UNIVERSITY OF INFORMATION TECHNOLOGY

FACULTY OF INFORMATION SYSTEMS

Predicting Stock Price of Recurrent Neural Network Variant, Multivariable Linear Regression and Multivariable Non-Linear Regression

- HÀ GIA HUY - 20521385
- ĐẬU ĐÌNH QUANG ANH - 20521059
- TRẦN ĐỨC DUY - 20521248

Predicting Stock Price of Recurrent Neural Network Variants, Multivariable Linear Regression and Multivariable Non-Linear Regression

1st Ha G. Huy
STAT3013.N12.CTTT – EN
University of Information Technology
20521385@gm.uit.edu.vn

2nd Dau Dinh Q. Anh
STAT3013.N12.CTTT – EN
University of Information Technology
20521059@gm.uit.edu.vn

3rd Tran D. Duy
STAT3013.N12.CTTT – EN
University of Information Technology
20521248@gm.uit.edu.vn

Abstract – Deep learning algorithms have been implemented in several application domains and as such new algorithms have been developed to improve the performance. This paper presents a comparative study of three Recurrent Neural Networks including Long Short-Term Memory with Attention Mechanism, Bidirectional LSTM and Gated Recurrent Unit, on the accuracy aspect. The RNN Models are applied by Relu activation function on different data sub-set based on the different ratio of Training, Testing and Validation Dataset. The Multivariable Linear Regression is implemented by the existing function in Sklearn Library while he Multivariable Non-Linear Regression is implemented based on Random Forest Regression algorithm. The study showed that RNN Models is the most powerful for Time-Series Dataset while Multivariable Non-Linear Regression is showing a good result.

I. INTRODUCTION

Recurrent Neural Network (RNN) models are the most effective sequence model and they have been applied in many applications with temporal or sequential data. For performance improvement and specific purposes, people have developed many variants of RNN algorithms including Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM), Long Short-Term Memory with Attention Mechanism (A-LSTM) and Gated Recurrent Unit (GRU). RNN variants have been applied successfully in sequence application. For example, people have been used extensively in polyphonic music processing, namely speech signal processing [1], natural language processing [2], and sequence generation [3].

The main element leads to the success of them is the gating network signals that is used to control the current input values and old memory to update the current activation neural layer [4]. The set of weights in Gated RNN can be changed after each time step.

Multivariable Linear Regression (MLR) and Multivariable Non-linear Regression (MNLR) are the primitive algorithms presenting simple functions. Regression Models are mostly used for functional Data Analyst. By D. Nguyen et al in 2011, Linear Regression Model has been used for Author age prediction via text dataset. In 2008, KW. Lau et al, have implemented Non-Linear Regression Model for Local Prediction on complex time series.

In this paper, we focus on the RNN models, Multivariable Linear Regression (MLR) model and Multivariable Non-Linear Regression on a public dataset. Using the VNM dataset, we will predict the stock value after each day based on the “Close” column. The observational data will be put in a 988 x 1 matrix. In next section, we will introduce to some related research about Model Comparation and application of RNN Models in real-world. A short review of Models’ architectures and the dataset’s source will be also showed in the paper for the closer look of Models’ mechanic. We will apply Relu activation function in to RNN Models and take the results recorded for the summary at the last part. RNN model will be used for predict the value in next one month and compare to the actual situation.

II. RELATED WORK

Since RNN is introduced in 1986, there was many research papers about this concept and its variant. In this section, some related research about using RNN and Regression Models to predict Time Series Data.

M. Ravanelli at al. [1] revised Gated Neural Network for ASR. As the results, the authors developed the traditional GRU design to become a more efficiency Gated-model.

Nana Liu [2] implemented Bidirectional Gated Neural Network for Music Emotion Recognition and compare to other models. At last, the BiGRU model can identify well Sad and Happy music.

A. Graves [4] showed that the power of a RNN Model to predict text and online handwritting just by processing a data point at a time.

C. Wang et al. [5] revised LSTM with attention mechanism for propose a consumer credit scoring method by online operation behavior data. They treat each event in the dataset as a word and put the events into a vector as a sentence.

S. Yang et al. [6] conducted a comparative study of tradition LSTM and Gated-Recurrent Unit. The research revealed that the GRU Model is faster than LSTM in training data. The GRU Model will be also more efficiency in processing a long paragraph and small dataset. This led to the cost of GRU will be higher than LSTM.

K. Cho et al. [7] used RNN Encoder, RNN Decoder and Gated Recursive Convolutional Neural Network. The results showed that the Models performs well one short sentences without unseen data. But in longer sentences with more unseen data, the Models will not preform well and rapidly degraded.

G. Chrupala et al [8] consist two Gated Recurrent Unit networks in a model. The authors used the context of sentences to visualize the context and predict the next work in the sentences.

J. Chung at al. [9] revealed that a GRU RNN Model is comparable to LSMT Model. Showing that GRU Model is also one of the best model in Time Series by evaluate the models on polyphonic music and speech signal.

Regression Models are the very simple models and they are one of a very friendly concepts of Deep Learning. A regression Model provides a function for showing the relationship between the independent values known as the features and a target value. The models will use the features as the input of the functions to predict the target value.

I. Chenini et al. [10] successfully analyzed the ground-water quality and the chemicals contained in the water by using Multiple Linear Regression. The study's result provided methods to identify ground-water quality and model the hydrochemical data.

Khademi F. et al [12] show in their study that the Multiple Linear Regression Model to preliminary mix design of concrete by predicting compressive strength of concrete in 28 days.

III. MODEL ARCHITECTURES

A. *Recurrent Neural Network*

Sequential data types are typically processed by RNN. The RNN models have a recurrent hidden state as in

$$h_t = g(Wx_t + Uh_{t-1} + b)$$

where x_t is an *m-dimension* input vector at the current time (*t*), g is the *activation function*, such as logistic function, or the Rectified Linear Unit (ReLU) [2, 12]. W, U *and* b are defined as sized parameters where W is a *n* x *m* matrix, U is an *n* x *n* matrix, and b is a vector with *n* elements. These sized parameters in this case are treated as two weights and one bias.

Tsungnan Lin at al. [13] showed that the gradients may vanish or explode after a number of timesteps if use such as a simple RNN. In [2-4], the idea of using some variants of RNN (LSTM and GRU) to solve the problem. We will present these two models below in details for our purposes.

B. *Long Short-Term Memory (LSTM)*

S. Hochreiter at al. [14] showed the idea of making a path that can let the gradient flow for a long time. F.A. Gers at al. [15] revealed that a crucial improve of LSTM model is make the *weight parameters* can be adaptively change through each time step by making *gated self-loop* to control the *weights* and the time for each integration can be adapted dynamically. The time in each integration can be also changed by *fixed parameters (weights)* with a set of suitable input values since the output of the model is the time for each time step.

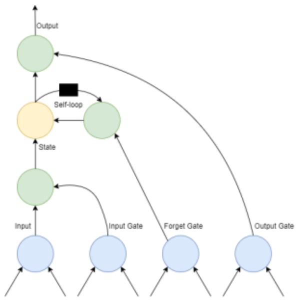


Figure 1. Diagram of a "cell" in LSTM model. Each cell connects with each other instead of regular hidden unit as in Graph Neural Network. Each input features are the result of a regular neural unit. The values can be stored in the State Unit of the cell. The State Unit has a linear self-loop with the weights are controlled by Forget Gate. All the Gate-Units can use non-linear sigmoid function, while the Input can use any non-linear function. The black square is the delay of each time step.

The abstract of a cell of LSTM model show in the *Figure 1*.

LSTM RNN use "LSTM cells" that have a self-loop. Each cell has the input and output values like a primitive recurrent network with more parameter and use *a system of gates* to control the information flow. As mentioned in the previous part, the *parameters (weigths)* of the self-loop are controlled by the *Forgot Gate* $f_i^{(t)}$(time step t and cell i) to calculate the parameter within 0 to 1 using Sigmoid function.

$$f_i^{(t)} = \sigma(b_i^f + \sum_j U_{i,j}^f \, x_i^{(t)} + \sum_j W_{i,j}^f \, h_i^{(t-1)})$$

where $x^{(t)}$ is the current input vector and $h^{(t)}$ is the current vector in hidden layer, contains the output values of the cell. b^f, U^f *and* W^f is the *bias value, weight* of input and *recurrent weight* of forget gate. The LSTM internal state is updated as follows:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma(b_i + \sum_j U_{i,j} x_i^{(t)} + \sum_j W_{i,j} h_j^{(t-1)})$$

where $g_i^{(t)}$ is calculated like the forget gate but with its own parameter

$$g_i^{(t)} = \sigma(b_i^g + \sum_j U_{i,j}^g \, x_i^{(t)} + \sum_j W_{i,j}^g \, h_i^{(t-1)})$$

The output $h_i^{(t)}$can be shut off by the output gate $q_i^{(t)}$ and can be calculated by:

$$h_i^{(t)} = \tanh(s_i^{(t)}) q_i^{(t)}$$

$$q_i^{(t)} = \sigma(b_i^q + \sum_j U_{i,j}^q \, x_i^{(t)} + \sum_j W_{i,j}^q \, h_i^{(t-1)})$$

C. *Gated Recurrent Unit (GRU) RNN*

Get the idea from LSTM architecture, Gated Recurrent Unit inherited some its necessary features. Answered in [7, 8], the main difference between LSTM and GRU is that GRU have a single gating unit to update the state unit and control the forgetting factor.

By J. Chung at al. [9], the result showed that GRU RNN are much more advance than LSTM in most cases. There is a variant of GRU RNN, e.g. the Minimal Gated Unit (MGU) RNN which only use one gate equation and give the compatible performance (in some cases) to the LSTM RNN.

The update equations:

$$h_i^{(t)} = u_i^{(t-1)} h_i^{(t-1)} + (1 - u_i^{(t-1)}) \sigma(b_i + \sum_j U_{i,j} x_j^{(t-1)} + \sum_j W_{i,j} r_j^{(t-1)} h_j^{(t-1)})$$

where u is update gate and r is reset gate. Update gate value is followed by the equation:

$$u_i^{(t)} = \sigma(b_i^u + \sum_j U_{i,j}^u \, x_i^{(t)} + \sum_j W_{i,j}^u \, h_i^{(t)})$$

and the reset gate value equation is:

$$r_i^{(t)} = \sigma(b_i^r + \sum_j U_{i,j}^r \, x_i^{(t)} + \sum_j W_{i,j}^r \, h_i^{(t)})$$

In this paper, we will focus on GRU RNN only and compare with the very basic Linear Regression model.

D. *Bidirectional LSTM*

In the traditional RNNs, the state at time t only captures the data from the past $x^1, \dots, x^{t-1}$ and the current input x^t. However, in real world situations, many applications are also need *the whole input sequence* for the prediction y^t. For instance, in speech recognition, the correct interpretation of the current word may depend on the next few words because the context of a sentences can be decided by a totally random word.

People have implemented successfully [16] in many applications such as handwriting recognition [17] and bioinformatics [18].

In figure 2, a basic Bidirectional RNN is described by the combination of a sub-RNN that move from the beginning and a sub-RNN from the end of the sequence. The h^t is stand for the RNN that move forward and the g^t is stand for the RNN that move backward.

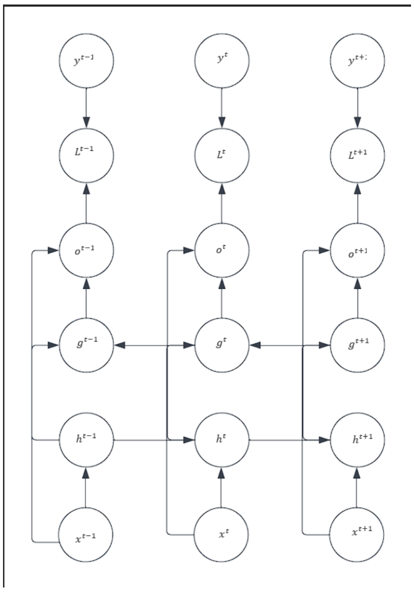


Figure 2. Bidirectional LST

E. Attention Mechanism

In Natural Language Processing, capturing the sematic of a very long sentence is very hard. There is an efficiency approach to this problem is *Attention Mechanism*. By this method, the RNN Models can read the whole sentence to get the general context and encode the words one at a time, each time focus on a specific part of the input sentence to predict the next word in the output sentence.

In figure 3 describes an abstract view of *Attention Mechanism* introduced by [19]. h^t is the state of a RNN Model at time t and a^t is the attention function at time t. People take the average weight of $weights\ h^t$ and $weights\ a^t$ to form the $vector\ c$ as a context vector.

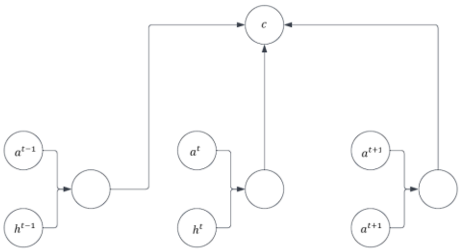


Figure 3. LSTM with Attention

F. Multivariable Linear Regression (MLR)

Multivariable Linear Regression (MLR) models have been used in various area to predict or classify data. MLR is a very basic model that take a vector $x \in R^n$ containing input values and vector $w \in R^n$ containing the weight values of each feature. We can define the output value by the following formula:

$$\hat{y} = w^T x$$

where $\hat{y}$ is the predicted value at the output.

In this case, w_i is a parameter(weight) to multiply with feature x_i and sum up all the value from feature x_i to get the predicted value ($\hat{y}$). Each weight in a MLR model is show that how importance of the data. If the weight is a positive number, increasing the value of its feature will also increase the predicted value. The predicted value in output neural will also decrease if the feature's value is increased while the weight is negative. This also concluded that if the weight is *zero*, the feature is mean nothing in our model.

In real-world situation, the MLR is sometimes more complicated with *intercept parameter (bias):*

$$\hat{y} = w^T x + b$$

where b is the bias value.

This will keep our model is still described as a straight line but does not need to go through the *origin point.*

Linear regression, basically, is a very basic algorithm and has a lot of disadvantages. But it will give us an overall view of *Machine learning algorithm.*

G. Multivariable Non-Linear Regression (MNLR)

Multivariable Non-Linear Regression is a version of regression analysis. The observational data are put in a non-linear model. We can define the output of the model by the following formula:

$$y \sim f(x, b)$$

where y is the output value. The function f is nonlinear and takes x and b as the vector of independent variables and bias values respectively.

IV. DATA PREPARATION

A. Data

The research is conducted on the VanEck Vietnam ETF (VNM) Dataset recored from 31/12/2018 to 30/11/2022 of 988 observation.

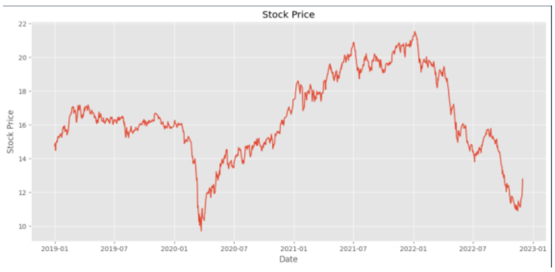


Figure 4. Dataset recorded

B. Data preparation

We chose the "Close" Columns in the dataset as the feature of the Models. The dataset is split into three sub-set following the ratio of training set, testing set and validation set: 80/10/10, 70/20/10 and 75/15/10 of RNN Model. In Regression Models, we also split the dataset into three sub-set of training and testing set as 80/20, 70/30 and 75/25.

V. Model settings

A. RNN

In this task, we built three RNN models with three layers. The first RNN Model is applied the Gated Recurrent Unit in each layer, the second RNN Model is a standard Bidirectional LSTM and the last one is a basic LSTM model with Attention Mechanism. The hyperparameters is showing in the table 1

Table 1. Hyperparameter

Hyperparameter	Values
Learning Rate	0.0001
Dropout	20%
Hidden Unit	64
Batch size	32
Optimizer	Adam
Epoch	100
Activation Function	Relu

B. *Regression Models*

In Multivariable Linear Regression Model, we use the Sklearn Library for the LinearRegression() function to auto fit the Model. The Multivariable Non-Linear Regression Model is built from the Random Forest Regression algorithm in Sklearn Library.

V. RESULT AND CONCLUSION

A. *RESULT*

TABLE 2. EVALUATE METRIC

Model	Ratio	MAPE	RMSE	MAE
GRU	80/10/10	14.0%	0.3	0.23
	70/20/10	19.0%	0.3	0.24
	75/15/10	13.5%	0.3	0.24
A-LSTM	80/10/10	14.5%	0.29	0.21
	70/20/10	18.9%	0.32	0.25
	75/15/10	13.6%	0.31	0.24
B-LSTM	80/10/10	14.3%	0.26	0.2
	70/20/10	19.2%	0.3	0.23
	75/15/10	13.7%	0.27	0.2
MLR	80/20	1.40%	0.28	0.21
	70/30	1.40%	0.28	0.21
	75/25	1.30%	0.27	0.21
MNLR	80/20	19.20%	0.3	0.23
	70/30	22.20%	0.3	0.2
	75/25	22.10%	0.34	0.25

The table 2 is showing the Loss values including MAPE, RMSE and MAE scores. The rows that were highlighted by the red color is determined as the best sub-set applied in the specific model.

Experiment revealed that RNN Models are comparative when gave the difference from the benchmark scores is close to each other's and the 75/15/10 sub-set is applied well.

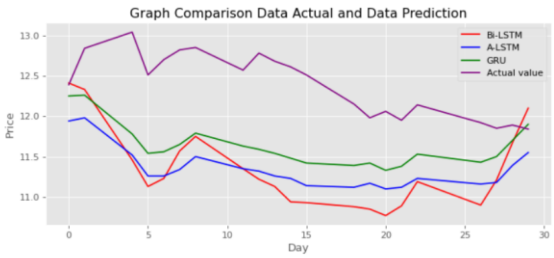
The Multivariable Linear Regression is not showing a big different between the sub-set but the Overfitting is seem to be appear on this model.

The multivariable Non-Linear Regression Model mostly efficiency on the 80/20 data sub-set. But the MAPE score of three data sub-set is around 20% and cannot be comparative to RNN Models

The RNN Models are used to predict the stock price in next 30 after applying unseen data. The result will return the stock price of VanEck Vietnam ETF from 12/1/2022 to 12/30/2022.

The results are showed in the figure 5.

FIGURE 5



The predicted results showed a good accuracy on three RNN Models. The lines followed the trend of the actual value and the difference of the predicted and the actual value is in a safe range.

B. CONCLUSION

The study showed that RNN Models are one of the best models for Time Series Dataset. They can be implemented well on complicated dataset which including Stock Price prediction, Natural Language Processing. Multivariable Non-Linear Regression show a good potential in prediction on Random Forest Regression algorithm but cannot be comparative to the RNN Models. Multivariable Linear Regression is very simple built, it is not suitable for applying to complex dataset.

REFERENCE

[1] M Ravanelli, "Light Gated Recurrent Units for Speech Recognition", arXiv:1803.10225, 2019

[2] Nana Niu, "Music Emotion Recognition Model Using Gated Recurrent Unit Networks and Multi-Feature Extraction", Mobile Information Systems, vol. 2022, Article ID 5732687, 11 pages, 2022. https://doi.org/10.1155/2022/5732687

[3] Alex Graves, “Generating Sequences With Recurrent Neural Networks”, arXiv:1308.0850, 2014.

[4] Ian Fellow, “Deep Learning Book”, pg 407-410, 2015.

[5] C. Wang, D. Han, Q. Liu and S. Luo, "A Deep Learning Approach for Credit Scoring of Peer-to-Peer Lending Using Attention Mechanism LSTM," in IEEE Access, vol. 7, pp. 2161-2168, 2019, doi: 10.1109/ACCESS.2018.2887138.

[6] S. Yang, X. Yu and Y. Zhou, "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example," 2020 International Workshop on Electronic Communication and Artificial Intelligence

[7] K. Cho, B. v. MerrienBoer, D. Bahdanau and Y. Bengio, “On the Properties of Neural Machine Translation: Encoder-Decoder Approaches”, arXiv:1409.1259, 2014

[8] Grzegorz Chrupała, Ákos Kádár, Afra Alishahi, “Learning language through pictures”, arXiv:1506.03694, 2015

[9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”, arXiv:1412.3555, 2014

[10] Chenini, I., Khemiri, S. Evaluation of ground water quality using multiple linear regression and structural equation modeling. Int. J. Environ. Sci. Technol. 6, 509–519 (2009). https://doi.org/10.1007/BF03326090

[11] Khademi F., Behfarnia K., “Evaluation of concrete compressive strength using artificial neural network and multiple linear regression models, 2016

[12] Quoc V. Le, Navdeep Jaitly, Geoffrey E. Hinton, “A Simple Way to Initialize Recurrent Networks of Rectified Linear Units”, arXiv:1504.00941, 2015

[13] Tsungnan Lin, Bill G. Horne, Peter Tino, C. Lee Giles, “Learning long--term dependencies is not as difficult with NARX recurrent neural networks”, 2001

[14] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997 Nov 15;9(8):1735-80. doi: 10.1162/neco.1997.9.8.1735. PMID: 9377276.

[15] F. A. Gers, J. Schmidhuber and F. Cummins, "Learning to forget: continual prediction with LSTM," 1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470), 1999, pp. 850-855 vol.2, doi: 10.1049/cp:19991218.

[16] Alex Graves, “Supervised Sequence Labelling with Recurrent Neural Networks”, 2012

[17] Alex Graves, Jürgen Schmidhuber, “Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks”, 2018

[18] Pierre Baldi, Kurt Hornik, “Neural networks and principal component analysis: Learning from examples without local minima”, Volume 2, Issue 1, Pages 53-58, ISSN 0893-6080, 1989, https://doi.org/10.1016/0893-6080(89)90014-2.

[19] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate”, arXiv:1409.0473, 2016