Original Article
ISSN (Online): 2456-8651

International Journal of Engineering Science Technologies
May-June 2022 6(3), 1–16

STOCK PRICE PREDICTION USING GRID HYPER PARAMETER TUNING IN GATED RECURRENT UNIT

Bhavsar Shachi [1], Ravi Gor [2]

[1] Research Scholar, Department of Mathematics, Gujarat University, Ahmedabad-380009, India
[2] Department of Mathematics, Gujarat University, Ahmedabad-380009, India

Received 23 April 2022
Accepted 28 May 2022
Published 24 June 2022

Corresponding Author

Bhavsar Shachi,

shachibhavsarfmg@gujaratuniversity.ac.in

DOI 10.29121/IJOEST.v6.i3.2022.345

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ABSTRACT

Nowadays people are using social media to show their talent, to voice their viewpoint to society, etc. The use of social media has drastically grown during and after pandemic. Since, the power of social media is known to us, it would be beneficial to invest in such trending companies. But, understanding market pattern will be required to get maximum benefit from stock market, otherwise it may lead to losses. Machine learning is an essential tool for predicting such tasks. Here deep learning based Gated Recurrent Unit neural network is used for prediction. To develop optimized model, grid search algorithm is used for Gated Recurrent Unit hyper parameter tuning. Also, the hyper parameter values obtained by the model was used to verify and predict stock prices for other companies.

Keywords: Machine Learning, Gated Recurrent Unit, Back Propagation, Neural Networks, Hyper Parameter Tuning

1. INTRODUCTION

Stock price prediction is one of the matters confronting investors in everyday financial market. Researchers are working to study the market behaviour and pattern. As understanding market will help investors to obtain higher trading benefits, to know trend and avoid losses. Sometimes the decision process of trend prediction may get biased due to aspects such as trader's psychology, investors demand of high returns, social media statements, political scenario, others opinion, influencers, etc. The trends might be predicted false or wrong if decisions were made in panic situations. Machine learning can be the best option to avoid being

How to cite this article (APA): Shachi, B., and Gor, R. (2022). Stock Price Prediction Using Grid Hyper Parameter Tuning in Gated Recurrent Unit. *International Journal of Engineering Science Technologies, 6*(3), 1-16. doi: 10.29121/IJOEST.v6.i3.2022.345

emotionally invested. Deep learning can provide promising results while working with highly volatile data.

Recurrent Neural Network (RNN) is one of the models of deep learning used for both classification and regression. The Gated Recurrent Unit (GRU) is an upgraded model of RNN which uses the back propagation algorithm to find optimal weights for model. RNN has mainly two sub models GRU and Long Short-Term Memory (LSTM). Here, GRU model was selected as it contains two gates as compared to three gates in LSTM which decreases the computational cost.

One of the results of efficient market hypothesis says that stock prices can be predicted by analysing historical price movements Fama (1970) Gao et al. (2021) Hence, historical stock data of previous years is used for the prediction. Due to the pandemic situation use of social media increased. People tends to move towards social media more not only to entertain but to share their opinions and ideas to world. Increasing use of social media accounts leads to study the pattern and behaviour, as investing in social media companies would be profitable. So, stock prices of the social media companies were analysed and predicted, to understand their pattern.

2. LITERATURE REVIEW

Shena et al. (2018) used hybrid of Gated recurrent unit and support vector machines to overcome gradients explosion and vanishing gradient problem for financial signal forecasting. Three stock indexes S&P 500 Index, Heng Seng Index and Deutscher Aktien index of previous 27 years were used for forecasting. The hybrid approach of model consist of traditional layers with Gated recurrent unit cell and output layer was changed with support vector machine model. The obtained results were compared with traditional machine learning models such as support vector machines and deep neural network. The comparison of results concluded that hybrid model outperformed another traditional model. Shena et al. (2018)

Raut and Sethia (2018) predicted (t+5) th day stock prices and provided buying and selling strategies. For model 50 different indicators were calculated from which independent component analysis was applied for dimension reduction. Performance of Support Vector Machine, Artificial neural network, Long Short-Term Memory, and Gated Recurrent Unit were compared to find optimal model where Long Short Term Memory model provided highest returns. Raut and Sethia (2018)

Das et al. (2019) predicted stock prices of four stock indices by means of evolutionary algorithm and Machine learning algorithm. Firefly algorithm, Genetic algorithm, Principal component analysis and Factor analysis were used as a feature reduction technique. Reduced features were applied on Extreme learning machine, online sequential extreme learning machine and Recurrent backpropagation neural network and the results were compared to check optimised model. The results shows that combination of firefly algorithm for feature reduction and online sequential extreme learning machine for prediction gives optimized results. Das et al. (2019)

Selvamuthu et al. (2019) used traditional machine learning models i.e., Support vector machines and Neural networks for forecasting stock price. In Neural networks three models Scaled conjugate gradient, Levenberg-Marquardt and Bayesian regularization were applied on 15 min dataset of India company and tick dataset. Every model provides similar accuracy but first two models take few hours to forecast, and Bayesian regularization model takes few days to forecast providing best accuracy from all the models used. Selvamuthu et al. (2019)

Polamuri and Srinivas (2019) used supervised learning technique Random Forest for stock price forecasting. Extremes randomize tree regressor is used in Random Forest model, to improve forecasting results. This extreme randomize tree offers reduction in variance. Linear regression, Multivariate regression, Decision tree and Support vector machine algorithms were used for forecasting where Random Forest algorithm with extreme randomize regressor give best result. Polamuri and Srinivas (2019)

Pengfei and Yan (2019) used the combined PSR and Long Short-Term Memory model to predict prices for six stock indices. ARIMA, Support Vector Regressor, Multi-layer Perceptron, Long Short-Term Memory were used to compared with the proposed combined model. The compared result shows that, combined LSTM and PSR model gives better predictions as compared to others for stock indices. Pengfei and Yan (2019).

Assaf and Kolasani (2020) predicted stock movements with the help of sentiment analysis. Different traditional machine learning models were applied on different sentiments for prediction in which Support vector machine performed well. So, Support Vector Machine was selected for stock sentiment analysis based on tweets made with word stock. The sentiment values of previous day were used for next day's prediction. Boosted regression tree and multi-layer perceptron Neural Network were compared to train sentiment values for stock index value estimation. Results showed that from both model multi-layer perceptron Neural Network performed better. Assaf and Kolasani (2020)

Shahi et al. (2020) compared Long Short-Term Memory and Gated Recurrent Unit for stock price prediction. Information such as last traded price, news headline and body were used with stock price data. Experiments were carried out with news sentiments and without news sentiments where the news sentiment model performed better. Both models with the financial news data performed equally well where Gated Recurrent Unit performed slightly better compared to Long Short-Term Memory. Shahi et al. (2020)

Touzani and Douzi (2021) studied the trading strategies of Moroccan stock market. Two models Long Short-Term Memory and Gated Recurrent Unit were used for prediction. Long Short-Term Memory model was used for short term predictions and Gated Recurrent Unit model was used for medium term predictions. Based on prediction, rules were implemented to buy or sell stocks. Transactions were simulated applying these rules to estimate returns for stocks selections. Touzani and Douzi (2021)

Ghosh and Gor (2022) used K-means clustering and Random Forest Regression algorithms for sales prediction. They used clustering methods for ad campaigning analysis. First, ad groups are created using the K-Means clustering algorithm then Random Forest Regressor algorithm is used to optimize sales conversion and predict future sales. Impressions, clicks, and spent are used as independent variables to predict total number of people that asked about the product after viewing the ad on Facebook. They also calculated Mean Absolute Error and Root Mean Square Error. The integration of two algorithms K-means clustering and Random Forest regressor gives permissive result with 75% accuracy. Ghosh and Gor (2022), Bhavsar and Gor (2022)

Srivinay et al. (2022) predicted stock prices using Prediction rule ensemble technique and Deep neural network. Initially moving average technical indicators were used with 20, 50, 200 days average. Then with the help of prediction rule ensemble technique various rules were applied for stock price prediction. Also, Hyper parameter such as layers, learning rate, epochs and neurons were hyper tuned. At the end the results of both methods were combined by average which shows that the combined results show better performance than Artificial neural network and Deep neural network model. Srivinay et al. (2022)

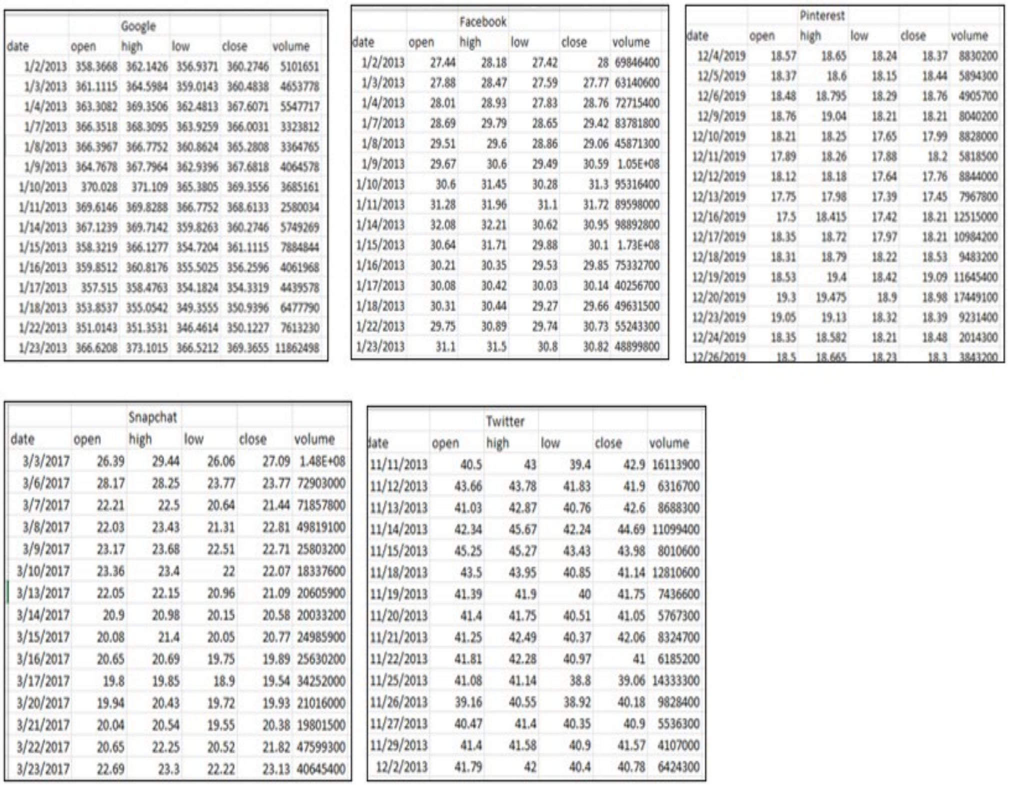
3. DATA DESCRIPTION AND DATA PRE-PROCESSING

Data Collection: Here, stock price data for five different social media companies Google (2344 days), Facebook (2344 days), Snapchat (1295 days), Pinterest (601 days) and Twitter (2127 days) was obtained namely dataset1 to 5 respectively. The dataset includes high, open, low, close, adjusted close price and volume of the stocks. (fin) (wsj)

Data cleaning: There was no null/Nan values in dataset. Also, no outliers were removed form data. The column with adjusted close price was removed as it was highly corelated with close price. (Bhavsar & Gor, Predicting Restaurant Ratings using Back Propagation Algorithm, 2022)

Train and test data: 70% / 30% of ratio was used for training and testing purpose respectively. (Bhavsar & Gor, Predicting Restaurant Ratings using Back Propagation Algorithm, 2022)

Figure 1



		Google			
date	open	high	low	close	volume
1/2/2013	358.3668	362.1426	356.9371	360.2746	5101651
1/3/2013	361.1115	364.5984	359.0143	360.4838	4653778
1/4/2013	363.3082	369.3506	362.4813	367.6071	5547717
1/7/2013	366.3518	368.3095	363.9259	366.0031	3323812
1/8/2013	366.3967	366.7752	360.8624	365.2808	3364765
1/9/2013	364.7678	367.7964	362.9396	367.6818	4064578
1/10/2013	370.028	371.109	365.3805	369.3556	3685161
1/11/2013	369.6146	369.8288	366.7752	368.6133	2580034
1/14/2013	367.1239	369.7142	359.8263	360.2746	5749269
1/15/2013	358.3219	366.1277	354.7204	361.1115	7884844
1/16/2013	359.8512	360.8176	355.5025	356.2596	4061968
1/17/2013	357.515	358.4763	354.1824	354.3319	4439578
1/18/2013	353.8537	355.0542	349.3555	350.9396	6477790
1/22/2013	351.0143	351.3531	346.4614	350.1227	7613230
1/23/2013	366.6208	373.1015	366.5212	369.3655	11862498

		Facebook			
date	open	high	low	close	volume
1/2/2013	27.44	28.18	27.42	28	69846400
1/3/2013	27.88	28.47	27.59	27.77	63140600
1/4/2013	28.01	28.93	27.83	28.76	72715400
1/7/2013	28.69	29.79	28.65	29.42	83781800
1/8/2013	29.51	29.6	28.86	29.06	45871300
1/9/2013	29.67	30.6	29.49	30.59	1.05E+08
1/10/2013	30.6	31.45	30.28	31.3	95316400
1/11/2013	31.28	31.96	31.1	31.72	89598000
1/14/2013	32.08	32.21	30.62	30.95	98892800
1/15/2013	30.64	31.71	29.88	30.1	1.73E+08
1/16/2013	30.21	30.35	29.53	29.85	75332700
1/17/2013	30.08	30.42	30.03	30.14	40256700
1/18/2013	30.31	30.44	29.27	29.66	49631500
1/22/2013	29.75	30.89	29.74	30.73	55243300
1/23/2013	31.1	31.5	30.8	30.82	48899800

		Pinterest			
date	open	high	low	close	volume
12/4/2019	18.57	18.65	18.24	18.37	8830200
12/5/2019	18.37	18.6	18.15	18.44	5894300
12/6/2019	18.48	18.795	18.29	18.76	4905700
12/9/2019	18.76	19.04	18.21	18.21	8040200
12/10/2019	18.21	18.25	17.65	17.99	8828000
12/11/2019	17.89	18.26	17.88	18.2	5818500
12/12/2019	18.12	18.18	17.64	17.76	8844000
12/13/2019	17.75	17.98	17.39	17.45	7967800
12/16/2019	17.5	18.415	17.42	18.21	12515000
12/17/2019	18.35	18.72	17.97	18.21	10984200
12/18/2019	18.31	18.79	18.22	18.53	9483200
12/19/2019	18.53	19.4	18.42	19.09	11645400
12/20/2019	19.3	19.475	18.9	18.98	17449100
12/23/2019	19.05	19.13	18.32	18.39	9231400
12/24/2019	18.35	18.582	18.21	18.48	2014300
12/26/2019	18.5	18.665	18.23	18.3	3843200

		Snapchat			
date	open	high	low	close	volume
3/3/2017	26.39	29.44	26.06	27.09	1.48E+08
3/6/2017	28.17	28.25	23.77	23.77	72903000
3/7/2017	22.21	22.5	20.64	21.44	71857800
3/8/2017	22.03	23.43	21.31	22.81	49819100
3/9/2017	23.17	23.68	22.51	22.71	25803200
3/10/2017	23.36	23.4	22	22.07	18337600
3/13/2017	22.05	22.15	20.96	21.09	20605900
3/14/2017	20.9	20.98	20.15	20.58	20033200
3/15/2017	20.08	21.4	20.05	20.77	24985900
3/16/2017	20.65	20.69	19.75	19.89	25630200
3/17/2017	19.8	19.85	18.9	19.54	34252000
3/20/2017	19.94	20.43	19.72	19.93	21016000
3/21/2017	20.04	20.54	19.55	20.38	19801500
3/22/2017	20.65	22.25	20.52	21.82	47599300
3/23/2017	22.69	23.3	22.22	23.13	40645400

		Twitter			
date	open	high	low	close	volume
11/11/2013	40.5	43	39.4	42.9	16113900
11/12/2013	43.66	43.78	41.83	41.9	6316700
11/13/2013	41.03	42.87	40.76	42.6	8688300
11/14/2013	42.34	45.67	42.24	44.69	11099400
11/15/2013	45.25	45.27	43.43	43.98	8010600
11/18/2013	43.5	43.95	40.85	41.14	12810600
11/19/2013	41.39	41.9	40	41.75	7436600
11/20/2013	41.4	41.75	40.51	41.05	5767300
11/21/2013	41.25	42.49	40.37	42.06	8324700
11/22/2013	41.81	42.28	40.97	41	6185200
11/25/2013	41.08	41.14	38.8	39.06	14333300
11/26/2013	39.16	40.55	38.92	40.18	9828400
11/27/2013	40.47	41.4	40.35	40.9	5536300
11/29/2013	41.4	41.58	40.9	41.57	4107000
12/2/2013	41.79	42	40.4	40.78	6424300

Figure 1 Dataset description with all parameters (fin) (wsj)

4. MODEL DESCRIPTION OF BIDIRECTIONAL GRU WITH DENSE LAYERS CHO ET AL. (2014) RAJPUROHIT ET AL. (2021)

Gated Recurrent Unit (GRU) can be considered as an upgraded or improvised version of standard Recurrent Neural Network. It was introduced by Cho, et al. in

2014 and known as variation of LSTM. GRU's main aim is to solve vanishing gradient problem which arises in typical Recurrent neural networks. Hidden layer of GRU contains mainly two gates Update gate and Reset gate that helps in how much or which information to be passed in future and which past information to be forget shown in figure (A). Rajpurohit et al. (2021)

Figure 2

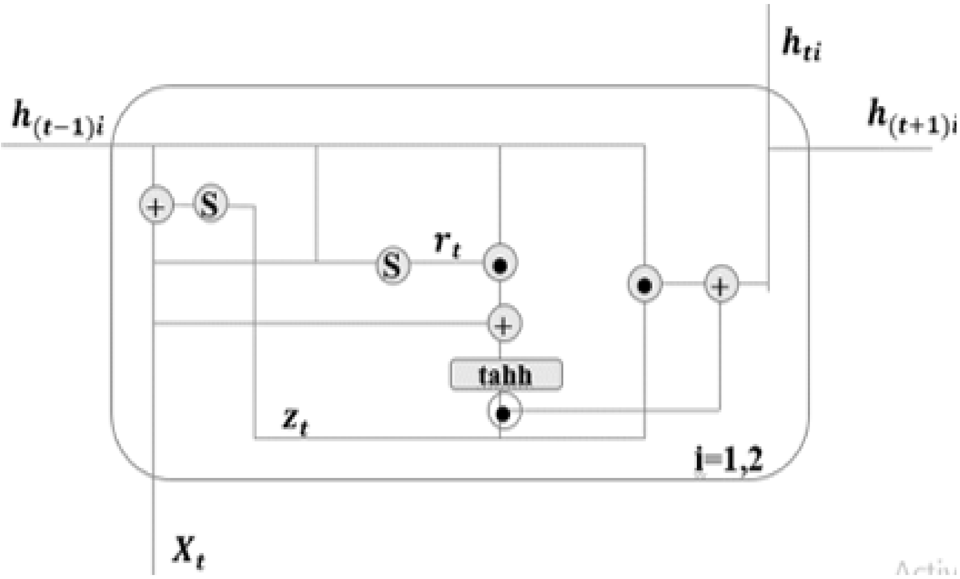


Figure 2 GRU structure for update and reset gate Rajpurohit et al. (2021)

For time step t, update gate can be termed as z_t and defined as $z_t = \sigma(W_z x_t + U_z h_{t-1})$. Similarly reset gate can be termed as r_t and defined as $r_t = \sigma(W_r x_t + U_r h_{t-1})$. Correspondingly, hidden state computed as $h_t' = tanh\ (W x_t + r_t \odot U h_{t-1})$. And the final output computed as $h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t'$. Where, x_t is an input of current state, h_{t-1} is an output of previous hidden state and W_z, W_r, U_z, U_r, W, U are weights. $\odot$ this notation denotes the pairwise multiplication or a Hadamard product. Cho et al. (2014), Rajpurohit et al. (2021), Bhavsar and Gor (2022)

Adaptive moment estimation – Adam optimizer Kingma and Ba (2015), Dozat (2015)

As GRU is a neural network, it also forward and backward propagate in similar manner to Artificial Neural Network. Here Adaptive moment estimation optimizer was used for backpropagation. Following is the equation for Adam optimizer:

$w'_t = w_t - \frac{\propto V'_t}{\sqrt{S'_t + e}}$ where $V'_t = \frac{V_t}{1-\beta_1^t}$, $S'_t = \frac{S_t}{1-\beta_2^t}$, $V_t = \beta_1 V_{t-1} + (1-\beta_1)\frac{\partial error}{\partial w_t}$ and $S = \beta_2 S_{t-1} + (1-\beta_2)\left[\frac{\partial error}{\partial w_t}\right]^2$ Where V is exponential moving average and S squared moving average of gradients, initially taken as 0. Default values taken as ∝=0.001, β_1=0.9, β_2=0.999 and e=10^ (-8). Kingma and Ba (2015), Dozat (2015)

Figure 3

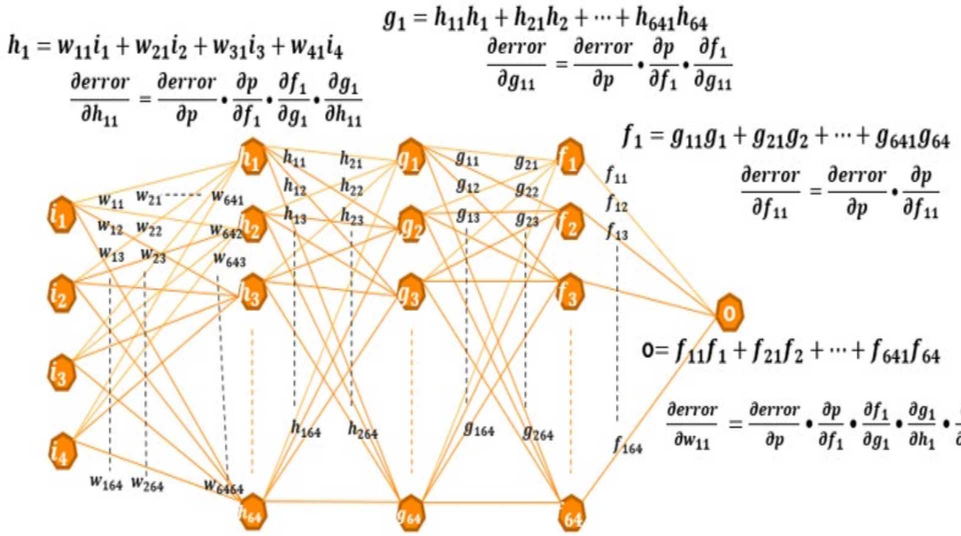

$g_1 = h_{11}h_1 + h_{21}h_2 + \cdots + h_{641}h_{64}$
$\frac{\partial error}{\partial g_{11}} = \frac{\partial error}{\partial p} \cdot \frac{\partial p}{\partial f_1} \cdot \frac{\partial f_1}{\partial g_{11}}$
$h_1 = w_{11}i_1 + w_{21}i_2 + w_{31}i_3 + w_{41}i_4$
$\frac{\partial error}{\partial h_{11}} = \frac{\partial error}{\partial p} \cdot \frac{\partial p}{\partial f_1} \cdot \frac{\partial f_1}{\partial g_1} \cdot \frac{\partial g_1}{\partial h_{11}}$
$f_1 = g_{11}g_1 + g_{21}g_2 + \cdots + g_{641}g_{64}$
$\frac{\partial error}{\partial f_{11}} = \frac{\partial error}{\partial p} \cdot \frac{\partial p}{\partial f_{11}}$
$o = f_{11}f_1 + f_{21}f_2 + \cdots + f_{641}f_{64}$
$\frac{\partial error}{\partial w_{11}} = \frac{\partial error}{\partial p} \cdot \frac{\partial p}{\partial f_1} \cdot \frac{\partial f_1}{\partial g_1} \cdot \frac{\partial g_1}{\partial h_1} \cdot \frac{\partial h_1}{\partial w_{11}}$
i_1
i_2
i_3
i_4
w_{11} w_{21} w_{641}
w_{12} w_{22} w_{642}
w_{13} w_{23} w_{643}
w_{164} w_{264} w_{6464}
h_1
h_2
h_3
h_{64}
h_{11} h_{21}
h_{12} h_{22}
h_{13} h_{23}
h_{164} h_{264}
g_1
g_2
g_3
g_{64}
g_{11} g_{21}
g_{12} g_{22}
g_{13} g_{23}
g_{164} g_{264}
f_1
f_2
f_3
64
f_{11}
f_{12}
f_{13}
f_{164}
o

Figure 3 Proposed Neural network and Backpropagation Algorithm

5. PROPOSED METHODOLOGY

The model is divided into mainly three parts

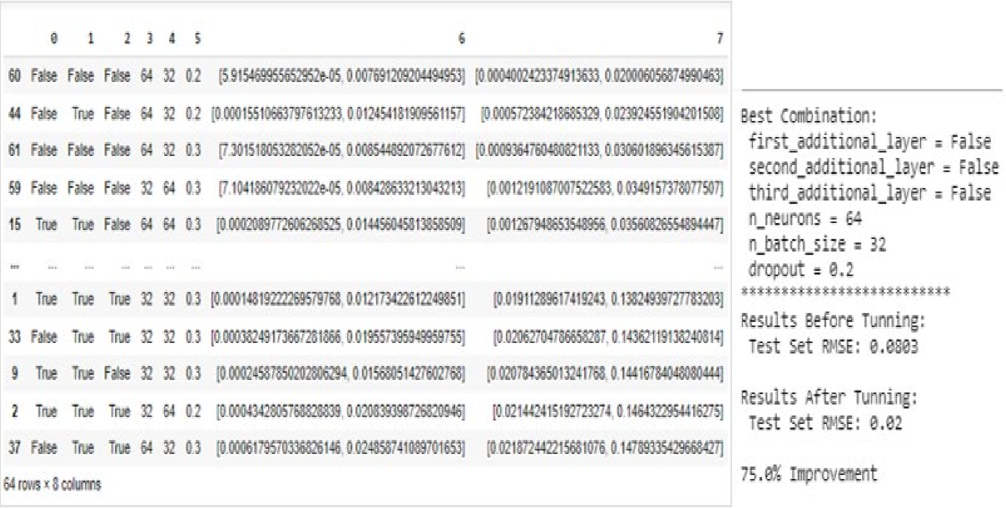
- Data preparation which includes pre-processing, scaling, and cleaning.
- Hyper-parameter tuning with the help of Grid Search algorithm on training dataset.
- Stock price prediction with the help of Gated Recurrent Unit.

The GRU model with following configuration was developed. Same as the number of parameters i.e., 5 neurons are taken in input and 1 neuron in output layer.

- For hidden layer, 'tanh' activation function was taken.
- Adam optimizer was used for backpropagation. (The optimizer was considered in model because it has ability to update learning rate as well as gradient at every step.)
- Proposed Backpropagation Algorithm is shown in Figure 3
- Truncated normal was taken as kernel initializer.
- 40 epochs were considered for model with early stopping criteria.
- As the model was regression type, output layer contains linear activation function.
- Optimized values for number of neurons, layers, dropout value, etc were obtained by hyper parameter tuning.

Hyper parameters help to control the process of learning for given model. Hyper parameter tuning provides optimal values of hyper parameters for the proposed model. Here, hyper parameters such as number of hidden layers, number of neurons in each layer, batch size and dropout are tuned with the help of Grid search algorithm. Where the optimal values obtained for number of hidden layers was 3, number of neurons was 64, batch size was 64 and dropout 0.2. Once the model gets tuned on training dataset, optimized valued were used in testing dataset to study the results. Model is developed in Python language.

Figure 4



	0	1	2	3	4	5	6	7
60	False	False	False	64	32	0.2	[5.915469955652952e-05, 0.007691209204494953]	[0.0004002423374913633, 0.020006056874990463]
44	False	True	False	64	32	0.2	[0.00015510663797613233, 0.012454181909561157]	[0.000572384218685329, 0.023924551904201508]
61	False	False	False	64	32	0.3	[7.301518053282052e-05, 0.008544892072677612]	[0.0009364760480821133, 0.030601896345615387]
59	False	False	False	32	64	0.3	[7.104186079232022e-05, 0.008428633213043213]	[0.001219108700752583, 0.0349157378077507]
15	True	True	False	64	64	0.3	[0.0002089772606268525, 0.014456045813858509]	[0.001267948653548956, 0.03560826554894447]
...	...	...	...	...	...	...	...	...
1	True	True	True	32	32	0.3	[0.00014819222269579768, 0.012173422612249851]	[0.01911289617419243, 0.13824939727783203]
33	False	True	True	32	32	0.3	[0.00038249173667281866, 0.019557395949959755]	[0.02062704786658287, 0.14362119138240814]
9	True	True	False	32	32	0.3	[0.00024587850202806294, 0.01568051427602768]	[0.020784365013241768, 0.14416784048080444]
2	True	True	True	32	64	0.2	[0.0004342805768828839, 0.020839398726820946]	[0.021442415192723274, 0.1464322954416275]
37	False	True	True	64	32	0.3	[0.0006179570336826146, 0.024858741089701653]	[0.021872442215681076, 0.14789335429668427]

64 rows × 8 columns

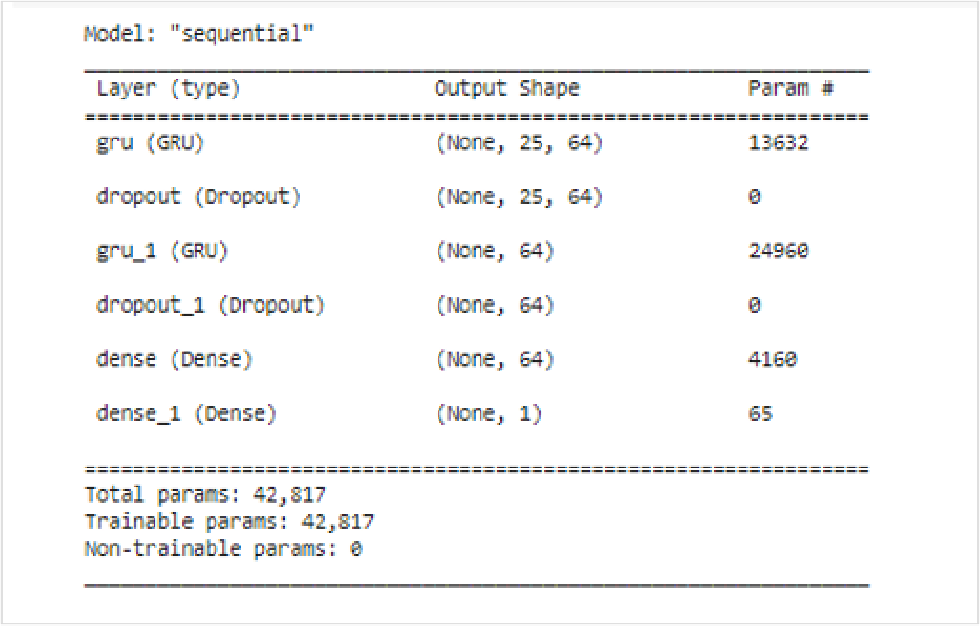
```
Best Combination:
 first_additional_layer = False
 second_additional_layer = False
 third_additional_layer = False
 n_neurons = 64
 n_batch_size = 32
 dropout = 0.2
*************************
Results Before Tunning:
 Test Set RMSE: 0.0803

Results After Tunning:
 Test Set RMSE: 0.02

75.0% Improvement
```

Figure 4 Hyper parameter tuning of hidden layers, neurons, batch size and drop out

Figure 5



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru (GRU)                   (None, 25, 64)            13632

 dropout (Dropout)           (None, 25, 64)            0

 gru_1 (GRU)                 (None, 64)                24960

 dropout_1 (Dropout)         (None, 64)                0

 dense (Dense)               (None, 64)                4160

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 42,817
Trainable params: 42,817
Non-trainable params: 0
_________________________________________________________________
```

Figure 5 Summary of purposed model

6. RESULT AND DISCUSSION

The hyper parameter tuning algorithm was applied on the google stock price dataset. The obtained values of hyperparameters were applied on other four stock price dataset, to check efficiency of the proposed hyper parameter tuned GRU model.

The accuracy of the model was calculated by r2 score and loss was measured by Root mean square error (RMSE). Results showed that the model performed well with every dataset. As shown in the table dataset 2 and 3 performed bit less as compared to the other dataset, though the results were good enough. The number of days was taken less in dataset 2 and 3, as compared to other datasets. Due to the less data fed to the model, performance of the model was decreased.

Table 1

Table 1 Predicted stock price of test dataset

	RMSE	r^2 score
Dataset 1	0.0015	0.9543
Dataset 2	0.0276	0.9488
Dataset 3	0.055	0.8996
Dataset 4	0.0466	0.8104
Dataset 5	0.0237	0.9812

Figure 6

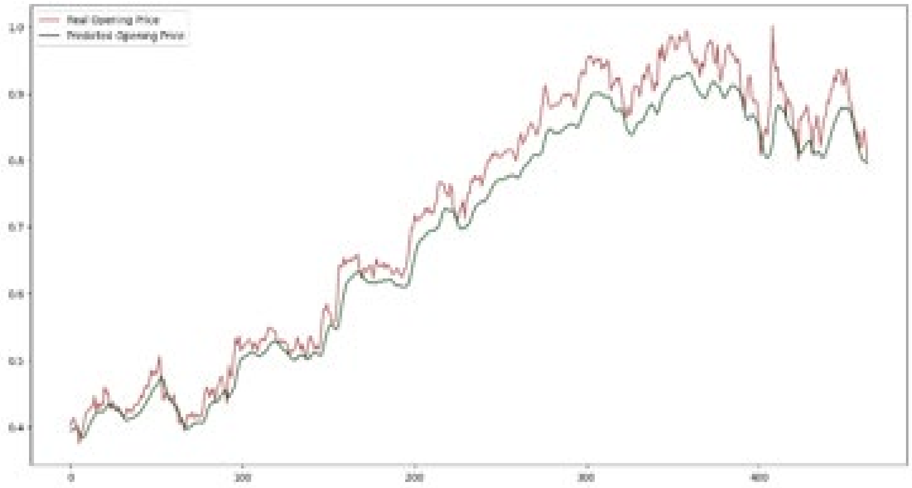


Figure 6 Predicted & actual price of Dataset 1

Figure 7

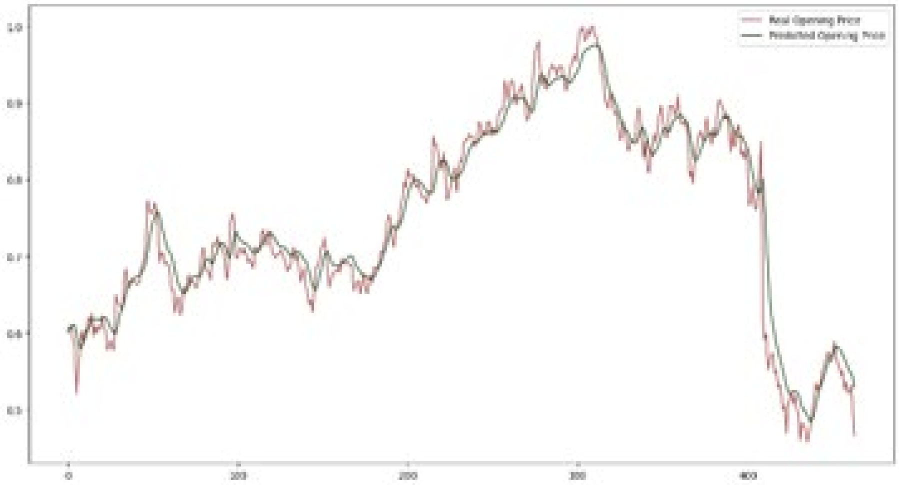


Figure 7 Predicted & actual price of Dataset 2

Figure 8

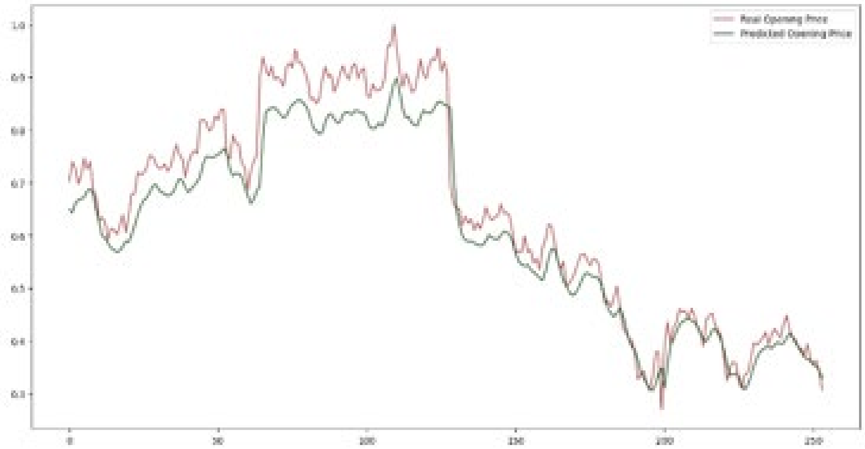


Figure 8 Predicted & actual price of Dataset 3

Figure 9

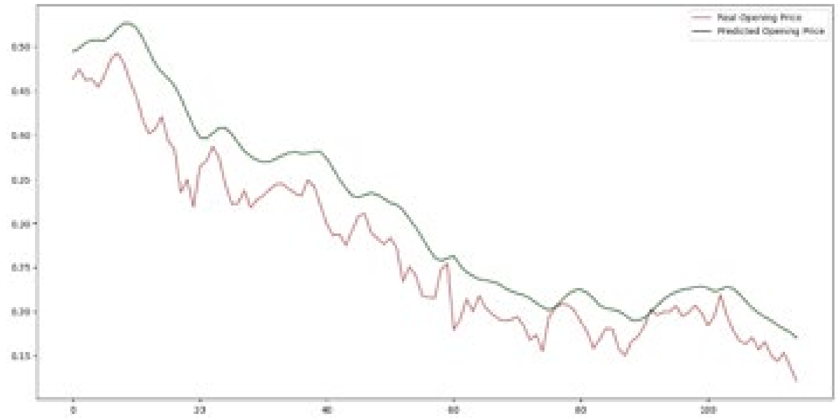


Figure 9 Predicted & actual price of Dataset 4

Figure 10

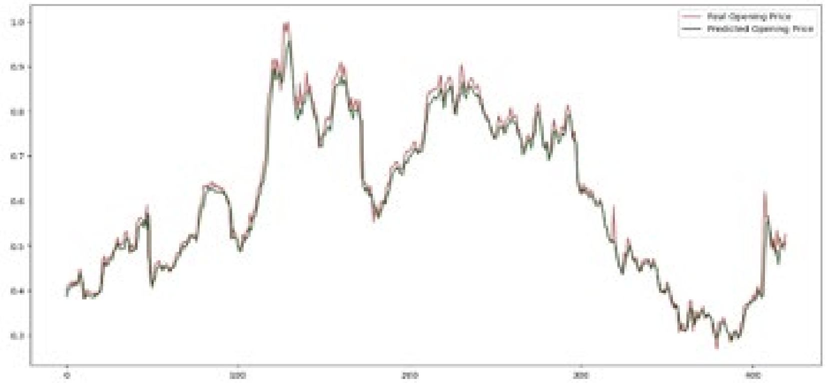


Figure 10 Predicted & actual price of Dataset 5

7. CONCLUSION

Here, stock price prediction for social media companies was carried out with the help of Deep Learning GRU model. Grid search algorithm was used to tune the hyper parameters for the GRU model. The benefit of hyper parameter tuning is to avoid the manual trial and error method. Grid search will suggest optimized hyper parameter values for model. The reason for selecting GRU was due to its faster computational efficiency. The model was developed to study the market behaviour on social media companies. The results showed that companies with a smaller amount of data performed little less as compared to the companies with large amount of data. GRU model overall achieved good performance, which shows the ability of model to perform well even with less amount of dataset. Future work can be carried out using different evolutionary algorithms for hyper parameter tuning.

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

Assaf, R. & Kolasani, S. (2020). Predicting Stock Movement Using Sentiment Analysis of Twitter Feed with Neural Networks. Journal of Data Analysis and Information Processing, 8, 309-319. https://doi.org/10.4236/jdaip.2020.84018

Baheti, R. Shirkande, G. Bodake, S. Deokar, J. & Archana, K. (2021). Stock Market Analysis from Social Media and News using Machine Learning Techniques. International, Journal on Data Science and Machine Learning with Applications, 1(1), 59-67.

Bhavsar, S. & Gor, R. (2022). Comparison of Back propagation algorithms : Bidirectional GRU and Genetic Deep Neural Network for Churn Customer (In press). International Organization of Scientific Research Journal of Computer Engineering (IOSR-JCE).

Bhavsar, S. & Gor, R. (2022). Predicting Restaurant Ratings using Back Propagation Algorithm. International Organization of Scientific Research journal of Applied Mathematics (IOSR-JM), 18(2), 5-9.

Cho, K. Merrienboer, B. V. Bahdanau, D. & Bengio, Y. (2014). On the Properties of Neural Machine Translation : Encoder-Decoder Approaches.

Das, S. Mishraa, D. & Rout, M. (2019). Stock market prediction using Firefly algorithm with evolutionary framework optimized feature reduction for OSELM method. Expert Systems with Applications, 1-24. https://doi.org/10.1016/j.eswax.2019.100016

Dozat, T. (2015). Incorporating Nesterov Momentum into Adam.

Fama, E. F. (1970). Efficient capital markets : a review of theory and empirical work. 5e Journal of Finance, 25(2), 383-417. https://doi.org/10.2307/2325486

Gao, Y. Wang, R. & Zhou, E. (2021). Stock Prediction Based on Optimized LSTM and GRU Models. Hindawi Scientific Programming, 8. https://doi.org/10.1155/2021/4055281

Ghosh, M. & Gor, R. (2022). Ad-Campaign Analysis and Sales prediction using K-means Clustering and Random Forest Regressor. International Organization of Scientific Research Journal of Applied Mathematics, 18(2), 10-14.

Guan, C. Liu, W. & Cheng, J. C. (2021). Using Social Media to Predict the Stock Market Crash and Rebound amid the Pandemic : The Digital 'Haves' and 'Have mores'. Annals of Data Science, 9(1), 5-31. https://doi.org/10.1007/s40745-021-00353-w

Kingma, D. & Ba, J. (2015). Adam : A Method for Stochastic Optimization. 3rd International Conference for Learning Representations, San Diego.

Mehta, P. Pandya, S. & Kotecha, K. (2021). Harvesting social media sentiment analysis to enhance stock market prediction using deep learning. PeerJ Computer Science. https://doi.org/10.7717/peerj-cs.476

Pengfei, Y. & Yan, X. (2019). Stock price predictions based on deep neural networks, Neural computing and applications. 1609-1628. https://doi.org/10.1007/s00521-019-04212-x

Polamuri, S. Srinivas, K. (2019). Stock Market Prices Prediction using Random Forest and Extra Tree Regression. International Journal of Recent Technology and Engineering (IJRTE), 8(3), 1224-1228. https://doi.org/10.35940/ijrte.C4314.098319

Rajpurohit, V. Bhavsar, S. & Gor, R. (2021). A comparision of GRU-based ETH price prediction. Proceeding of International Conference on Mathemaitcal Modelling and Simulation in Physical Sciences (MMSPS-2021), 424-431.

Raut, A. & Sethia, P. (2018). Application of LSTM, GRU and ICA for Stock Price Prediction. Proceedings of ICTIS, 2, 1-10. https://doi.org/10.1007/978-981-13-1747-7_46

Selvamuthu, D. Vineet, K. & Mishra, A. (2019). Indian Stock Market prediction using Artificial Neural Networks on tick data. Financial Innovation, 16(5), 1-12. https://doi.org/10.1186/s40854-019-0131-7

Shahi, T. Shrestha, A. Neupan, A. & Guo, W. (2020). Stock Price Forecasting with Deep Learning : A Comparative Study. Mdpi journal, 8, 1-15. https://doi.org/10.3390/math8091441

Shena, G. Tana, Q. Zhanga, H. & Ze, P. (2018). Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions. 8th International Congress of Information and Communication Technology, 895-903. https://doi.org/10.1016/j.procs.2018.04.298

Soni, P. Tewari, Y. & Krishnan, D. (2022). Machine Learning Approaches in Stock Price, Prediction : A Systematic Review. Journal of Physics : Conference Series. https://doi.org/10.1088/1742-6596/2161/1/012065

Srivinay, Manujakshi, B. C. Kabadi, M. G. & Naik, N. (2022). A hybrid stock price prediction model based on PRE and DeepNeural Network. MDPI. https://doi.org/10.3390/data7050051

Touzani, Y. & Douzi, K. (2021). An LSTM and GRU based trading strategy adapted to the Moroccan market, Journal of Big data, 1-16. https://doi.org/10.1186/s40537-021-00512-z