.10	2202.10	2161.54	14733134
2021-03-04	RELIANCE	EQ	2202.10	2180.00	2189.95	2157.70	2174.00	2175.85	2175.57	9892597
2021-03-05	RELIANCE	EQ	2175.85	2156.00	2211.95	2153.05	2174.55	2178.70	2184.35	11773630
2021-03-08	RELIANCE	EQ	2178.70	2168.50	2231.90	2168.00	2193.00	2191.10	2205.66	9002404
2021-03-09	RELIANCE	EQ	2191.10	2200.00	2213.80	2146.60	2190.55	2191.05	2181.47	6993792
2021-03-10	RELIANCE	EQ	2191.05	2207.00	2215.10	2170.25	2179.40	2181.95	2187.67	5316182

CONCLUSION:

Machine learning as we have seen till now, is a very powerful tool and as evitable, it has some great applications. We have seen till now that machine learning is very much dependent upon data. Thus it is important to understand that data is quite invaluable and as simple as it may sound, data analysis is not an easy task. Machine learning has found tremendous application and has evolved further into deep learning and neural networks, but the core idea is more or less the same for all of them.

We have successfully implemented a two model prediction algorithm, which accurately predicts the stock price for the majority of the stocks. In the future, the LSTM model can be further improved by increasing the number of epochs. Due to the time constraint, in our project, we have taken 200 epochs. As this is a neural network, more epochs will result in a better prediction. Further improvements, like the loading time (time taken for prediction is 10 seconds), can be reduced in the future. Our project currently predicts the next 30 days. This can be improved in the future.