Modeling Stock Market Returns using Deep Learning

Devansh Gupta*
Computer Science and
Artificial Intelligence
IIIT Delhi
devansh19160@iiitd.ac.in

Divyansh Rastogi*
Computer Science and
Artificial Intelligence
IIIT Delhi
divyansh19464@iiitd.ac.in

Rupanshu Yadav*
Computer Science and
Artificial Intelligence
IIIT Delhi
rupanshu19475@iiitd.ac.in

Abstract

Stock market forecasting has long been a popular and lucrative field of study. In the subject of financial prediction and forecasting, machine learning applications have been shown to produce improved accuracy and returns. Although deep learnings techniques are extensively used in stock market forecasting, some models such as Generative Adversarial Networks and Neural ODEs have not been properly assessed. In this paper, we propose a Generative Adversarial Network with LSTM and CNN as generator and discriminator respectively. Furthermore we also explore a class of models called Neural ODEs whose properties are used to create an "infinite" depth classifier and a variational autoencoder which utilizes the properties of neural ODE to generate realistic closing price data according to the dynamics learnt.

1 Introduction

The stock market provides investors with the opportunity to share the profits of companies. More than five trillion dollars worth of stocks are traded every single day [1]. Stock market forecasting has long been a popular and lucrative field of study, with more than 500 publications in the past five years (Hu et al., 2021a). Investors can maximize their profits by investing rationally based on technical analysis (de Souza et al., 2018). The time series prediction problem is complex due to the widely accepted semi-strong efficient market hypothesis (Malkiel and Fama, 1970) and their volatile & non-stationary structure (Adam et al., 2016) . Hence features such as news sentiment, related markets & 8-K Reports are widely used. Prediction of assets or commodities such as currencies, gold, & crypto-currencies fall into the same categories of problems.

[1]https://www.nasdaq.com/articles/forex-market-overview-2019-06-07

Stock market analysis methods fall into two categories, technical analysis, and fundamental analysis. Technical analysis involves the prediction of stock prices using historical data, primarily price and volume. It is worth noting that the more efficient the market is, the harder it would be to earn profits using technical analysis. On the other hand, the fundamental analysis makes predictions based on the intrinsic value of the investment by studying macroeconomic features, industry features, and company's features, including financial reports, competitors, & viability.

Statistical models such as Auto Regressive Moving Average (ARMA), Auto-Regressive Integrated Moving Average (ARIMA), Smooth Transition Autoregressive (STAR) are widely used for time series prediction purely based on the historical price of a stock. Machine learning has been intensively researched for its use in financial market forecasting. Decision trees, XGBoost, and Naïve Bayes all have shown to be beneficial in stock market price prediction (Ballings et al., 2015). Deep learning methods have also proven to be helpful in time series analysis (Fawaz et al., 2019).

2 Literature Review

Financial markets have shown to be volatile and non-stationary (Dionisio et al., 2007; Adam et al., 2016) with having a close resemblance to an ensemble of particles in statistical mechanics, precisely Brownian motion (Osborne, 1959). Despite the existence of multiple opposing views on the efficiency of stock markets, research reveals that financial markets are, to some extent, predictable (Ferreira and Santa-Clara, 2011; Kim et al., 2011; Bollerslev et al., 2014). Studies have evaluated the predictive relationships of numerous financial and economic variables (Enke and Thawornwong, 2005; Kim and Han, 2000; Vellido et al., 1999), showing that the closer historical data time is to the present, the stronger the data's impact on the predictive model (Liao and Wang, 2010).

Various statistical & econometric machine learning methods (Agrawal et al., 2013) have been employed to predict stock returns/directions like ARIMA, SVM (Schumaker and Chen, 2009; Lee, 2009; Cao and Tay, 2001), Boosted Regression Trees (Pierdzioch et al., 2018) and Logistic Regression (Chong et al., 2017). Ou and Wang 2009 discussed & applied ten different data mining techniques to predict price movement in the Hong Kong stock market Hang Seng index. For a survey on various employed machine learning techniques for stock market predictions, we refer the reader to Strader et al. 2020.

With the rise of deep learning applications in stock markets (Hu et al., 2021b), its shown to help model complex intrinsic relationships and extract abstract features from data without relying on econometric assumptions or human expertise (Chong et al., 2017). Although to achieve accurate results with neural networks, it is important to have a deliberate selection of input variables (Lam, 2004). For market prediction, various feature selection & extraction methods have been used alongside machine learning techniques such as Genetic Algorithms (Kim and Han, 2000), Autoencoders (Chong et al., 2017), Restricted Boltzmann Machine (RBM) (Chong et al., 2017) and Principal Component Analysis (PCA) (Zhong and Enke, 2019; Chong et al., 2017). Zhong and Enke 2019 provides a comprehensive study on PCA and its non-linear fuzzy variants used along with Artificial Neural Networks (ANN) and Deep Neural Networks (DNN) for forecasting the daily return direction of the SPDR S&P 500 ETF index. There also has been limited work in exploration of generative adversarial models for stock market prediction. Zhang et al. 2019 used GANs to predict the daily closing price of the S&P500 index.

3 Dataset

We used the CNNPred dataset used in (Hoseinzade and Haratizadeh, 2019) to get results on the existing models. This dataset consisted of market prices of NASDAQ, NYSE, S&P500, DJI, and RUSSELL from 2010 to 2017. This dataset had the market's closing price at a particular day along with a diverse set of features consisting of technical indicators like Exponentially Weighted Average of data from a varying number of previous days, returns, and their weighted averages. Along with the technical indicators, the dataset consisted of additional data on commodities like oil, gold, and silver prices. The features also had the currency exchange rates, returns from other markets, and stock futures which have shown to be important in determining the stock's current price on the market.

4 Baselines

We have chosen our baselines to explore a wide range of machine learning methods to stock price prediction and trend classification. Our baselines start with statistical models, transitioning into classical machine learning, and then to deep learning.

4.1 ARIMA

ARIMA is widely used for prediction in time series data modeling. ARMA requires the time series to be stationary; which means a constant mean, constant variance and non-seasonal. Since, stock market prices don't have constant mean, hence a transformed feature, defined as

$$y_i = price_i - price_{i-d}$$

where d is the lag. Auto-Regressive(AR) and Moving Average(MA) models are parameterized by auto-correlation coefficient(p) and Partial auto-correlation coefficient(q). A search is performed given the maximum values of p and q to find the best model. An RMSE of 423.24 was observed. However, if the first testing point is included in the training data to predict the next point, an RMSE of 49.72 was observed.

4.2 PCA+ANN

This model has been given in Zhong and Enke 2019 for forecasting daily return direction of the SPDR S&P 500 ETF index. Upon examination, we discovered that all of the 60 financial variables evaluated in the study's dataset are already present in & analogous to the 82 features of our chosen dataset, CNNPred. Thus, the study's model is extrapolated to the CNNPred dataset's S&P market.

Zhong and Enke explored multiple data transformation techniques including PCA and its variants, fuzzy robust principal component analysis (FR-PCA) and kernel-based principal component analysis (KPCA), among others. Their results showed that traditional PCA outperformed all non-linear

Market Name	MAE	RMSE	Max Closing Price
NASDAQ	2177.418	2251.247	341682.12
NYSE	2461.052	2538.778	388508.06
S&P500	799.763	815.964	24420.32
RUSSELL	1350.78	1355.90	12414.37
DJI	7288.813	7455.721	2980330.8

Table 1: Initial results on the Regression metrics on the test set using ODE VAE model trained for 150 Epochs on the sequential closing price data of the CNNPred Stock market Dataset using a GRU encoder and a NeuralODE Decoder

techniques on real-world data. Thus, PCA is chosen as our data transformation technique and PCA-represented dataset with 82 principal components is used.

4.2.1 Data Preprocessing

A classical statistical principle is used for detection of outliers based on inter-quartile ranges (Navidi, 2011). These outliers are accordingly adjusted similar to a method used by Cao and Tay 2001. The cleaned data is split in 70/15/15 ratio for train, validation and test dataset respectively. The data is standardized with the mean and variance of the training dataset.

4.2.2 Model & Training

The PCA-represented dataset is classified using a ANN network comprising of 4 layers with RELU activation & Sigmoid activation for the last layer. Dropout has been introduced in the network to avoid overfitting. Binary cross entropy loss is used as the loss criterion. The initial learning rate was set to 0.0001 with ADAM optimizer for training over maximum of 100 epochs. Early stopping is implemented with the use of validation set.

4.2.3 Results

We obtained an accuracy of 0.559 and a F1-score of 0.656 on the test set.

Market Name	Average	Maximum
NASDAQ	0.49	0.536
NYSE	0.432	0.566
S&P500	0.491	0.67
RUSSELL	0.489	0.521
DJI	0.486	0.5

Table 2: F-Scores on the test set using CNNPred3D trained for 100 Epochs on the CNNPred Stock market Dataset

4.3 CNN-Pred2D

This model (Hoseinzade and Haratizadeh, 2019) classifies the change in the closing price of the market using only the data for the market under analysis. It takes the input of all the features from last 60 days and leverages 2D convolution filters for making feature maps and finally classifying the change in price. Since CNNs are good at capturing short range data and hierarchically extract features from day wise data, they serve as a good method for predicting stock prices. The initial learning rate for training this network was set to 0.001, and the ADAM optimizer was used for training over 100 epochs. Since the network was quite small with only one fully connected layer and three convolution layers, a weight decay of 0.0001 was sufficient to regularize the network. Each feature was normalized according to the training set and was used during the validation and test time.

Market Name	Average	Maximum
NASDAQ	0.438	0.552
NYSE	0.491	0.492
S&P500	0.434	0.573
RUSSELL	0.497	0.691
DJI	0.471	0.696

Table 3: F-Scores on the test set using CNNPred2D trained for 100 Epochs on the CNNPred Stock market Dataset

4.4 CNN-Pred3D

This model (Hoseinzade and Haratizadeh, 2019) classifies the change in the closing price of the market using the data of various markets. It takes the input as a 3D block of all the features from the 5 markets in the dataset over the last 60 days and leverages 3D convolution filters for making feature maps and finally classifying the change in price by using data across many markets. The usage of 3D convolution filters is the same as 2D convolution filters only that these filters hierarchically extract features and are temporally sensitive over an additional dimension. Similar to the CNN-Pred2D, the initial learning rate for training this network was set to 0.001, and the ADAM optimizer was used for training over 100 epochs with a weight decay of

0.0001. Each feature was normalized according to the training set and was used during the validation and test time.

4.5 CNN-LSTM Model

This model (Lu et al., 2020) considers learning sequence aware features as the stock market is an event which moves in the temporal dimension, thus it is difficult to ignore the sequential information present in the latent embeddings for the downstream tasks. This work explored the price prediction perspective for more informed stock market trading and hence was a regression task. The convolutional features were used to calculate the temporal features over time stamps and then an LSTM was used to capture the sequential features. The structure of LSTM is designed in such a manner that it works on selectively learning which information to hide and which to infer on over a certain time step and pass both the states for the next time step (Hochreiter and Schmidhuber, 1997). Thus, this overparameterization leads to a delayed stability of LSTMs in terms of metrics but provably results to a more optimal result in lesser number of iterations (Arora et al., 2018). This model gave sufficiently good results on predicting the closing price of a market on training with 100 epochs with a learning rate of 0.001 on ADAM optimizer, with a weight decay of 0.0001.

5 Final Models

5.1 GANs

GANs, or Generative Adversarial Networks were introduced by Goodfellow et al. (2014). A discriminative model D that calculates the probability that a sample originated from the training data rather than G, and a generative model G that generates the data distribution. This is similar to a two player minimax game, defined as :

$$\begin{aligned}\min_G \max_D (&E_{x\sim p_{data}(x)}[\log D(x)] \\ &+E_{z\sim p_z(z)}[\log(1-D(G(z)))])\end{aligned} \tag{1}$$

Their effect has been seen mostly in the realm of computer vision, where realistic picture and video manipulation, particularly generation, has made tremendous progress. While these developments in computer vision have gotten a lot of press, GAN applications have expanded to include time series and sequence creation. GANs have been successfully used in time series anomaly detection, generation, imputation (Brophy et al., 2021).

5.2 Neural ODEs

Neural Ordinary Differential Equations, introduced by Chen et al. (2018), are a class of models which are trained by solving an ordinary differential equation rather than backpropagating gradients. We mainly use it as an infinite depth residual network to make an end to end classifier for predicting the stock trends. The ResNet expression can be given in Equation 2. This can be seen as an Euler's discretization of a continuous transform. Hence we can take the continuous case as given in Equation 3 and take each time step to be a layer, constituting a solution of an "infinite" depth neural network on ODE solving.

As an additional study into generative modelling using Neural ODEs, we also used an ODE Variational Autoencoder(ODEVAE)(Chen et al., 2018). It learns to generate realistic time series based on samples of time series given. Neural ODEs have known to be excellent models for capturing the dynamics of the system and maintaining continuity in time series models. Hence, a deep sequential encoder to generate the appropriate hidden state giving out the mean and the logvar of the initial state distribution that the Neural ODE is to take in and then ODE solve the time steps t as entered. We use the loss as given in (Kingma and Welling, 2014) consisting of the reconstruction and the KL Divergence loss in order to learn the distribution of sequences.

$$h_{t+1} = h_t + f(h_t, \theta_t) \tag{2}$$

$$\Delta_t h(t) = f(h(t), t, \theta) \tag{3}$$

6 Results & Analysis

6.1 Neural ODE Classifier

This model uses the same inputs as the (Hoseinzade and Haratizadeh, 2019) with the same feature normalization performed. We used an already made implementation of the forward and backward pass for ODE solving and made the feature map extraction architecture of our own with the ODE solving as a module taking the feature map as the input. The feature map extractor is a simple 2D CNN architecture which takes in the 3D input, as represented by an instance. We got faster training where the training converged to a stable f-score

Market Name	MAE	RMSE	R^2	Max Closing Price
NASDAQ	645.21	854.89	0.99	341682.12
NYSE	433.31	577.70	0.99	388508.06
S&P500	143.58	192.79	0.99	24420.32
RUSSELL	162.90	205.95	0.99	12414.37
DJI	1899.07	2551.76	0.99	2980330.8

Table 4: Regression metrics on the test set using CNN-LSTM model trained for 100 Epochs on the CNNPred Stock market Dataset

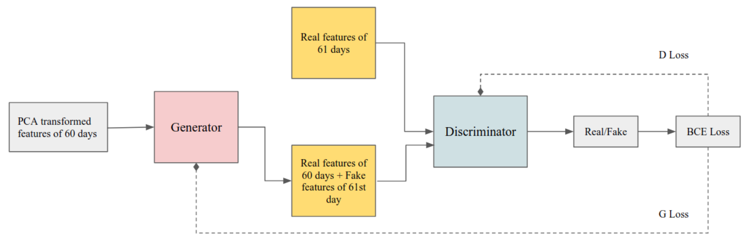

Figure 1: Structure of GANs Model

in 15 epochs as compared to the 100 epochs of CNNPred2D and CNNPred3D (Hoseinzade and Haratizadeh, 2019). We also got improved classification results on training this classifier on our dataset as shown in Table 5, for using only data of a single market as input, and Table 6 for using data of all markets as input. The optimizer used was ADAM with a learning rate of 0.001 for 15 epochs.

Market Name	Average	Maximum
NASDAQ	**0.48**	**0.579**
NYSE	**0.531**	**0.567**
S&P500	**0.542**	**0.622**
RUSSELL	**0.517**	0.584
DJI	**0.519**	0.589

Table 5: F-Scores on the test set using infinite depth classifier using NeuralODE trained for 15 Epochs on the CNNPred Stock market Dataset on single market data as input. Bold entries denote an improvement over the results obtained on the CNNPred2D baseline

Market Name	Average	Maximum
NASDAQ	0.438	**0.55**
NYSE	**0.567**	**0.624**
S&P500	**0.541**	**0.556**
RUSSELL	**0.491**	0.495
DJI	**0.521**	**0.527**

Table 6: F-Scores on the test set using infinite depth classifier using Neural ODE trained for 15 Epochs on the CNNPred3D Stock market Dataset on all the market data as input. Bold entries denote an improvement over the results obtained on the CNNPred3D baseline

6.2 Generative Models

6.2.1 Neural ODE VAE

We used a GRU encoder to encode the sequence to a hidden state and then used a neural ODE as a decoder. The encoder-decoder networks were trained on a data of 6o days and during inference time, data of 59 consecutive days were given in order for the model to predict the closing price for the 60th day while ODE solving the ODE decoder using Euler's method. We only use two dimensional data consisting of the generated time-steps and the closing price on which we have done the training and made the inferences. We used a latent dimension of 4 for the GRU and trained for 100 epochs using the ADAM optimizer with a learning rate of 0.001. Even though we get inferior results as compared to CNN-LSTM as given in Table 1. We also concluded from this set of experiments that the data that we have was not sufficient in training the ODE-VAE model as generative model like VAEs require a lot of samples for training. Furthermore, we also made the ODE on the closing price as that was the only available data in the dataset. A knowledge of the exact opening, closing, and median prices would have been a more ideal setting in training this model.

6.2.2 Generative Adversarial Networks

The GAN model builds upon the idea to predict PCA transformed features as utilized in Zhong and Enke (2019). The model i.e. the generator is trained in an adversarial setting. The model is tested on the S&P 500 index. Initially data is pre-

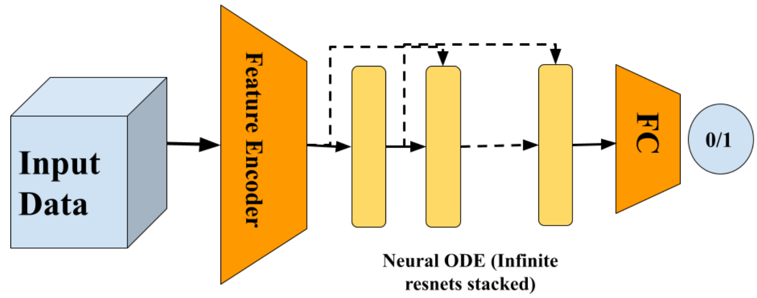

Figure 2: Structure of Neural ODE Classifier

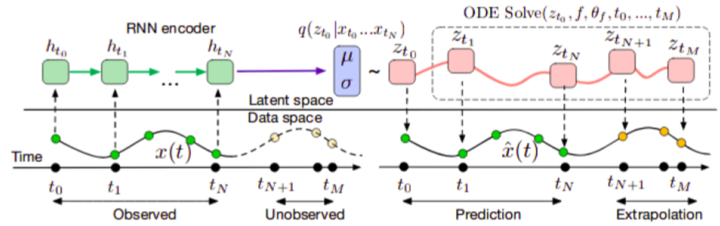

Figure 3: Structure of Neural ODE VAE (Chen et al., 2018)

processed with outlier adjustments as similar to (Cao and Tay, 2001) and then the data is split in 80/20 ratio for train and test set respectively. Furthermore, the data is standardized with the mean and variance of the training dataset.

The generator utilizes a shallow LSTM single layer network followed by a fully connected layers comprising of TanH and ReLU activations along with batch normalization. The generator's aim is to take features of past 60 days and generate the new features of the 61st day. The discriminator is a deep CNN network that takes in features of 61 days and predicts whether the this trend is real or fake. The loss function used to train the discriminator is BCE Loss. Finally, we use the output of the generator to predict the direction of the closing price of the index.

The accuracy on the train set and the test set were observed to be 0.545 & 0.504 respectively.

7 Conclusions and Future Work

We investigated the feasibility of GANs and Neural ODEs in capturing the distributions of real stock market data. On comparison with traditional PCA+DNN, GANs performed slightly worse. This could be majorly attributed to the careful tuning required for training a GAN. It should be noted that GANs are significantly harder to tune compared to CNN and LSTM. Neural ODEs as infinite depth classifiers gave superior results as compared to our baselines but the ODEVAE results were slightly inferior, but we hypothesized that since such generative models require a lot of examples to learn, hence a better dataset would be ideal in our scenario.

8 Contributions

All authors denoted by * contributed equally, the ordering of naming is alphabetical.

References

Klaus Adam, Albert Marcet, and Juan Pablo Nicolini. 2016. Stock Market Volatility and Learning. *The Journal of Finance*, 71(1):33–82.

Janhavi Agrawal, Vijay S. Chourasia, and A. K. Mittra. 2013. State-of-the-art in stock prediction techniques. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Energy*, 2:1360–1366.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. 2018. On the optimization of deep networks: Implicit acceleration by overparameterization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 244–253. PMLR.

Michel Ballings, Dirk Van den Poel, Nathalie Hespeels, and Ruben Gryp. 2015. Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 42(20):7046–7056.

Tim Bollerslev, James Marrone, Lai Xu, and Hao Zhou. 2014. Stock return predictability and variance risk premia: Statistical inference and international evidence. *Journal of Financial and Quantitative Analysis*, 49(3):633–661.

Eoin Brophy, Zhengwei Wang, Qi She, and Tomas Ward. 2021. Generative adversarial networks in time series: A survey and taxonomy. *arXiv:2107.11098 [cs]*. ArXiv: 2107.11098.

Lijuan Cao and Francis Tay. 2001. Financial forecasting using support vector machines. *Neural Computing and Applications*, 10:184–192.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. 2018. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*.

Eunsuk Chong, Chulwoo Han, and Frank Park. 2017. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, 83.

Andreia Dionisio, Rui Menezes, and Diana A. Mendes. 2007. On the integrated behaviour of non-stationary volatility in stock markets. *Physica A: Statistical Mechanics and its Applications*, 382(1):58–65. Applications of Physics in Financial Analysis.

David Enke and Suraphan Thawornwong. 2005. The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, 29:927–940.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963. ArXiv: 1809.04356.

Miguel A. Ferreira and Pedro Santa-Clara. 2011. Forecasting stock market returns: The sum of the parts is more than the whole. *Journal of Financial Economics*, 100(3):514–537.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ehsan Hoseinzade and Saman Haratizadeh. 2019. Cnnpred: Cnn-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, 129:273–285.

Zexin Hu, Yiqi Zhao, and Matloob Khushi. 2021a. A Survey of Forex and Stock Price Prediction Using Deep Learning. *Applied System Innovation*, 4(1):9.

Zexin Hu, Yiqi Zhao, and Matloob Khushi. 2021b. A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, 4(1).

Jae H. Kim, Abul Shamsuddin, and Kian-Ping Lim. 2011. Stock return predictability and the adaptive markets hypothesis: Evidence from century-long u.s. data. *Journal of Empirical Finance*, 18(5):868–879.

Kyoung-jae Kim and Ingoo Han. 2000. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*, 19:125–132.

Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Monica Lam. 2004. Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. *Decision Support Systems*, 37(4):567–581. Data mining for financial decision making.

Ming-Chi Lee. 2009. Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, 36:10896–10904.

Zhe Liao and Jun Wang. 2010. Forecasting model of global stock index by stochastic time effective neural network. *Expert Systems with Applications*, 37:834–841.

Wenjie Lu, Jiazheng Li, Yifan Li, Aijun Sun, and Jingyang Wang. 2020. A cnn-lstm-based model to forecast stock prices. *Complexity*, 2020:6622927.

Burton G. Malkiel and Eugene F. Fama. 1970. Efficient Capital Markets: A Review of Theory and Empirical Work*. *The Journal of Finance*, 25(2):383–417.

Navidi. 2011. Statistics for engineers and scientists. *McGraw-Hill, New York*, 3rd edn.

M. F. M. Osborne. 1959. Brownian motion in the stock market. *Operations Research*, 7(2):145–173.

Ou and Wang. 2009. Prediction of stock market index movement by ten data mining techniques. *Modern Applied Science*, 3.

Christian Pierdzioch, Rangan Gupta, Hossein Hassani, and Emmanuel Silva. 2018. Forecasting Changes of Economic Inequality: A Boosting Approach. Working Papers 201868, University of Pretoria, Department of Economics.

Rob Schumaker and Hsiu-chin Chen. 2009. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Trans. Inf. Syst.*, 27.

Matheus José Silva de Souza, Danilo Guimarães Franco Ramos, Marina Garcia Pena, Vinicius Amorim Sobreiro, and Herbert Kimura. 2018. Examination of the profitability of technical analysis based on moving average strategies in BRICS. *Financial Innovation*, 4(1):3.

Troy J. Strader, John J. Rozycki, Thomas H. Root, and Yu-Hsiang Huang. 2020. Machine learning stock market prediction studies: Review and research directions. *Journal of International Technology and Information Management*, 28:63–83.

A. Vellido, P.J.G. Lisboa, and K. Meehan. 1999. Segmentation of the on-line shopping market using neural networks. *Expert Systems with Applications*, 17(4):303–314.

Kang Zhang, Guoqiang Zhong, Junyu Dong, Shengke Wang, and Yong Wang. 2019. Stock market prediction based on generative adversarial network. *Procedia Computer Science*, 147:400–406. 2018 International Conference on Identification, Information and Knowledge in the Internet of Things.

Xiao Zhong and David Enke. 2019. Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financial Innovation*, 5.