JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

FORECAST THE CHARACTERISTICS OF A STOCK

[1]Aniket Madhukar Suradkar, [2]Prashant Pal

[1]B-Tech Student, [2]Scientist "B"
[1]Electronics And information Technology
[1]National Institute Of Electronics And Information Technology Aurangabad (Maharashtra), INDIA.

Abstract : Stock market has been constant fascinating topic but since last few years stockholders desire to hardback return on day to day basis got glorified , then with the support of machine learning stockholders initiate stress-free approaches to get squint of forthcoming market trends. This paper benevolences LSTM grounded methodology to forecast the characteristics of a specific stock, our method put greater concentration on sets of hundred days moving averages. Hundred days moving average is a technical method followed by the stock market professionals to forecast the forthcoming trends of market, which represents the current as well as the characteristics which is stock going to show in upcoming days. As we are scraping data with the API and creating our own dataset hence our method is comfortable with ambiguous data besides it provides output with high accuracy. The results indicates that our method attained superior upshot than other approaches**.**

IndexTerms **-LSTM (long and short term memory), API (Application program interface), machine learning, Hundred days moving average, market trends.**

I. INTRODUCTION

The stock market appears within news every day if we pay attention to it and analyze its characteristics every time it set new zenith and nadir of it pervious prices the charge funding and enterprise possibilities inside the stock market cam make growth if and green set rules can be devised to expected the quick time period rate of an persons stock. Aforementioned strategies of stocks involve usage of artificial neural networks and convolutional neural networks which has an blunders loss at median of 20% , if we see there is possibility of formulating a version the use of recurrent neural network with an expected stock control with much less percentage of mistakes. Dealing with financial time series forecasting or prediction is an imperative focus in today's world. Moreover, forecasting the forthcoming market trend is a perplexing task. Because there are various factors which affects the stock market those are like political, environmental, economy, and the performance of particular sector or company. The various models existing in field of machine learning made algorithmic trading very handy to everyone. Words leading banks, hedge funds and recognized shareholders use computer engendered trading.

.

II. LITERATURE SURVEY

In last few years there are numerous forecasting models are invented and also implemented for analyzing stock market under the umbrella of machine learning and AI. A brief study has been done of models have been used in market since long time. First study is about the Hidden Markov Model (HMM), this model was primarily invented for the in the Speech recognition systems. But later on its ability to handle past time series data it were stated using for forecasting the data which is have linear time intervals. The research published by Kavitha G., Udhakuma A, Nagarajan D on stock market trend analysis using hidden markov model, the model proposed here discuss about the stock market trend analysis by considering difference between in closing value for a particular period foe given observations sequence of state and probability of its valid value to be found for a particular time period.

The second model is part of study is Holt Winters Model, it's an unpretentious method which consist of trend and seasonal factor. Holt winters model fundamentally discuss about three trends, level, and seasonal smoothening parameters. Holt winters model can be divided in two parts first one is additive Holt winters smoothening model and the second one is multiplicative hold winters model. If we make a comparison between holt winters and hidden markov model then holt winters always gain upper hand as it comfortable with long and short both forecasting of economic development trends, holt winters exponential smoothing method is used as well as seasonal fluctuations is usually used. The research proposed by Omar Sharif and Md Zubair Hasan on forecasting stock price using by holt winters model, Thus the method is suitable for intraday trading at other side it won't hold great dexterity when it comes to long term trading or even trend of stock market. But the major limitation of this model is the multiplicative feature of the seasonality it always rises an issue with time frames have very low amount. Later on Artificial neural network comes in the frame, it is capable with AI which exploits parallel computing to gain intelligence from input data on basis of upcoming trend and this model was proposed un a reasech by Wanjwa, Barak Wamkay and Muchemu Lawren as ANN model for Market prediction. It propose high rate of accuracy approximately around 90% as compared to previous models when it comes to financial time series forecasting at other side the major drawback of this model its ot compatible with blackbox because it won't reveal each variables weight and it we over train it will take long time to get train as it consist of many nodes as well as high possibility of recurring errors may occur.

Next model is ARIMA model which is also known as “Autoregressive Integrated Moving Average”, Box and Jenkins introduced ARIMA model in 1970. This model is most prominent method when it comes to the financial forecasting. In ARIMA model future value of stock or variable is a linear combination of past values as well as past errors. Hence this model is only suitable for short term predictions. The another model is part of literature survey is Time Series Linear Model when it comes to constructing of linear time series model, The epitome linear model is first created and data is merged into it that it reflects the characteristics of actual data. The benefit of this model of the time series is actual data is merged with the epitome linear model. Thus seasonal trends and traditional trend both are part of it. Now the last model stands in to be part of survey is recurrent neural network. The recurrent neural network consist of propagation which is used at back to train along with this its node consist of feedback mechanism. Hence is only forecast values on the basis of recent historical data of a particular stock. Hence the major disadvantage stands out with recurrent neural network is possible to feed those words in through much smaller set of input..

III. Recurrent Neural Networks And LSTM

Before heading to the point why LSTM (Long And Short Term Memory) is needed first we need to understand problems with recurrent neural network, for that lets consider an example of vanilla recurrent neural network which aren’t actually used very often now the question is why is it not being used. The major issue is vanishing gradient. Now for recurrent neural networks initially we want long memories, thus the networks will be able to establish connection in between data and noteworthy distance in time. This type of neural networks can make real progress in understanding how language and narrative works and that’s how stock market events are correlated and may more. Now here the more time steps we have the chances of back propagation gradients increased either accumulating, exploding or vanishing down nothing. Now consider he following equation for representing recurrent neural networks

$$ht = \sigma (Uxt + Vht-1)$$

Where, U and V are weight matrix which establish connection in between input and recurrent output. However, if we head back to three time steps in recurrent neural network we can go with the following equation

$$ht = \sigma (Uxt + V (\sigma (Uxt-1 + V (\sigma (Uxt-2)))$$

Above equation states that we work back way in time, essentially adding deeper and deeper layer to our network and this thing causes gradient of the error with respect to weight to the weight matrix U during back propagation through time. Now to mitigate the vanishing and exploding gradient problems and hence it permits deeper networks and recurrent neural network to perform well in practical setting it creates necessity to reduces the multiplication of gradient which are less than zero. The LSTM cell is specifically designed unit of logic that will help reduces the vanishing gradient problem sufficiently to make recurrent neural network more useful for long term memory task that is text sequence prediction. This is done by building an internal memory state which get simple added to the process input which greatly reduces the multiplicative effect of small gradients. The dependency of time and effect of previous inputs are supervised by an interesting concept which is known as forget gate, which concludes which state is to be remembered or forget. Then two other gates are there known as input gate and output gate are featured in LSTM cells. The explanation of LSTM cell is given below and hoe it reduces the vanish gradient problem. The diagram given below represents the classic LSTM cell structure.

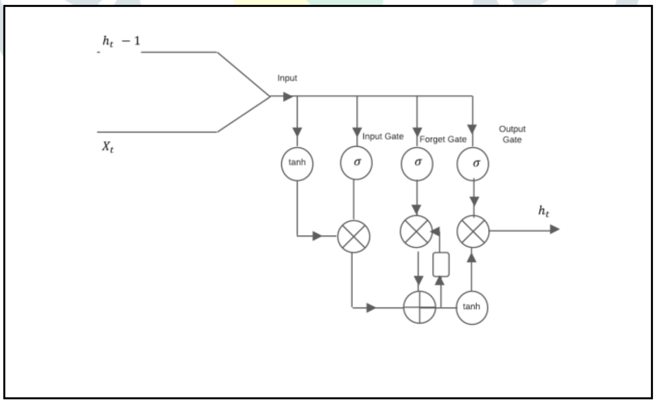


Fig.1. LSTM cell diagram

The dataflow starts from the left side and end up at the right hand side with present input Xt and the previous output cell ht-1 concatenated with each other and getting admitted at the top “data rail”.

Each gate of LSTM cell is explained as follows

I.The input gate: Originally, the offers input is compressed between -1 and 1 using tanh activation function. The expression to represent it as follows

$$g=\tanh(b^g+X_t U^g+h+V^g)$$

Where, U^g and V^g are the weights for the input cells output, respectively and b^g is the input bias. Here the exponent g are not a raised power but rather it signifies that these are the input weight and bias value. This squashed input is then multiplied elements wise by the output of the input gate. Input gate is basically a hidden layer of sigmoid activation nodes with weight X_t and h_(t-1) input values, which gives output values between 0 and 1 when multiplied element wise by the input determines which inputs are switched on and off. In simple words it's just a filter or gate for input gate expression is as follows

$$i= \sigma (b^{i} + X_t U^i + h_{t-1} V^i)$$

Now the output of input stage of LSTM cell can be expressed as below where the ^ ο operator express element wise multiplication 〖 g〗 ^ ο i, In input and output as the weights for squashed input g.

II. The internal stage and The forget gate: This is kind of magical stage of LSTM. Now here new variable is introduced S_t which exist in inner state of LSTM cell. This state is delayed by one time step and ultimately added to the to 〖 g〗 ^ ο i input to provide an internal recurrent loop to learn the relationship between input separated by time. Two thing to notice first there is a forget gate here this gate is again a sigmoid activated set of notes here the previous state should be remembered and which should be forgotten. This allows the LSTM cell to learn appropriate context. The forget gate can expressed as

$$f= \sigma (b^f + X_t U^f + h_{t-1} V^f)$$

The output of the element wise products of the previous state and the forget gate is expressed as S_(t-1) ^o f now here once more forget gate acts like weight for internal state. The second thing to notice about this stage is that the forget gate filtered state is simply added to the input rather than multiplied by it, or mixed with it via weight and sigmoid activation function as occurs in a standard recurrent neural network. This is important to reduce the issue of vanishing gradients. The output from this stage S_t is expressed by

$$S_t= S_{t-1} \circ f g \circ i$$

III. The output gate : The final stage of the LSTM cell is input gate, the output gate has two components another tanh squashing function and output sigmoid gate function. The output sigmoid gating function like the other gating function in the cell is multiplied by the squashed states S_t to determine which values of the state are output from the cell. The LSTM cell is very flexible with gating function controlling and what is input and what is remembered in internal state variable and finally output from LSTM cell. Now here the output gate can be expressed as

$$o= \sigma (b^o + X_{t} U^o + h_{t-1} V^o)$$

That the final output of the cell can be expressed as

$$h_t = \tanh(S_t) \circ O$$

IV. Proposed work

Before heading toward the technical aspects of model fires take glace of exponential moving average. Why we are using exponential average the reason behind it is if we compare the standard average and exponential moving average the standard average leads to the failure as it calculating only with the fixed set of that but in case of exponential moving average it is always get updated with new values to the set at each time lap if we see the equation of standard average

$$X_{t-1} = \frac{1}{N}\sum_{i=t-N}^{t} X_i$$

Where t-1 is average value of stock price. like as mentioned above the exponential moving average value is maintained over the time,

$$X_{(t+1)}=EMA_t=\gamma.EMA_{(t-1)}+[1-\gamma]\ X_1$$

Above equation basically calculates the exponential moving average where fron t+1 time step and uses that one step for next prediction at next step.γ Decides what will be the significant role of the most recently added value or a predicted value is to the exponentially moving average. For example a $\gamma = 0.1$ gets only 10% of the current value into the EMA. Because only very small fraction of the recently added value is taken which allows to preserve much older values can see in very early stage of average. Lets consider $X_t= 0.5$ EMA=0.5 and $\gamma=0.5$ and if we put these values in above given equation it will be,

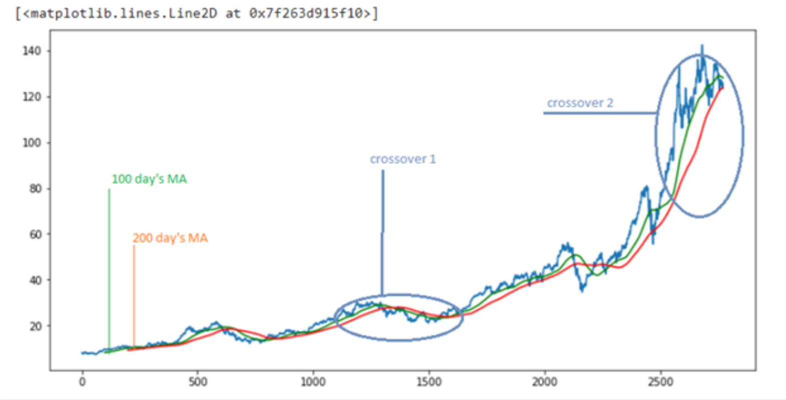
$$X_{(t+1)}=EMA_t=0.45$$

And still if we keep going with same values it will take larger set of values, if we see the rigid formula to calculate the Exponential moving average is

$$EMA = \text{Closing price X multiplier} + \text{EMA (previous day) X (1-multiplier)}$$

This EMA gives a higher weigh to the recent prices while the standard moving average assign equal weight to all values. The technical analyst of market follows the a hundred sticks and two hundred sticks approach. The approach has an relation with the previous two hundred days patter and a hundred days sample if the 100 days candlestick pattern is bullish over the two hundred days sample then stock going to show the uptrend if it goes beneath that then there is large risk that stock can pass down trend for sure

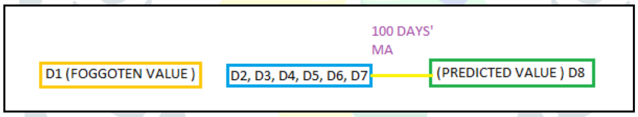
The following graph shows a hundred and two hundred days' moving average of tesla stock.A term moving averages is nothing but the average of last 100 and 200 days' closing price of the stock data. As we can observe in the graph the 100 days moving average won't start with the preliminary data of stock it will start fluctuating after 100 values and same in case of 200 days' moving average it will



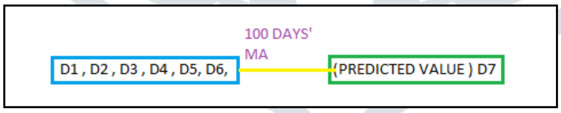
also appear on the graph after 200 values. As mentioned above that the technical analyst of market follows the strategy that a hundred days' data is superimposing two hundred days' data then stock is going to be uptrend for next few days. And if a hundred days data is beneath of two hundred days moving average data then stock will fall down or downtrend.

The same thing can be observed in above graph at crossover1, here the thing can be observed as the a hundred days' moving average is over two hundred days moving average the original vale of stock is in uptrend or peaks can be observed. Now if we focus on the second crossover2 a hundred days' moving data is over the two hundred days data for a long time as a result the original value of the stock is continuously getting height although it can be observed the as the distance between a hundred and two hundred days' moving data is getting close to each other as result there are bit down trend in price of stock.

Here I'm proposing same concept with the help of LSTM model which will take the moving average of a hundred values of closing price of stock. And once new value is predicted the initial value of data will get eliminated through the forget gate (ft)



And after that the predicted value will be stored with cell state vector and it will feed for the next fetch to the dataset trough the input state vector now the previous data is updated with the last predicted value



The following diagram represents the functional block diagram of the whole model

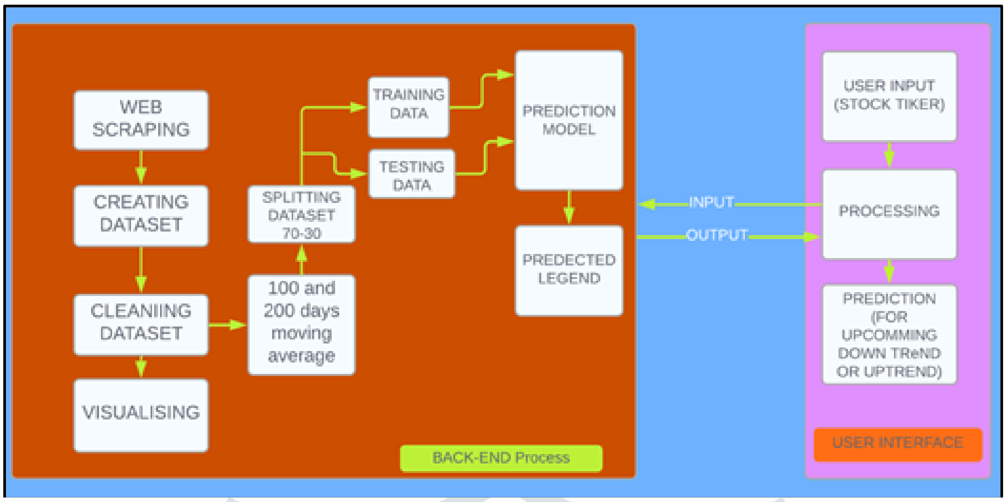


Fig.2. Functional block diagram

I. At initial stage the data will be scrapped front the internet

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2010-06-02	8.089959	8.097910	7.961212	8.071917	688548000	0.0	0.0
2010-06-03	8.109526	8.120842	7.963654	8.046530	650106800	0.0	0.0
2010-06-04	7.896373	8.009218	7.786893	7.827566	758304400	0.0	0.0
2010-06-07	7.898821	7.925121	7.662122	7.674049	886942000	0.0	0.0
2010-06-08	7.744387	7.761513	7.512276	7.624815	1000770400	0.0	0.0
...	...	...	...	...	...	...	...
2021-05-28	125.040116	125.269148	124.024423	124.084167	71311100	0.0	0.0
2021-06-01	124.552181	124.821039	123.416993	123.755554	67637100	0.0	0.0
2021-06-02	123.755550	124.711498	123.526525	124.532257	59278900	0.0	0.0
2021-06-03	124.153873	124.323154	122.610411	123.018684	76229200	0.0	0.0
2021-06-04	123.546445	125.627630	123.327372	125.358765	75169300	0.0	0.0

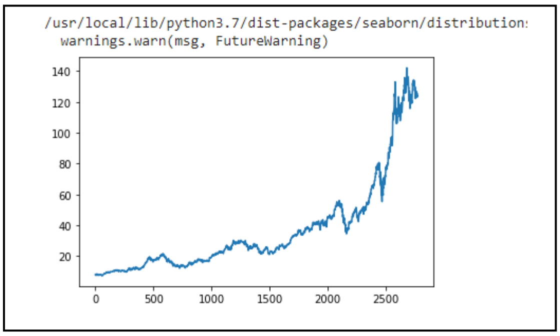
2772 rows × 7 columns

.

II. At second stage it will get clean like all the empty rows which don't have any value is dropped out

	Open	High	Low	Close	Volume
0	8.089959	8.097910	7.961212	8.071917	688548000
1	8.109526	8.120842	7.963654	8.046530	650106800
2	7.896373	8.009218	7.786893	7.827566	758304400
3	7.898821	7.925121	7.662122	7.674049	886942000
4	7.744387	7.761513	7.512276	7.624815	1000770400

III. At next stage data is visualized to get actual glance of the stock price.



IV. Then 100 and 200 days moving average is taken from the acquired data.

```
0              NaN
1              NaN
2              NaN
3              NaN
4              NaN
           ...
2767    127.963521
2768    127.944435
2769    127.890234
2770    127.809682
2771    127.783003
Name: Close, Length: 2772, dtype: float64
```

V. All this data is divided in the 70 and 30 percent of its actual size

```
(1940, 1)
(832, 1)
```

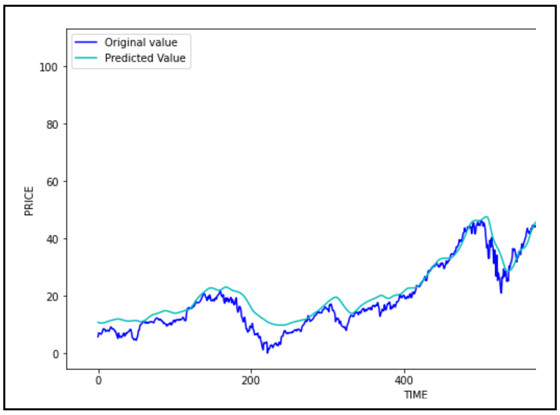
VI. Creating model.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 100, 50)           10400

 dropout (Dropout)           (None, 100, 50)           0

 lstm_1 (LSTM)               (None, 100, 60)           26640

 dropout_1 (Dropout)         (None, 100, 60)           0

 lstm_2 (LSTM)               (None, 100, 80)           45120

 dropout_2 (Dropout)         (None, 100, 80)           0

 lstm_3 (LSTM)               (None, 50)                26200

 dropout_3 (Dropout)         (None, 50)                0

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 108,411
Trainable params: 108,411
Non-trainable params: 0
_________________________________________________________________
```

VII. Training Model.

```
58/58 [==============================] - 13s 216ms/step - loss: 0.0038
Epoch 38/50
58/58 [==============================] - 13s 217ms/step - loss: 0.0035
Epoch 39/50
58/58 [==============================] - 13s 217ms/step - loss: 0.0033
Epoch 40/50
58/58 [==============================] - 13s 218ms/step - loss: 0.0034
Epoch 41/50
58/58 [==============================] - 13s 217ms/step - loss: 0.0034
Epoch 42/50
58/58 [==============================] - 13s 217ms/step - loss: 0.0036
Epoch 43/50
58/58 [==============================] - 13s 218ms/step - loss: 0.0037
Epoch 44/50
58/58 [==============================] - 13s 223ms/step - loss: 0.0036
Epoch 45/50
58/58 [==============================] - 13s 220ms/step - loss: 0.0040
Epoch 46/50
58/58 [==============================] - 13s 217ms/step - loss: 0.0035
Epoch 47/50
58/58 [==============================] - 13s 218ms/step - loss: 0.0033
Epoch 48/50
58/58 [==============================] - 12s 215ms/step - loss: 0.0035
Epoch 49/50
58/58 [==============================] - 13s 219ms/step - loss: 0.0031
Epoch 50/50
56/58 [===========================>..] - ETA: 0s - loss: 0.0035
```

VIII. At this stage model will find out the difference between the 100 and 200 days moving average as it can be able to find the nearest line to the actual price of stock. And final stock prediction legend is out



V.Conclusion

Long and short term memory is not totally relies on intraday trading, as well as it won't relies our ability to predict the current state of the economy not even the characteristics of the stock market. It is totally depends on the closing price of the stock and investors uses this method for long term investments. But at other side the difference between moving averages between closing values founded in optimum sequence. When we observed the steady probability of the market pattern and prediction of model is values as obtained. LSTM model can compete reasonably well with emerging forecasting techniques in long as well as short term predictions

VI.REFERENCES

[1] https://www.researchgate.net/publication/269405423_ANN_Model_to_Predict_Stock_Prices_at_Stock_Exchange_MarketsJ. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] https://www.researchgate.net/publication/228607056_Using_neural_networks_to_forecast_stock_market_pricesK. Elissa, "Title of paper if known," unpublished.

[3] https://www.researchgate.net/publication/345991990_Stock_Market_Trend_Prediction_Using_Hidden_Markov_Model.

[4] https://www.researchgate.net/publication/360913562_Forcasting_Stock_Price_PT_Indonesian_Telecomunication_with_ARCH-GARCH_ModelM. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[5] https://www.researchgate.net/publication/360913562_Forcasting_Stock_Price_PT_Indonesian_Telecomunication_with_ARCH-GARCH_Model

[6] https://www.researchgate.net/publication/261179224_Stock_price_prediction_using_the_ARIMA_model

[7] https://www.researchgate.net/publication/340720727_A_Time_Series_Analysis-Based_Stock_Price_Prediction_Framework_Using_Machine_Learning_and_Deep_Learning_Models