Comparisons of Stock Prediction Methods Based on Recurrent Neural Networks

George Shao*

Interlake High School, Bellevue, Washington, 98007, United States

* Corresponding Author Email: s-shaog@bsd405.org

Abstract. These instructions give you guidelines for preparing papers for DRP. Use this document as a template if you are using Microsoft Word 6.0 or later. Otherwise, use this document as an instruction set. The electronic file of your paper will be formatted further at DRP. Paper titles should be written in uppercase and lowercase letters, not all uppercase. Avoid writing long formulas with subscripts in the title; short formulas that identify the elements are fine (e.g., "Nd-Fe-B"). Do not write "(Invited)" in the title. Full names of authors are preferred in the author field, but are not required. Put a space between authors' initials. The abstract must be a concise yet comprehensive reflection of what is in your article. In particular, the abstract must be self-contained, without abbreviations, footnotes, or references. It should be a microcosm of the full article. The abstract must be between 100 - 300 words. Be sure that you adhere to these limits; otherwise, you will need to edit your abstract accordingly. The abstract must be written as one paragraph, and should not contain displayed mathematical equations or tabular material. The abstract should include three or four different keywords or phrases, as this will help readers to find it. It is important to avoid over-repetition of such phrases as this can result in a page being rejected by search engines. Ensure that your abstract reads well and is grammatically correct.

Keywords: Enter key words or phrases in alphabetical order, separated by commas.

1. Introduction

Playing the stock market has long been a source of opportunity that has large monetary benefits, even possibly sustaining various livelihoods. Thus, it has been a common endeavor to try to predict the stock market through the use of various methods. However, the market is notoriously volatile, and as a reflection of the economics of the world it is susceptible to unpredictable and extreme rises and drops [1]. Factors ranging from a business' own assets and liabilities to global socioeconomic factors makes this challenge interesting and difficult from a technological viewpoint, and consistently being able to predict the charts would provide a huge financial gain. Thus, it is no wonder that machine learning has been applied to the problem in an attempt to solve it [2]. When taking in as data, the market can be perceived as a series of values over a period of time, lending itself most well to the style of Recurrent Neural Networks (RNN) [3]. Of these RNNs, the most commonly applied towards this problem is the LSTM method [4], as its ability to keep track of subtle patterns across years of data gives it a better prediction accuracy [5]. Naturally, the closer the predictions are towards the actual data, the better the returns can be.

However, the LSTM method has long had multiple shortcomings that inhibit its ability to predict the graph, especially in the long term. Despite its unique structure that makes it able to process information in the long term, LSTMs still suffer from the vanishing gradient problem [6], giving a limit to its processing capabilities. Especially in datasets like stocks that span across several years, older data becomes exponentially likely to get filtered out and forgotten [7]. This information must be fed through the series of gates over and over again, which likely could result in the loss of vital information. It is this same bogging down of data that makes LSTMs less flexible and volatile. Due to the steps required for an LSTM to process the data, often times it is usually unable to adequately keep up with the rapid changes that stock market graphs often produce.

In contrast to this, a relatively new application of the ESN method has caused it to become a candidate for use in the attempt to make stock market predictions [8] Unlike the LSTM, ESNs are much more promising in terms of adapting to the turbulence [9] of the graphs. Through the unique

reservoir layer [10], the method can balance between the retained long-term information and the abrupt changes in the values of the market. Additionally, the ESN may not face the same issues as the LSTM in regards to losing information, as data within the reservoir layer is not constantly filtered through more and more gates like an LSTM. Instead, the information is allowed to "echo" and influence the output, lending itself well to this task. Thus, given the right hyperparameters, it may be possible for this method to outperform the standard LSTM.

2. Methodology

2.1. Data

The dataset used in this study contains the opening market prices of the stocks of Amazon, Bitcoin, Netflix, and Domino's Pizza from May 2013 to May 2019 [11]. This data is accessed as four different sets of time series data, all showing the signs of upward growth with varying degrees of volatility.

2.2. RNN Model

In order to predict the data, the two RNN algorithms, LSTM and ESN are implemented, following the idea that recurrent neural networks are able to more efficiently process sequential data. Thus, RNNs should be better at predicting time series data and long-term trends, and extrapolating upon them.

2.2.1 LSTM

The LSTM is a unique form of an RNN that includes the unique inclusion of gates. The gates of an LSTM allow for it to process and store certain information that the algorithm deems important. It does this through 3 types of gates, the "forget gate", the "input gate" and the "update gate" [12]. Through a sigmoid function, the forget gate decides on what to keep and discard from the previous generation. The data can then be selectively updated through the input gate, quantifying the importance of the newly introduced data. Finally, the update gate takes the information processed to make a prediction forward. This is then compared to the test data, and used to further train the model. As a result, the LSTM is capable of retaining the important long-term trends in time series data, which is better than a standard RNN [13], making it a commonly used method in stock predictions [14]. In its implementation, Keras TensorFlow [15] offers a robust implementation of multiple RNNs, including LSTM. Thus, implementing the algorithm per dataset, and then adjusting for hyperparameters would lead to the ideal result. After finding the best set of hyperparameters, the MSE is taken for the prediction and recorded.

2.2.2 ESN

The ESN method is another approach to address the issues posed by standard RNNs. This comes in the form of the unique layer structure that is commonly known as the "reservoir" of an ESN [9]. In an attempt to address the same issue of standard RNNs being unable to retain long term information, the reservoir of an ESN acts as a convoluted network of weights and neurons rather than set layers. The input signal is connected to the trainable reservoir and output node, thus creating the feedback loop of an RNN. However, since the reservoir tends to be quite loosely connected, much of the trained data still ends up "echoing" around the reservoir and influencing the data, hence its name. This method also makes it more sensitive to inputs in comparison to an LSTM, and has an easier time dealing with the vanishing gradient problem when encountering larger and larger datasets. There are few adequate methods of implementing an ESN algorithm, but a combination of the pyESN library [16] and manipulation of the dataset allows for training of the algorithm and its hyperparameters to each set of data. As with previous method, the MSE is recorded between the prediction and the test set after determining an ideal set of hyperparameters.

2.3. Experimentation

In order to determine the ideal hyperparameters, each algorithm is run and evaluated based on the randomly predetermined hyperparameters. Afterwards, the two most influential hyperparameters are chosen and adjusted on a certain range, and the best is chosen according to the MSE. For the LSTM, the two hyperparameters were the batch size and the dropout rate. For the ESN, the spectral radius and the noise were the two hyperparameters. Although this does not encompass all possible iterations of each algorithm, adjusting every hyperparameter across all ranges was too inefficient to produce results in a meaningful amount of time.

3. Results

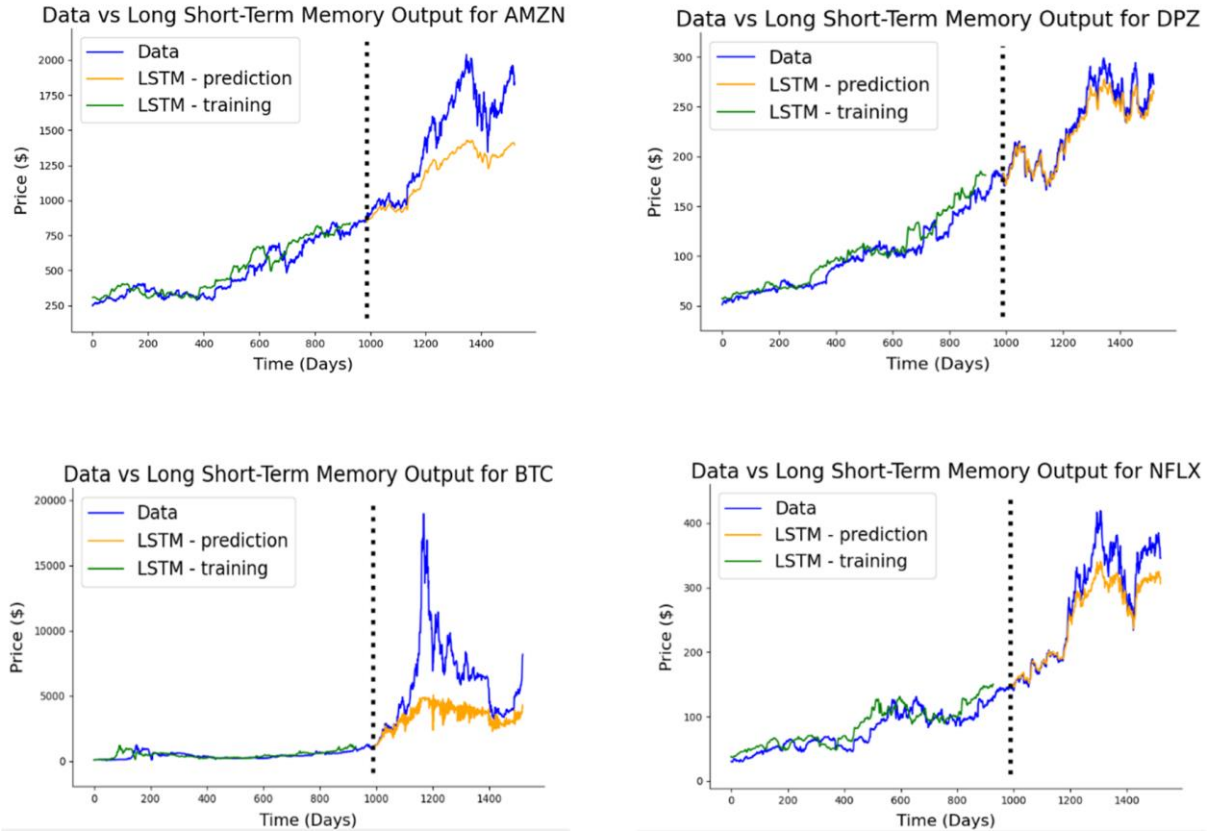


Figure 1. LSTM training and prediction results

Figure 1 illustrates the training and final predictions of the LSTM method with the datasets, including Amazon (AMZN), Domino's Pizza (DPZ), Bitcoin (BTC) and Netflix (NFLX). It begins the prediction after 1000 days, where the training finishes and the predictions begin. As predicted, the method was able to generally follow the trend of most datasets. However, it is immediately clear that there are significantly challenges faced by the method. The most notable is the inability to match the volatility of most of the graphs. Noticed most clearly with the Bitcoin graph, the LSTM consistently falls short of large jumps within the data, indicating underfitting thanks to the insufficient information being processed per step. With a large dataset like the one given, the LSTM also could face the vanishing gradient problem to an extent, with data being exponentially likely to get filtered out over more and more data points, indicated through the Netflix data. As the prediction progresses, it noticeably deviates from the test data, indicating possible loss of information. As predicted, the model did the best with the least volatile dataset, Domino's Pizza, and became incrementally less effective as the graphs became steeper and more unpredictable, to the point which the Bitcoin prediction was less than satisfactory.

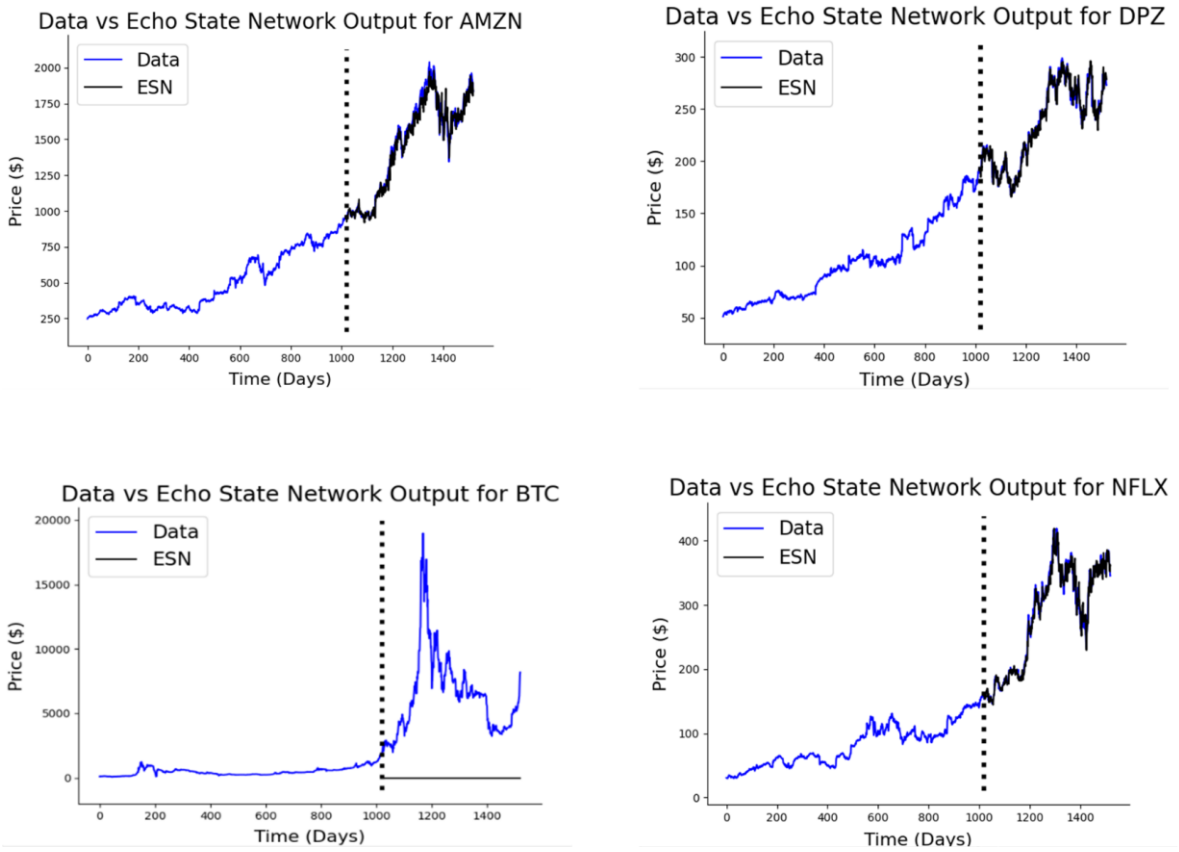


Figure 2. ESN prediction results

Figure 2 depicts the predictions of the ESN method as per each of the datasets from Amazon (AMZN), Domino's Pizza (DPZ), Bitcoin (BTC) and Netflix (NFLX). It begins the prediction after 1000 days, where the training finishes and the predictions begin. The ESN method noticeably outperforms the LSTM method in the majority of the graphs, clearly detailing the shapes of each set. The model immediately follows any changes in the data, and predicts the graphs with significantly better accuracy. In contrast to the LSTM method, the ESN might even face slight challenges with overfitting, overcorrecting in certain spots and making much larger jumps than necessary. The immediate outlier to notice is the Bitcoin graph, in which the model outright refuses to perform any sort of meaningful prediction. This could be caused by numerous reasons, such as the flat training set leading the rest of the graph to flatten out. However, the most likely reason is the previously mentioned overfitting, which leads the previous iterations of the prediction to massively overcorrect the massive spike within the graph. Considering the algorithm prioritizes the optimization of the mean standard error, all forms of overfitting may have resulted in larger errors than simply graphing a straight line. This may be a limitation of the method itself, or due to the lack of options presented by the hyperparameters. In general, the trend of the ESN predictions matched that of the LSTM predictions, doing better with less volatile graphs. Ultimately, this method proves to be more flexible and more responsive to the data in comparison to the LSTM.

Table.1. RNN model MSE per dataset

Dataset	LSTM	ESN
AMZN	4183.26	3083.38
DPZ	516.98	39.24
BTC	7255.81	13462.51
NFLX	812.39	131.92

Table 1 presents the specific information in regards to the error of both the LSTM and the ESN when predicting Amazon (AMZN), Domino's Pizza (DPZ), Bitcoin (BTC) and Netflix (NFLX). For

AMZN, DPZ and NFLX, the ESN manages to outperform the LSTM to varying degrees of success. However, the LSTM manages to have a better read on the BTC by a considerable margin. However, both methods perform the worst with the BTC graph, which suggests an outlier reason specific to the data itself.

4. Conclusions

Predicting the stock market has long been an intriguing and alluring problem that has been attempted through a multitude of different methods. From a machine learning perspective, the problem can be simplified into the task of predicting time series data through understanding intricate patterns that make up its values. However, most generalized RNNs are typically unable to match the task to an acceptable degree, to which certain variations on the model were created. Of these, the LSTM model has seen the most use in direct application to this challenge thanks to its ability to store long term information. However, this method faced shortcomings of its own, and a different model, the ESN, posed the possibility of outperforming an LSTM method. As such, the ESN and LSTM methods are trained between each of the multiple datasets and compared in order to determine the advantages that each ones hold. The ESN appears to track the data better than the LSTM, following the erratic data closely for the majority of the data. However, its dependence on its initial hyperparameters makes it not only harder to train, but also may force it to completely fail to process the given data.

Currently the application of ESNs in similar contexts has not been well documented, and thus it is difficult to reach a comprehensive conclusion on the advantages of the method. However, the ESN seems to vary in accuracy, given its predictions from the datasets. What remains a constant limitation, however, is the length of time required to train the data. Thanks to its sensitivity to the starting variables, training to find the optimal set of hyperparameters is incredibly inefficient and expensive. This, in combination with the significant uncertainty of the prediction based on the initial weights, leads to difficulties with gauging the true effectiveness of ESNs. The method does seem to be more scalable than LSTMs, and the results show that in general ESNs have a better performance, especially as the predictions lengthen in time. As it stands, while ESNs may offer superior predictions, they are heavily hindered by their lack of consistency, possibly making them a less attractive option overall.

References

[1] Jeff Stibel, 2014. Why We Can't Predict Financial Markets. https://hbr.org/2009/01/why-we-cant-predict-financial.

[2] Jingyi Shen, Shafiq M. Omair, 2020. Short-Term Stock Market Price Trend Prediction Using a Comprehensive Deep Learning System - Journal of Big Data. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00333-6.

[3] Katherine Li, 2022. Predicting Stock Prices Using Machine Learning. https://neptune.ai/blog/predicting-stock-prices-using-machine-learning.

[4] Roshan Adusumilli, 2020. Predicting Stock Prices Using a Keras LSTM Model. https://towardsdatascience.com/predicting-stock-prices-using-a-keras-lstm-model-4225457f0233.

[5] Bee Guan Teo, 2021. Stock Prices Prediction Using Long Short-Term Memory (LSTM) Model in Python. https://medium.com/the-handbook-of-coding-in-finance/stock-prices-prediction-using-long-short-term-memory-lstm-model-in-python-734dd1ed6827.

[6] Chi-Feng Wang, 2019. The Vanishing Gradient Problem. https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484.

[7] Eugenio Culurciello, 2019. The Fall of RNN / LSTM. https://towardsdatascience.com/the-fall-of-rnn-lstm-2d1594c74ce0.

[8] Matthew Stewart, 2020. Predicting Stock Prices with Echo State Networks. https://towardsdatascience.com/predicting-stock-prices-with-echo-state-networks-f910809d23d4.

[9] Naima Chouikhi, 2018. Genesis of Basic and Multi-Layer Echo State Network Recurrent Autoencoder for Efficient Data Representations. https://arxiv.org/ftp/arxiv/papers/1804/1804.08996.pdf.

[10] Herbert Jaeger, Harald Haas, 2004. EPORTS Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. http://www.columbia.edu/cu/biology/courses/w4070/Reading_List_Yuste/haas_04.pdf.

[11] Ahershy, 2019. AMZN, DPZ, BTC, NTFX Adjusted May 2013-MAY2019. https://www.kaggle.com/datasets/hershyandrew/amzn-dpz-btc-ntfx-adjusted-may-2013may2019.

[12] Michael Phi, 2020. Illustrated Guide to LSTM's and GRU's: A Step-by-Step Explanation. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.

[13] Tiago Miguel, 2021. How the LSTM Improves the RNN. https://towardsdatascience.com/how-the-lstm-improves-the-rnn-1ef156b75121.

[14] Derrick Mwiti, 2018. Using a Keras Long Short-Term Memory (LSTM) Model to Predict Stock Prices. https://www.kdnuggets.com/2018/11/keras-long-short-term-memory-lstm-model-predict-stock-prices.html.

[15] TensorFlow, 2022. Tf.keras.layers.LSTM: Tensorflow Core v2.9.1. https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM.

[16] Ekourkchi, 2021. PyEsn. https://pypi.org/project/pyEsn/.