Autoencoder based Generative Adversarial Networks for Stock Market Prediction

by

Md Nahid Hasan Shuvo
Department of Computer Science
College of Information Science and Technology
University of Nebraska at Omaha
Omaha, NE 68182-0500
nshuvo@unomaha.edu

For

CSCI 8110 Advance Topics in Artificial Intelligence
Fall 2022

Date: Monday, December 05, 2022

Introduction

Rising stock market valuations and robust returns have attracted more investors in recent years. Some institutions and pioneering thinkers persist in putting money into the stock market despite the obvious dangers. There has been research into stock market forecasting for quite some time, but the subject is so intricate that only a small subset of possible variables and data sources have been explored. Clearly, this is an extremely challenging endeavor. Investors can gain an advantage in making decisions by keeping up with the constantly fluctuating value of stocks. In the present day, there are around 60 main stock exchanges operating across the globe. The total value of all stocks and bonds traded worldwide rose steadily from $2.50 trillion in 1980 to $68.65 trillion by the end of 2018. By the end of 2020, the global market capitalization had topped $100 trillion, despite the fact that the stock market had remained functional and actively supported the economy during the pandemic's crisis era. The stability and regularity of stock market returns are what keep investors confident and the market functioning. This requires shareholders to keep up with the dynamic nature of stock prices. The construction of a reliable stock-price forecasting framework is one method that can be used to achieve this goal. In 2012, algorithms were expected to handle 85% of U.S. stock market trades. Classical models and machine learning-based approaches can predict stock prices. AR, ARIMA, and GARCH can compute linear correlations, but stock forecasting is non-linear. Neural network models are effective for this [1]. Financial markets are nonlinear, dynamic, and noisy, making forecasting challenging.

When it comes to analyzing sequence data, RNNs have been shown to be among the most efficient models [2], thanks to their ability to spot complex non-linear relationships. Pure RNN channels, however, can struggle with long-term memory. As a result, RNN variants like LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) have emerged as the most efficient solutions. To all intents and purposes, LSTM and GRU are the same thing, with the exception that GRU swaps out a few gates in the LSTM network for a more efficient and speedier memory system. Although LSTM seems more effective than GRU, there is still debate regarding when to use each algorithm. Bidirectional LSTM and GRU are two further varieties of RNN. Having two identical-based models facing in opposing directions, these networks have twice as much information as raw LSTM and GRU systems. In this way, Bidirectional LSTM has proven to be more effective than the LSTM network [3].

Generative Adversarial Network (GAN) is one of the most effective deep learning approaches, because it trained by adversarial manner [14]. After training generator, a part of the GAN can generate very realistic data, also, the other part, discriminator can discriminate between any two data sample very effectively. There are few studies on time series forecasting using GAN. Thus, in this paper I am going to explore the effectiveness of GAN based approach for time series prediction. To achieve the forecasting efficiently and quickly, we have chosen the LSTM and GRU based network for encoder and decoder to forecast the time series on this project. As we know, there are several cases, like weather forecasting, Cloud workload prediction, even for the cyber security the DDoS attack prediction, this type of forecasting technology might be very useful.

Related Works

Predicting the stock market accurately has long been a goal of education in the field of statical analysis. Many autoregressive problems are now within the capabilities of modern deep learning systems. Now, these algorithms are so expressive that they can detect patterns in extremely difficult regression tasks, such as stock market research. Thus, there has been a proliferation of studies on the application of deep learning to the task of forecasting time series. We will provide a quick overview of some of these works here.

Historically, MLPs were used to predict the stock market, but their high parameter count made them susceptible to overfitting problems. In constrained regression contexts, straightforward approaches like linear regression and KNN have demonstrated some forecasting efficacy. When combined with cutting-edge feature engineering methods, decision trees [4] have been shown to improve accuracy significantly. In contrast, SVMs, despite their seeming lack of complexity, have shown to be both very adaptable and significantly helpful when it comes to making reliable predictions about time series data. However, SVMs [5] cannot correlate the very unstructured data introduced by stock markets because of their restricted expressivity. In addition, a memory component is needed for such data in order to take into account previously significant signals, allowing for an educated prediction of the current situation. The majority of RNNs don't incorporate any sort of memory, despite the fact that many of them have shown great success with sequential input. The problem of long-term memory dependence was partially alleviated by the advent of LSTM [6] networks. Using an LSTM model, Shao et al. [7] established a system that can predict parking availability over a prolonged period of time. In order to foretell cloud resource use, Yuming et al. [8] presented the GRU-ES, which combines the gated neural recurrent unit model with the exponential smoothing method. In both single- and multi-step prediction tasks, the suggested technique excels over SOTA models. Recently, a number of high-profile academic publications have begun using a framework that allows LSTM networks to perform stock market forecasting tasks. In order to achieve the highest possible accuracy while predicting DSE data, Sadman et al. [9] proposed LSTM based RNN with thorough hyperparameter tuning. In their study, Kai et al. [10] shown that the LSTM model outperformed other regression models in terms of accuracy. Accurate time series data prediction can also be achieved with sliding window-based CNN [11] networks. This methodology is commonly used to construct powerful sentiment analysis and classification software. Sreelekshmy et al. [12] used LSTM and CNN-sliding window techniques to forecast the stock price of businesses listed on the NSE. To improve the precision of their forecasts in a setting where they care most about reducing the RMSE value, Arif et al. [13] used Bi-directional LSTM.

Methodology

A. Feature Extraction by 1DCNN:

Convolution layers are very good at finding features, so they were used to learn local patterns from the information that was given, even though Dense Networks learn different patterns from the feature space that was given. With ConvNet, the length of time series data can be cut down to a much more manageable length. Figure 1 shows, in a simplified way, how a 1-dimensional CNN works. At the beginning, unwanted sequences are filtered out with the help of a kernel with a fixed window size. A 1D convolution is used to split the sequences of inputs into their separate parts. The sequences that are retrieved are used to weight the layer in the dot production process, and then the output sequences are gotten. This output sequence has high-level features that can be used to feed into the dense layer of any neural network system. Figure 1 shows how the LSTM cell is put together.

B. Long Short-Term Memory (LSTM):

LSTM is a varient of RNN called "long-short-term memory" can learn to solve problems with long-term dependencies. One study says that LSTM is able to deal with long-term dependent time series data in the right way. LSTM neurons have an input gate, a forget gate, and an output gate. LSTM directs the flow of data through these three functional gates, which also store data from the past. In Figure 2, we can see how the LSTM cell is put

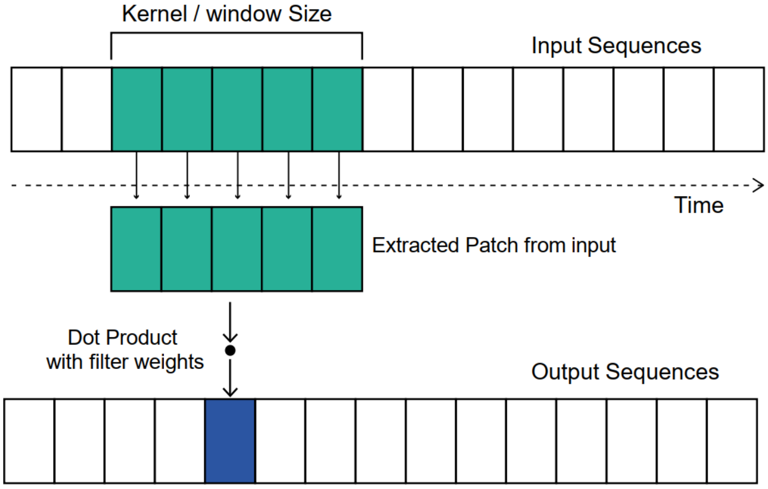

Figure 1: Working procedure of 1D CNN

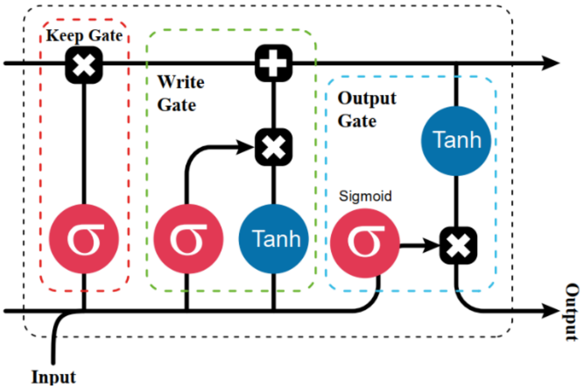

Figure 2: LSTM cell

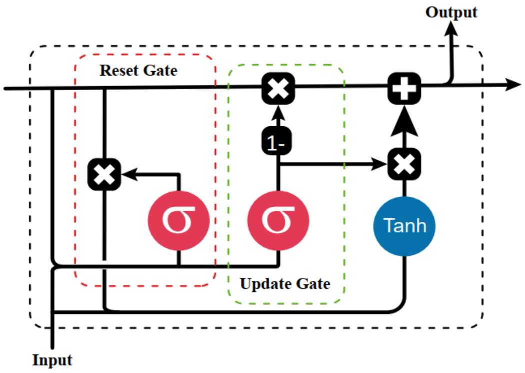

Figure 3: GRU cell

together. The most important part of an LSTM is the memory cell. It has important information that has been added to it over time, and the three gates change the new information at each step. So, LSTM is better than RNN at figuring out long-term dependencies. The keep gate tells how much information from the previous cell should be kept, and the write gate tells how much information should be sent to the memory state. This solves the vanishing gradient problem.

The key component of an LSTM is the memory cell. The three gates change the incoming data at each stage, and it stores the vital information it has acquired over time. Accordingly, LSTM is far better than RNN in learning long-term dependencies. Keep gates determine how much data is kept from the previous cell, while write gates determine whether data is written to the memory state, thus avoiding the vanishing gradient problem.

C. Gated Recurrent Unit (GRU):

Figure 2 depicts GRU, a variation of LSTM. GRU was invented because LSTM's complexity required more training time. GRU improves input and forget through update gate. Her update gate functions as an input gate in LSTM to keep necessary information. Along with the update gate, the GRU unit features a gate for discarding old data. Cell structure becomes two gates instead of three. GRU has improved computational efficiency and accuracy because to fewer components per cell than LSTM.

D. RNN based Autoencoder:

An autoencoder is a neural network that learns efficient coding of unlabeled data. It mainly comprises two parts encoder and a decoder. The encoder compresses input data from high-dimension to low-dimensional latent space, on the other hand, the decoder captured important information from the latent information by decoding it. RNN-based autoencoder mainly encodes the sequence and then the decoder predicts the future sequence.

E. Generative Adversarial Network (GAN):

GAN has two parts, Generator and Discriminator. This is basically a minmax problem. The generator tries to generate samples that are closest to the real sample, on the other side, the discriminator tries to differentiate between the original and generated sample. In our approach we used an autoencoder-based generator model using LSTM and GRU network, For the generator, we used LSTM as it has the capability to store longer sequences, thus encoder encoded the longer sequence and in the Decoder portion, we use GRU which efficiently decodes the information from latent space.

Moreover, the discriminator of our GAN architecture uses a convolution neural network that distinguishes between synthetic data and the original stock price. The sigmoid function of the discriminator will give the output between 0-1 which will determine the efficiency of the model by pointing out the real and fake data.

F. System Architecture:

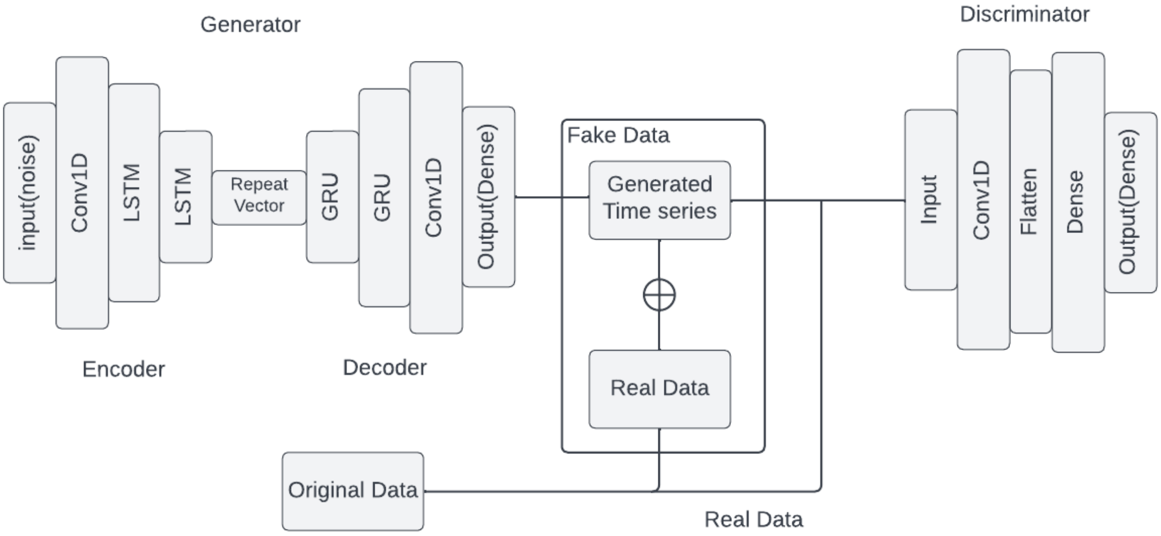

Figure 4: Encoder-decoder based GAN architecture

Here in this work, I proposed an Encoder-Decoder-based Generative Adversarial Network. In the generator autoencoder was used, so in the encoder network, Conv1D of 512 units was used, and LSTM network of 400, 150,60, and 8 units were used, On the other hand for Decoder 8,60,150, and 400 units of GRU layer used along with the 512 unit of Conv1D. And for the discriminator, I used a very simple discriminator which has 4 convolution layers and 4 dense layers. In figure 4 I have shown the top-level architecture of our approach.

Experiment

A. Dataset:

We use stock price and index data from Yahoo Finance. The target date was Apple. Inc. stock closing price. There are a total of 2497 data samples and 36 columns in the dataset. We splitted the data into 7:3 manners.

B. Result and Analysis:

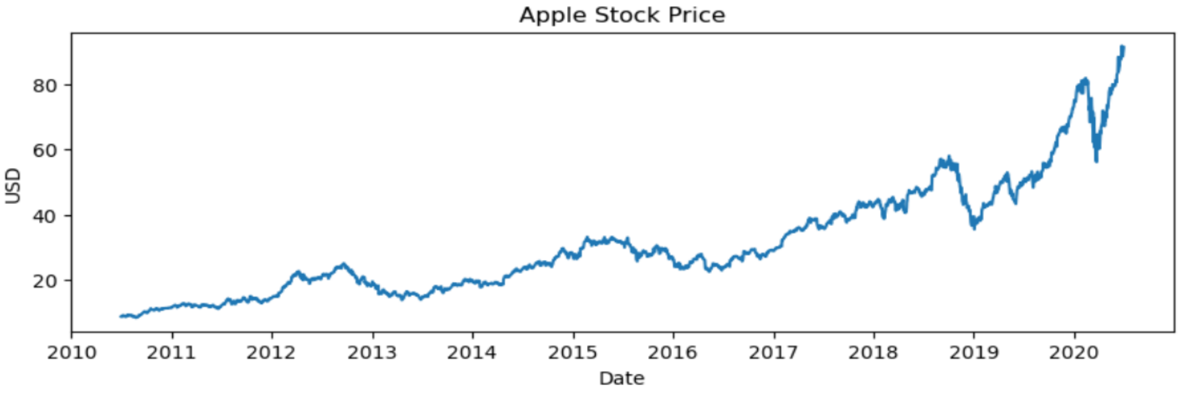

Figure 5: Real stock price data

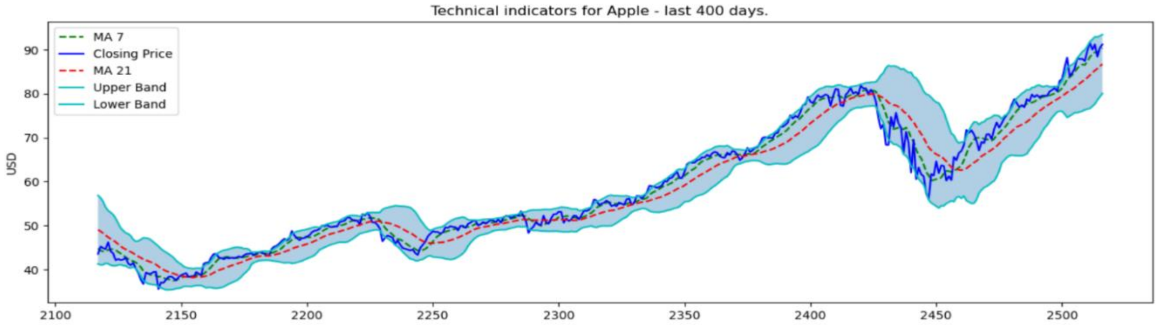

Figure 6: Technical indications for Apple

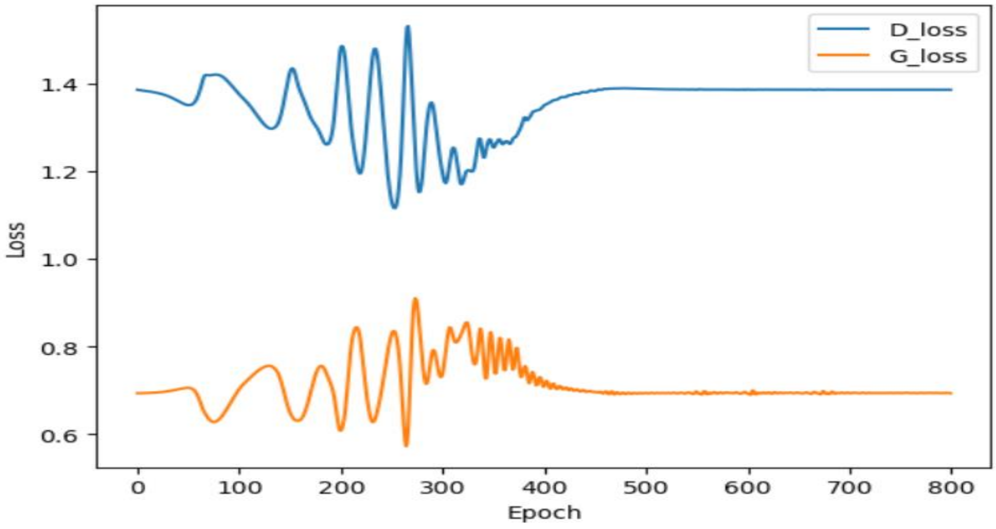

Figure 7: Generator vs Discriminator Loss

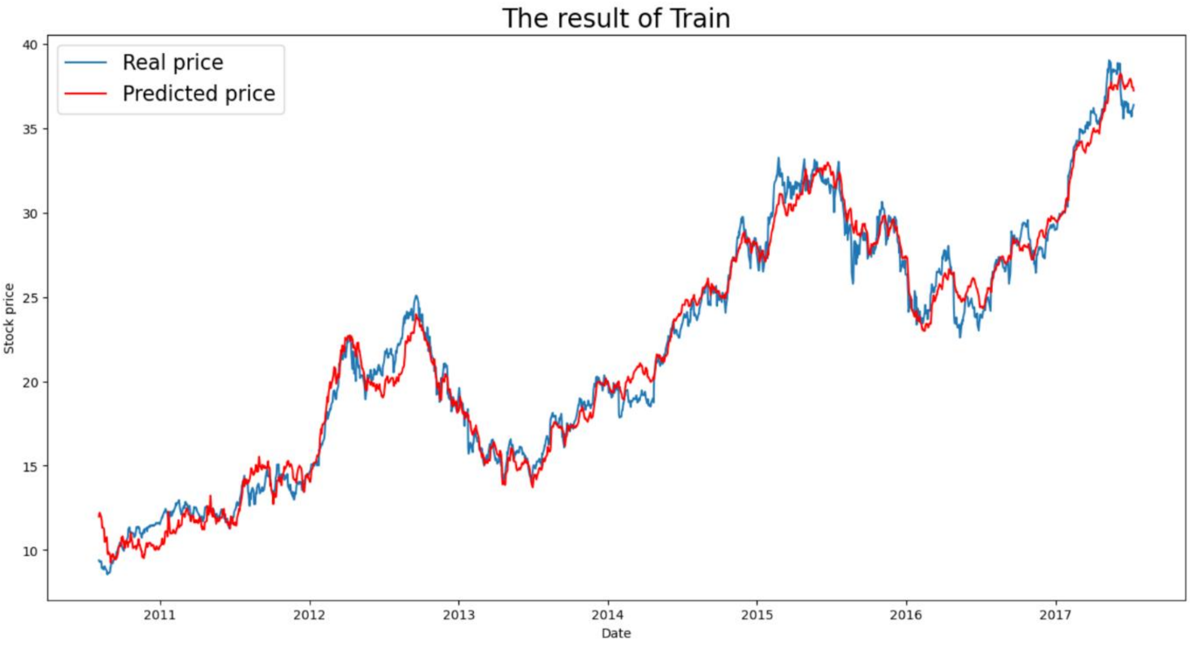

Figure 8: Generated vs original stock price data without GAN

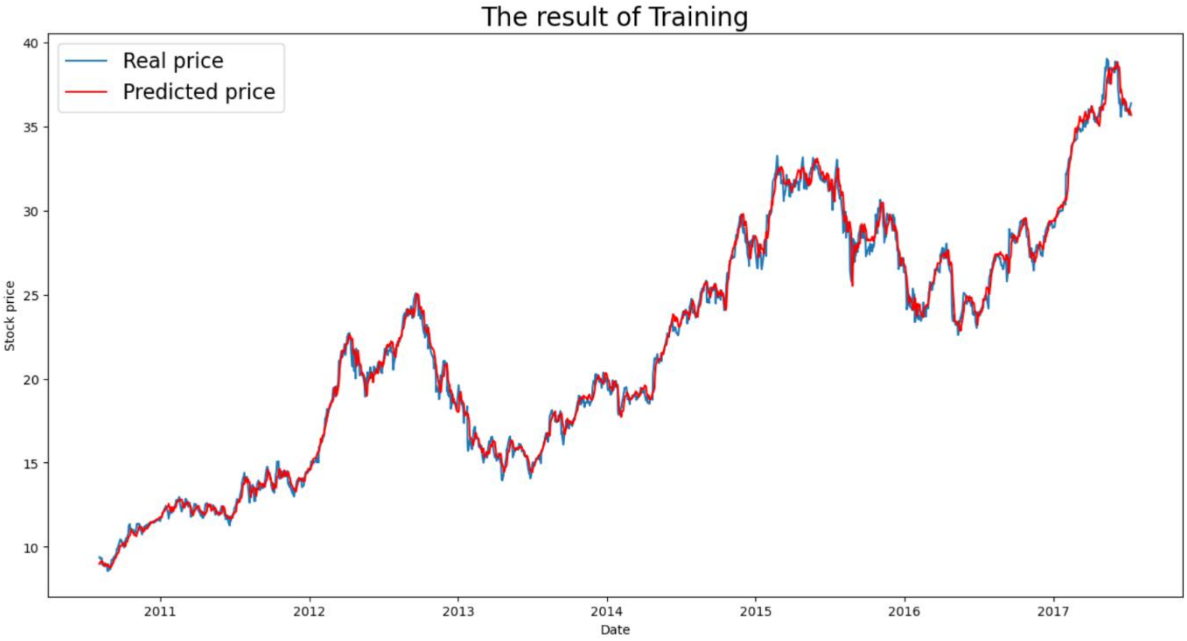

Figure 9: Generated vs original stock price data using GAN

In figure 6, we can see the closing stock price of Apple.Inc., and in figure 7 we can see some preprocessed results applied to the original dataset, I have analyzed the dataset and got new columns for the Moving average, upper band, and lower band of the data. These new features help to predict the stock market more accurately. In figure 7, we can see the loss of the generator and discriminator. We can see that during 500 epochs, the training was not stable, but after 500 epochs the training of both models become stable. Using the trained model, the generation of the stock price was done, and we found almost accurate data. We also compare our approach with LSTM and GRU, and we found GAN based model has lower errors than others. In figure 8, we can visualize the real vs predicted stock price data using a pure LSTM network, however, in figure 9 we see that output is more accurate, and the main reason behind that was the use of autoencoder and GAN. Also, the RMSE error using LSTM was **0.93** whereas the RMSE error using Autoencoder-GAN was **0.53**, which proves the superiority of the GAN-based approach. Moreover, if we look at the prediction result we can see that, our approach can detect sharp changes along with the slightly changes in price.

Conclusion

In this work, we see that LSTM used in the encoder can extract the long-term dependencies and compress that information into lower dimensions, and GRU decoded that information efficiently. Which makes the generation of the synthetic data more accurate. On the other hand, training the generator in an adversarial manner, effectively increase the efficiency of the prediction.

In the future, I will analyze, more on the effectiveness of the different GAN architectures for time series analysis, like data center workload forecasting, or weather forecasting.

Runnable Code Link: https://colab.research.google.com/drive/1jm0_Ef7rouC9i1F-Jz120fZOSRvJWkWW?usp=sharing

Reference

[1] Ferreira, Tiago AE, Germano C. Vasconcelos, and Paulo JL Adeodato. "A new intelligent system methodology for time series forecasting with artificial neural networks." *Neural Processing Letters* 28.2 (2008): 113-129.

[2] Pham, Dinh Tuan. "Stochastic methods for sequential data assimilation in strongly nonlinear systems." *Monthly weather review* 129.5 (2001): 1194-1207.

[3] Siami-Namini, Sima, Neda Tavakoli, and Akbar Siami Namin. "The performance of LSTM and BiLSTM in forecasting time series." *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019.

[4] Nair, Binoy B., V. P. Mohandas, and N. R. Sakthivel. "A decision tree-rough set hybrid system for stock market trend prediction." *International Journal of Computer Applications* 6.9 (2010): 1-6.

[5] Lin, Yuling, Haixiang Guo, and Jinglu Hu. "An SVM-based approach for stock market trend prediction." *The 2013 international joint conference on neural networks (IJCNN)*. IEEE, 2013.

[6] Yu, Yong, et al. "A review of recurrent neural networks: LSTM cells and network architectures." *Neural computation* 31.7 (2019): 1235-1270.

[7] Shao, Wei, et al. "Parking availability prediction with long short term memory model." *International Conference on Green, Pervasive, and Cloud Computing*. Springer, Cham, 2018.

[8] Cheng, Yuming, et al. "Gru-es: Resource usage prediction of cloud workloads using a novel hybrid method." *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2019.

[9] Islam, Sadman Bin, Mohammad Mahabubul Hasan, and Mohammad Monirujjaman Khan. "Prediction of Stock Market Using Recurrent Neural Network." *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2021.

[10] Chen, Kai, Yi Zhou, and Fangyan Dai. "A LSTM-based method for stock returns prediction: A case study of China stock market." *2015 IEEE international conference on big data (big data)*. IEEE, 2015.

[11] Sadouk, Lamyaa. "CNN approaches for time series classification." *Time Series Analysis-Data, Methods, and Applications* (2018): 1-23.

[12] Selvin, Sreelekshmy, et al. "Stock price prediction using LSTM, RNN and CNN-sliding window model." *2017 international conference on advances in computing, communications and informatics (icacci)*. IEEE, 2017.

[13] Sunny, Md Arif Istiake, Mirza Mohd Shahriar Maswood, and Abdullah G. Alharbi. "Deep learning-based stock price prediction using LSTM and bi-directional LSTM model." *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*. IEEE, 2020.

[14] Goodfellow, Ian, et al. "Generative adversarial networks." *Communications of the ACM* 63.11 (2020): 139-144.