Wojciech Świercz

e-mail: 187223@ue.wroc.pl

ORCID: 0009-0002-1906-842X

Radosław Szostak

e-mail: 187234@ue.wroc.pl

ORCID: 0009-0006-0156-7297

Wrocław University of Economics and Business

Comparative Analysis of Predictive Models in Stock Market Price Forecasting

DOI: 10.15611/2024.90.1.05

JEL Classification: C53

Quote as: Świercz, W., Szostak, R. (2024). Comparative Analysis of Predictive Models in Stock Market Price Forecasting. In M. Pauka, T. Słoński (Eds.), *Finanse* (pp. 60-78). Publishing House of Wroclaw University of Economics and Business.

Abstract: In this paper the authors test ARMA, ARIMA and LSTM neural network's model performance on one minute stock market data. Simulation of a random walk is also performed. Models are adjusted and/or trained on S&P500 data split 80:20. Test is performed on last 20% of S&P500 data and stocks: AAPL, 3M, GM. Correlations were checked to make correct conclusions. Out of all models ARIMA model performed best, achieving in some instances R2 score as high as 0.99996. All models performed well, with Random Walk simulation performing the worst.

Key words: ARMA, ARIMA, Neural Networks, stock market forecasting

1. Introduction

The stock market plays an important role in national economies by facilitating the exchange of capital, impacting macroeconomic and microeconomic activities. Furthermore, accurate stock price predictions are essential for optimal resource allocation and directing capital to the most profitable investments. Stock market participants need to determine the real value of stocks to make better decisions. Throughout the years many methods have been created on how to estimate the stock's value.

Numerous studies (Narayan et al., 2015; Fama & French, 1988, 1992) have shown that stock returns can be predicted using financial variables like book-to-market ratio, dividend yield, annual returns, price earnings ratio, term spread, default spread, trading volume etc. Although debates persist regarding the reliability of these predictions due to issues like spurious regressions, data mining, and return predictability instability, the general consensus in the literature is that stock returns have a predictable element (Rounaghi & Zadeh, 2016).

These essential tools, that are used for modeling financial models which involve predicting a variable arranged in chronological order are called Time Series Forecasting (TSF) methods. The objective is to forecast the system's behavior rather than understand its inner workings. Significant advancements in Operational

Research have introduced quantitative TSF methods, moving away from traditional intuition-based approaches. In the past twenty years, alternative nonlinear TSF techniques have emerged, with Artificial Neural Networks becoming particularly popular (Cortez & Rocha, 2004). Together with higher computational power, this allowed hedge funds to trade assets on shorter timeframes, with now time frames reaching milliseconds. Quant funds, which operate particularly in this sector, started specializing in utilization of machine learning and statistical model in asset trading.

On the other hand, auto-regressive moving average (ARMA), as well as auto--regressive integrated moving average (ARIMA) models are probably the most known and widely used methods of forecasting by amateur investors. These methods integrate autoregressive and moving average terms into an equation to create a model for forecasting new values. The autoregressive component links the future value to past and present values, while the moving average part connects the future value to the errors of previous forecasts. However, the ARMA and ARIMA methods are simplistic models that cannot detect complex, subtle patterns in time series data (Rounaghi & Zadeh, 2016).

In this work, the authors will test three models (ARMA, ARIMA, Neural Networks), which are commonly used in time series forecasting. A simulation of random walk will also be included, as it is often assumed that stock market prices follow a normal distribution (Burton, 1973). The models will undertake the predictions of prices for four different financial instruments. Subsequently, the authors will test the performance, compare pros, cons, assumptions and difficulties in calculating such models. From many methods of verifying model's performance the authors chose Mean Squared Error (MSE), Mean Absolute Error (MAE) and R-squared (R^2).

MSE indicates how close the predictions are to the actual data, with lower values representing better accuracy by measuring the average of the squares of the errors between predicted and actual values. On the other hand, MAE provides insight into the accuracy of predictions, by giving the value of mean deviation between actual and predicted values in the predicted period and finally, R-squared, with higher values indicating a better fit of the model to the data. It represents the proportion of the variance in the dependent variable that is predictable from the independent variables.

2. Methodology

Data and theory behind stock picking

The authors chose three different stocks and one index, prices of which were recorded as one minute opening prices, all from NASDAQ market. It has been suggested that mature markets are more suitable for predictions, compared to emerging markets (Pang et al., 2018). The financial data used in this study was obtained from the SAFE Data Room and Bloomberg Terminal (2024) at Goethe University Frankfurt (SAFE Data Room and Bloomberg Terminal, 2024). The S&P500 Index is representative of the biggest 500 NASDAQ stocks, it has very high liquidity, thus being less volatile. As it represents stocks combined, the authors decided to use it for model training. Three stocks were chosen randomly, from the S&P500 index, each representing a different sector. A pseudo-random algorithm was used. Sectors may differ in trends, that is why authors decided not to use only one specific sector. Results obtained from stock for a specific sector should be consistent for other stocks in the discussed sector, thus making the findings of the article more useful. Stocks used for algorithmic trading should have high liquidity and the S&P500 index includes a diverse range of large-cap companies, that is why it is an excellent source for stock picking. The findings of the article should only be applicable to mature markets, and large-cap companies, with high liquidity, and low volatility. Trends amongst markets from different countries may also differ; however, the authors did not test the model performance with other markets.

The time frame is chosen to minimize the lag that is observed when training model on larger time frames. In a paper proposed by (Zhang, 2023) it was observed, that with minimizing the time frame (in case of neural networks), the lag also minimized, thus making predictions more useful. All models were trained on S&P500 index, split 80 to 20, for train and tests sets respectively. All stocks data was used for testing in case of neural networks, and 80 to 20 split was made for ARIMA models. Obtained data pertains to the period from 2023-05-09 to 2024-05-23. Data set is perceived as small for the purpose of model training, on the other hand it provides faster training, which means less resources used.

Tools

All data splitting, normalization, evaluation of the models, etc. was conducted in Python and MS Excel, with the use of multiple libraries and extensions. For performance evaluation Sklearn was used. For neural networks, Tensorflow and Keras were used. Figures were created with the use of Matplotlib. For ARMA and ARIMA models Statsmodels package was used.

Comparison methods

To compare models, three statistical measures were chosen. R2 is a measure commonly used in accuracy testing, which can also be used on normalized data, and is easy to interpret (Dziechciarz & Błaczkowska, 2003). It is important to note, however, that in the case of following research, this does not have to be between 0 and 1, as it can also reach negative values, due to model being evaluated not on the data it was fitted to (Nian Wei, 2022).

$$R^2 = 1 - \frac{SSR}{SST},$$

where: *SSR* – explained sum of squares; *SST* – total sum of squares.

A score of 1 means that the model makes perfect predictions. The lower the score, the smaller proportion of variance is predicted. Negative values simply mean that model makes terrible predictions.

MSE, which translates to Mean Squared Error, is a metric that is commonly used in machine learning, as it heavily penalizes large errors. It is simply an average squared difference between actual and predicted values. Low MSE means that model performs well with values predicted being close to actual values. High MSE could mean that there are many outlier errors influencing the score.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2,$$

where: n – number of observations; y_i – actual value; $\hat{y}_i$ – value predicted.

MAE (Mean Absolute Error) is yet another well-known measurement, one which is more resilient to outliers than MSE. MAE does not take into account the direction of the errors, hence making it easier to interpret. Interpretation is to be concluded in the same units as the units of data. The lower the overall score, the better predictions model makes. A higher score means that the average absolute value of errors is bigger, thus encouraging to adjust the model.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|e_i\right|,$$

where: n – number of observations; e_i – errors.

2.1. Random Walk (RW)

It is often assumed that stock prices have a normal distribution, and follow a random walk pattern. The term random walk has been popularized by Burton Malkiel in his book “A Random Walk Down Wall Street”, and since then, there have been many debates and research on whether the theory is righteous. Recent findings, like (Durusu-Ciftci et al., 2019) seem to have a different outcome, depending on the statistical method used. Other researchers undermine the theory of RW, and propose alpha stabile distributions as an answer to stock price movements, also trying to estimate said distributions (Król, 2004).

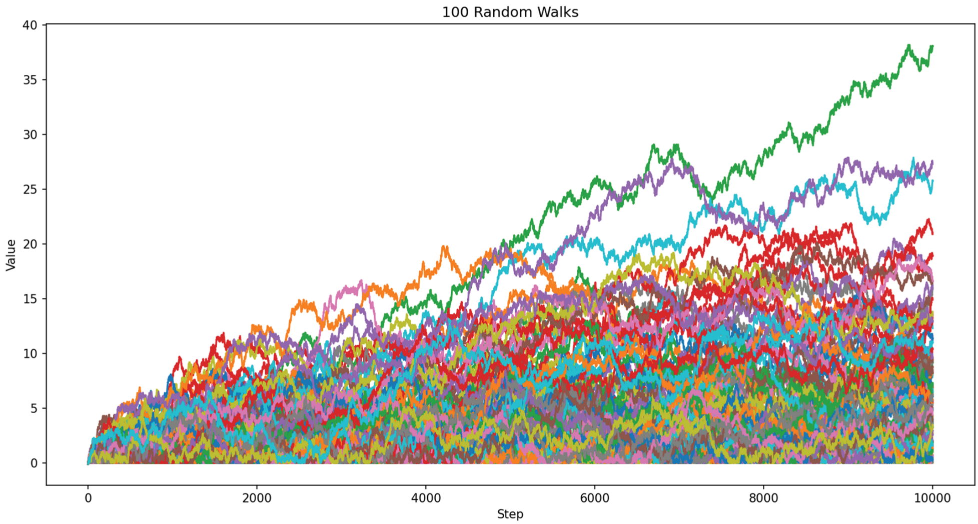

Figure 1. Random walk

Source: own elaboration.

For this research, the authors will use the theory of RW. Looking at the Figure 1 of 100 random walks, one might think that the theory of RW is correct, as some of the lines seem to resemble stock price movement. The authors will check, if predictions made with random walk, as a continuation of 80% of stock price movement data will be in any way correct.

2.2. Neural Network

When it comes to neural networks, there is a great number of architectures that can be tested. Not only that, but developers can also choose from a varying range of optimizers, parameters, etc. A type of network which is commonly used in stock market prices forecasting are Recurrent Neural Networks. Their advantage over architectures like Convolutional Neural Networks comes from the direction of the flow of the information.

There have been many attempts at trying to capture the best parameters of a model, for example (Maknickienė et al., 2011) in her work tries to capture the most optimal number of neurons, epochs and amount of data. However, it is important to note that models are trained on different data and may have varying performance with proposed parameters. The authors of the article have settled on an architecture consisting of 5 layers in which one layer is a Long Short Term Memory layer, 121 neurons and 170 epochs. LSTM layers are often used when modelling time series data, as the amount of data can be extremely large, which can lead to gradient vanishing or explosion. Architecture of LSTM layer protects from gradient vanishing

(Johnson et al., 2017). The sequence length for the data is 110, which conforms to findings of (Maknickienė et al., 2011). For this article, the authors found that a higher number of neurons than advised gave better predictions. To prevent from overfitting, two dropout layers are added. Loss function is MSE and optimizer is Adagrad.

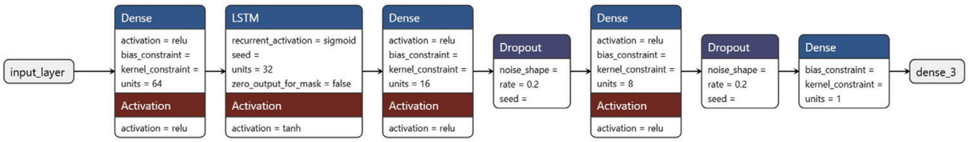

Figure 2. Network architecture

Source: own elaboration.

Data for S&P500 index was sliced into training (80%) and testing set (20%). Predictions on 3M, GM and AAPL were made with model pretrained on S&P. As for 3M, GM and AAPL predictions were made on the data of dates, on which model was trained for the S&P index, it is important to check correlations, so as to see if the results given are not corrupted by high correlations, thus having artificially better predictions on historical data.

Table 1. Correlations

	3M	AAPL	GM
S&P500	0.496	-0.078	0.810

Source: own elaboration.

As seen in Table 1 GM displays high correlation with the index. With that in mind, GM will be tested on full and splitted data set. 3M shows moderate correlation, however not high enough to influence the score dramatically. AAPL stock has a very small, negative correlation, which cannot have any influence on the score. All stocks with similar correlations, across all markets, should give the same or close results, to the ones obtained in the article. This is the only generalization, regarding different markets that one can be sure of.

2.3. ARMA and ARIMA models

An Autoregressive and Moving Average (ARMA) model is the combination of autoregressive AR (p) and moving average MA(q) models. ARMA model has been used to forecast stock market series (Adebayo et al., 2014) as well as non-financial phenomena like, for instance, short-term rainfall (Burlando, Rosso, Cadavid & Salas 1993). The general form of ARMA model is:

$$\hat{Y} = \Theta + \alpha_1 Y_{t-1} + \ldots + \alpha_p Y_{t-p} + \beta_0 u_t + \beta_1 u_{t-1} + \ldots \beta_q u_{t-q},$$

where: $\hat{Y}$ is the value at time t, Θ is a constant term, α and β are the coefficients. For the ARMA model to be applicable, the time series data must be stationary, meaning the mean, variance are constant over time and data is not seasonal (Sin et al., 2020).

If the prices of the financial instruments seem to be stationary, an ARMA model will be estimated. To verify stationarity, the authors will use the ADF test. If the data appears non-stationary, it will be transformed using methods such as first differencing. This is the first step to convert ARMA into ARIMA model. The "I" in the model's name stands for "integrated". If this modified model does not adequately explain the actual data, the authors will employ other common transformations such as logarithmic, square root, cubic root, and seasonal differencing to reconstruct the model.

The model's structure will be defined by the analysis of the ACF and PACF diagrams as well as the AIC ratio. These are the most common methods for ARIMA models used i.a. by Huang (2022). The data has been divided into two parts. In--sample and out-of-sample periods to verify the model's accuracy. The in-sample period lasts from 23rd May 2023, 15:30 to 15th February 2024 22:00. The out-of--sample period is significantly shorter and it starts on 16th February 15:30 and ends on 21st May 2024 22:00. The division was conducted in order to estimate the model on the first data sample and test it on the second data sample. Finally the model will be tested by MAE and MSE scores.

3. Results

3.1. Random Walk

Random Walk simulation was performed only on S&P500 data, as if stock price distribution were to follow normal distribution, the choice of the dataset would not matter. The results provided by the simulation of random walks were not satisfactory.

Table 2. Random walk scores

Random Walk simulation		
Measure	Best	Worst
R^2	0.6431	−14.671
MAE	0.0341	0.2432
MSE	0.0017	0.078

Source: own elaboration.

After 100 trials, the best results gave score of 0.6431. It is crucial to note that the most of the simulations gave a lower score. Naive simulation of a random walk is clearly not suitable for estimating prices, but can be used for time series generating for methods like Monte Carlo. Figure 3 displays the worst and the best simulation.

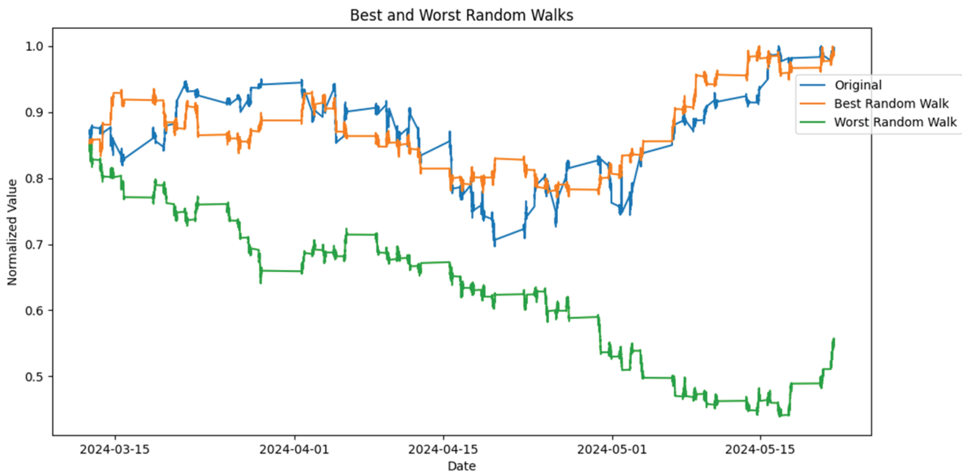

Figure 3. The best and the worst random walk

Source: own elaboration.

It can be observed that charts look similar to stock price movements. Still however, actual values are not close to the random walk simulations, and using RW to make predictions would be completely inefficient. If stock prices followed normal distribution, machine learning methods like neural networks would provide perfect predictions.

3.2. Neural Networks

Training score on S&P data reached of 0.985 and test score of −0.24. Test score is very low, which may indicate overfitting of the model to training data. To avoid it, more dropout layers can be inserted into the model, or a validation set can be introduced. Despite low test scores, the authors will verify the model on stocks data, as neural networks could have received sufficient information from training data. Figure 4 shows the curve of loss function.

Training of the network was not stopped, however it can be observed that around 120 epochs, there were no more significant loss changes, thus early stopping should be included in further research to protect from unnecessary use of resources.

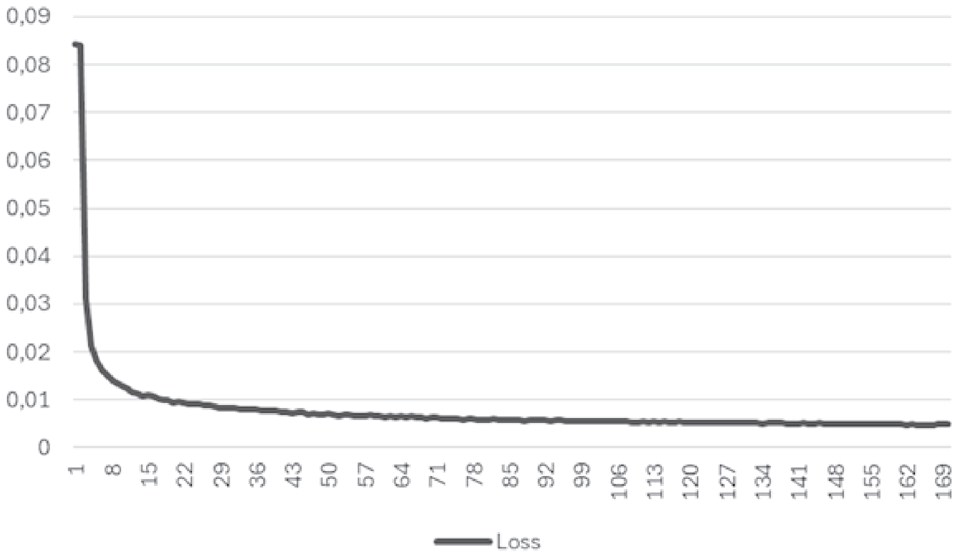

Figure 4. Loss

Source: own elaboration.

Figures 5, 6, 7, 8, 9 show the actual data vs. the predictions.

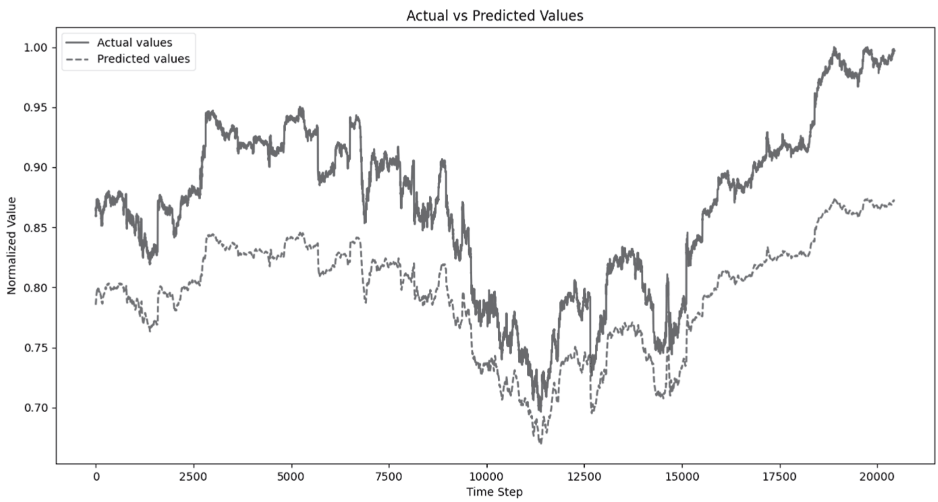

Figure 5. S&P500

Source: own elaboration.

After looking at Figure 5 it seems that despite values predicted being significantly lower, directions of price movements still looked to be correct. When creating a trading strategy, researchers may omit using actual prices, and focus only on percent changes, that is why the authors decided to test the model on stocks data.

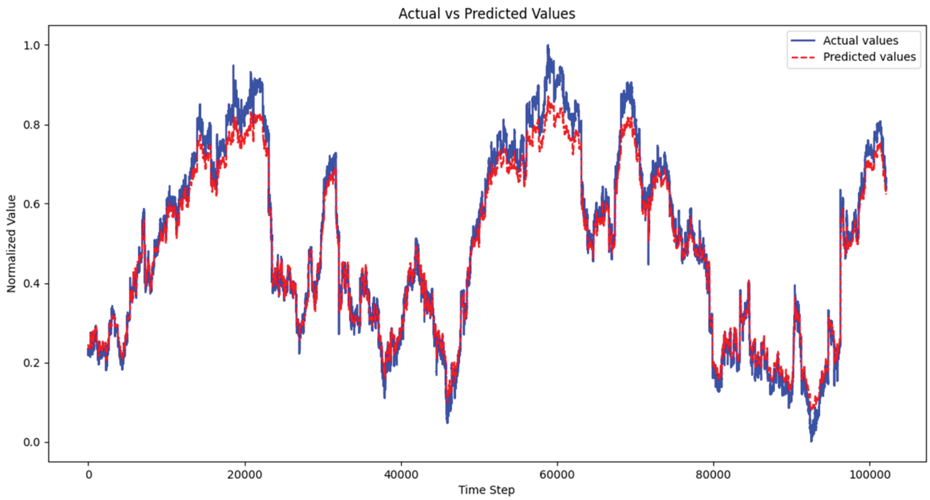

Figure 6. AAPL

Source: own elaboration.

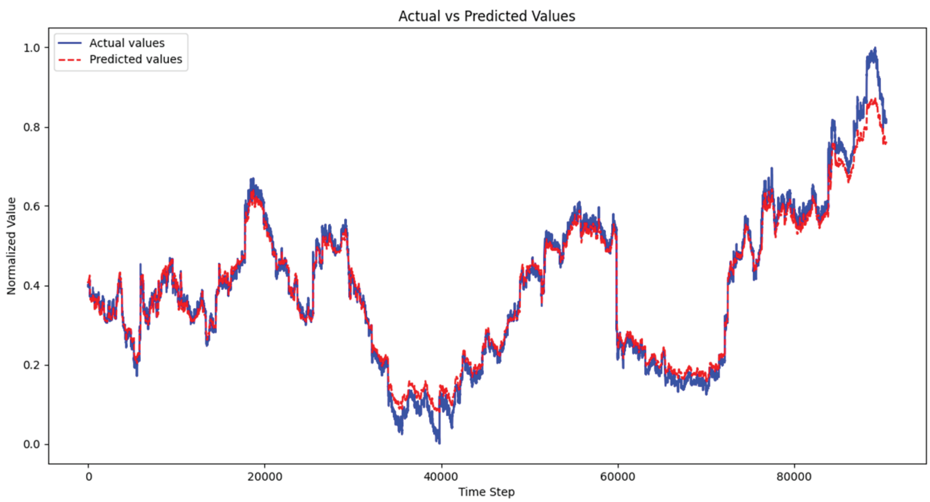

Figure 7. 3M

Source: own elaboration.

Despite low correlation of AAPL stock price movement which the data model was trained on, the model still predicts the data very well, better than the original test set. To avoid the possibility of partial autocorrelation affecting the score on the data of time index same as the training data, it was also tested on last 20%, but the obtained results were similar. The model seems to predict prices correctly, with all statistical measures, as seen in Table 3, prompting the conclusion.

Predictions made on 3M data are satisfactory, with big differences only observable at the end of the time period. The mentioned differences might indicate that the model needs more data to learn the movement better, although performance is still sufficient. As moderate correlation with training data was observed, the model was also tested on last 20% of the data, but obtained results were similar, with statistical measures showing great results. The model seems to predict prices correctly, with all statistical measures, as seen in Table 3, prompting the conclusion.

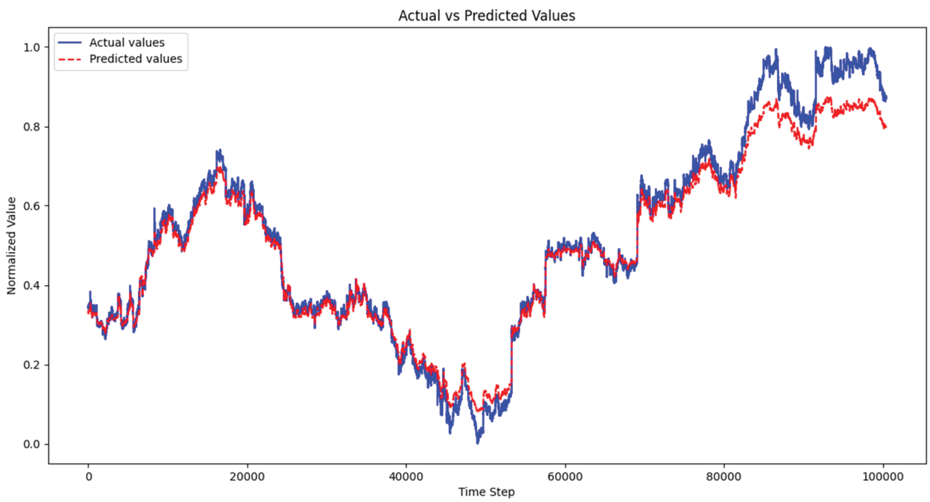

Figure 8. GM

Source: own elaboration.

Predictions made on GM may seem promising, however, as GM data was highly correlated with S&P500, performance needs to be checked also on the last 20% of the data.

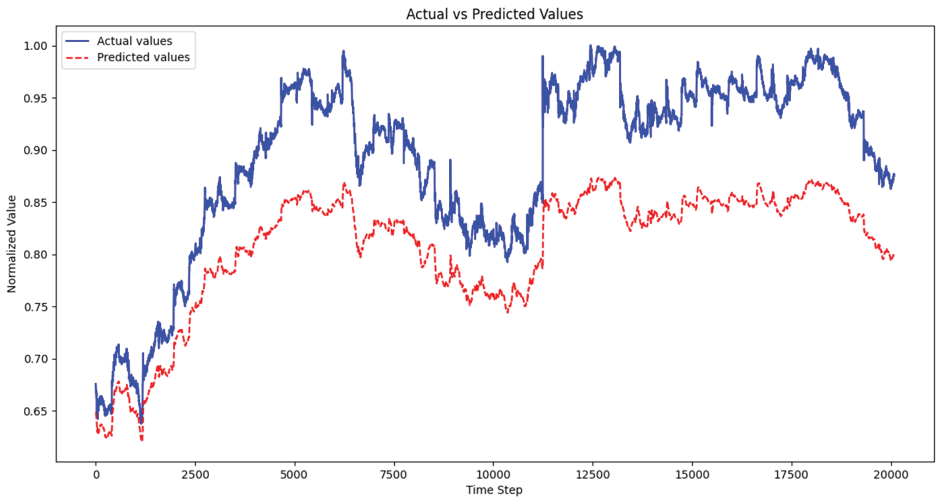

Figure 9. GM – 20%

Source: own elaboration.

Results after the split became significantly worse. It is due to data having high correlation, thus problem of overfitting occurring, and having its effect on the scores, despite model not being trained on GM data. Still directions seem to be correct, but a small lag can be observed in some parts of the plot.

RNN network provided good price prediction results with uncorrelated and moderately correlated data and terrible price predictions, but good directions with highly correlated data. However, as mentioned before, predictions of the direction are more important to potential researchers, as when creating a trading strategy, actual prices can be omitted.

Table 3. Test scores Networks

	S&P500	AAPL	3M	GM	GM – 20%
R^2	−0.242	0.979	0.983	0.970	−0.04
MAE	0.075	0.026	0.019	0.031	0.084
MSE	0.006	0.001	0.001	0.002	0.008

Source: own elaboration.

In the best instance, model explains 98.3% of variance. MAE of 0.019 and MSE of 0.001 indicate very accurate predictions. Obtained scores provide a favorable verdict, denoting neural networks as a tool suitable for predicting stock prices. Still, however, further research needs to be conducted, with possibly more data, and better prevention of overfitting.

3.3. ARMA and ARIMA models

Looking at the database for S&P500 index, it can be concluded that the data is not stationary.

Figure 10. S&P500

Source: own elaboration.

To achieve stationarity the authors used a common method to make data stationary by calculating the model on first difference (which is a difference between one value and the following one):

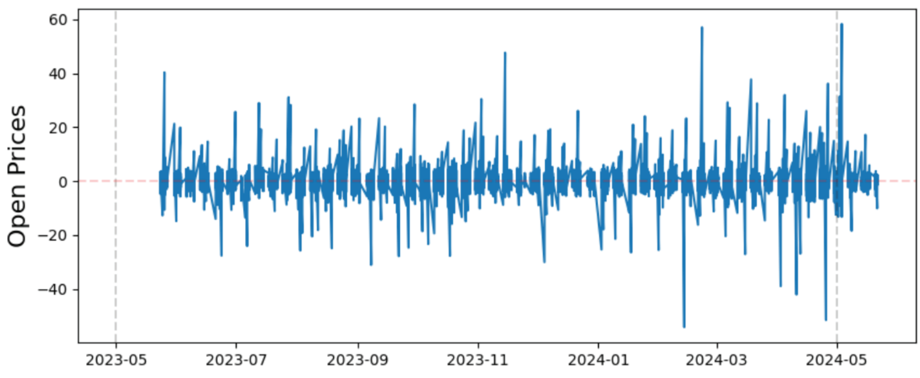

Figure 11. First differences

Source: own elaboration.

The ADF Test for Stationarity has been conducted. With ADF Statistic equal to −126.89 and Critical Value 1% at the amounts to −3.430 it can be concluded that the modified data is stationary. The model's structure has been based on ACF, PACF plots as well as the lowest AIC ratio the authors decided to create ARIMA (1, 1, 2) model for S&500 index which looks like that:

$$\hat{Y} = 0.071451 \cdot y_{t-1} + 0.999 \cdot u_{t-1} + 0.00025 \cdot u_{t-2}.$$

The model does not contain the constant value, because the *p*-value for the constant equals 0.96, which indicates the rejection for every acknowledged statistical significance in literature. This model explains the actual data quite well. It is visible on provided chart (see Figure 12):

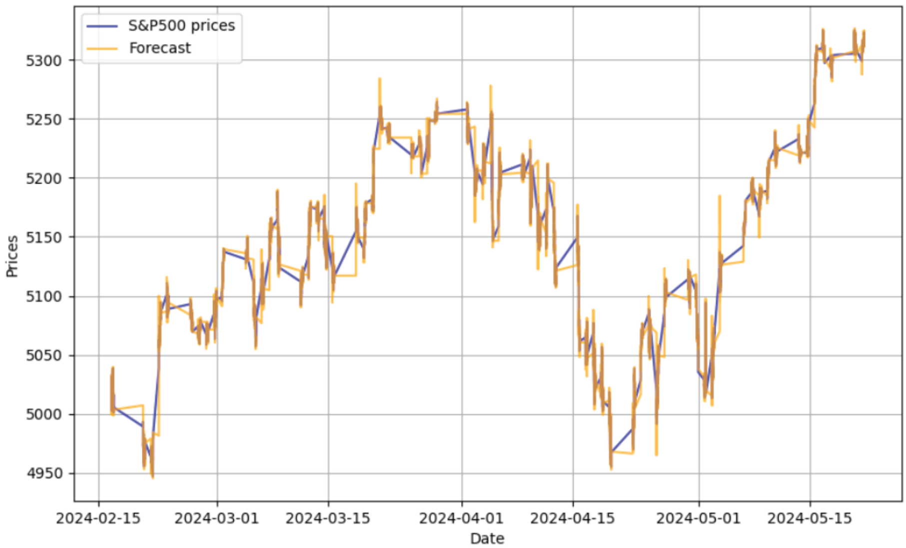

Figure 12. S&P Prices vs. Forecast

Source: own elaboration.

This ARIMA (1, 1, 2) model has a very high R squared, low MSE and MAE thus the authors decided not to find another way to transform the data and accept this model.

All of the presented steps were conducted for S&P500, 3M, Apple, GM prices and turned out to have similarly good outcomes. GM had ARIMA (1, 1, 2), Apple ARIMA (1, 1, 1) and 3M ARIMA (2, 1, 2).

Figure 13. 3M Prices vs. Forecast

Source: own elaboration.

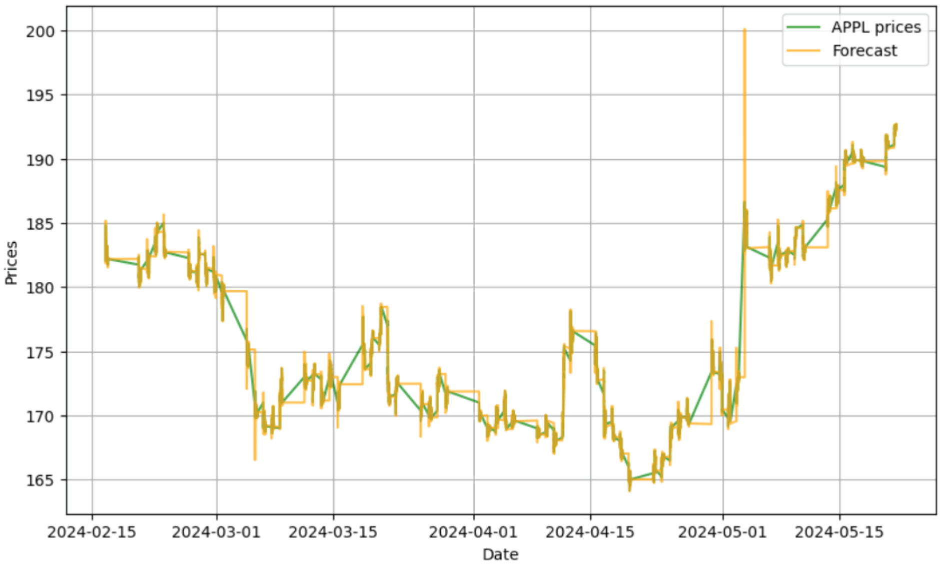

Figure 14. AAPL Prices vs. Forecast

Source: own elaboration.

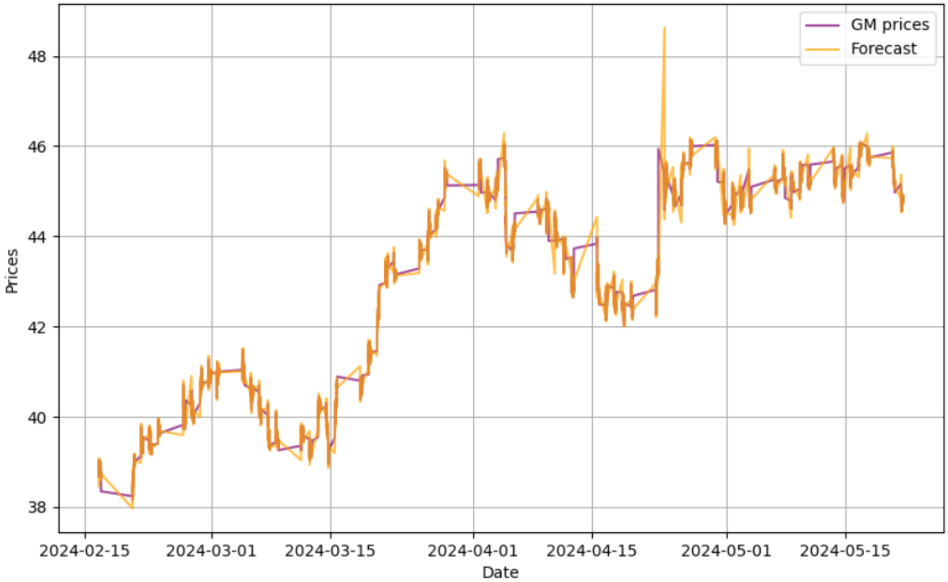

Figure 15. GM Prices vs. Forecast

Source: own elaboration.

Due to the different ARIMA models, in Table 4, p and q are written that symbolize number of coefficients for AR and MA parameters respectively.

Table 4. Test scores ARIMA

	ARIMA (p, 1, q)			
	S&P500	AAPL	3M	GM
R^2	0.99996	0.99956	0.99978	0.99989
MAE	1.40144	0.09505	0.05291	0.02789
MSE	4.79294	0.03146	0.01053	0.00274

Source: own elaboration.

The reason for such a good outcome is that there is a very big difference between actual prices of the financial instruments and their errors. Changes that occur in stock prices every minute are minor in comparison to the actual prices, therefore while calculating SST (total sum of squares), SSR (residual sum of squares) turns out to be very low and R^2 becomes considerably high. Since the models turn out to be successful one could claim that they are very useful when it comes to investing, but it is not that simple. The statistical metrics indeed claim that the models are great, however the predictions do not perform well when it comes to short-term trends. Trading using these models would turn out to be unprofitable due to the low efficiency of predictions. For instance, opening and closing positions according to the S&P500 model was successful only in 19% of cases. In 81% of cases the actual price of the index went in a different direction to the model. Neural networks achieved better results with about 50% success rate. The authors recommend to check the precision of direction of the prices being predicted correctly.

Discussion of findings

When comparing the performance of models proposed in the article with relevant literature, with respect to the errors of prediction, it can be observed that neural networks usually tend to perform better than ARIMA models and RW simulations (Darmanet et al., 2021; Murillo, 2018), which opposes the findings of our research paper. Such differences might occur due to the different architectures used, as well as different data pre-processing methods. (Kobiela et al., 2022) in their work found that ARIMA model performed better than LSTM based network. It might be due to the fact that the architecture proposed by them is undeniably shallow for a neural network – consisting of only 2 LSTM components. What is also crucial to be mindful of is the possibility to train the Neural network on multiple features, not only price data, which would likely increase the efficiency of the model. In terms of direction precision our network performance was better than that of ARIMA model, which also fosters its superiority over ARIMA and RW. The difference between errors was very low as well – different pre-processing methods would likely result in findings similar to that of (Darman et al., 2021; Murillo, 2018). To summarize, a Neural network with well constructed architecture and features will likely outperform ARIMA models and RW simulations in terms of stock market price prediction, leaving up to discussion the performance of simpler models.

4. Conclusion

ARIMA models are slightly better at forecasting the financial instrument's prices than Neural networks on one minute data, which require very large datasets to be able to learn price movements correctly. In terms of direction precision, neural networks seem to perform better than ARIMA models, thus indicating better potential for implementation in a trading strategy. The problem of overfitting in neural networks persisted, despite introducing dropout layers. Still however, both methods provide good results in terms of prediction measured by statistical indicators; however, it is mostly due to the errors being small in minute data relative to the actual value. In future works the authors will consider different kinds of autoregressive models like ARCH and GARCH with different data sets and try to create more reliable financial models that could be used in real life environment such as implementing the model into a trading strategy. Also, different architectures of neural networks will be tested, possibly on more data, with the inclusion of a validation set. Apart from neural networks, other machine learning methods should be inspected, such as Hidden Markov Models or combinations of different sets of models. Lastly, the authors will consider utilizing a DPA measure, as a potentially better way of verifying model's performance.

References

Adebayo, F. A., Sivasamy, R. & Shangodoyin, D. K., (2014). Forecasting Stock Market Series with ARIMA Model. *Journal of Statistical and Econometric Methods*, *3*(3), 65-77. Retrieved from https://www.scienpress.com/Upload/JSEM/Vol%203_3_3.pdf

Burlando, P., Rosso, R., Cadavid, L. G. & Salas, J. D. (1993). Forecasting of Short-Term Rainfall Using Arma Models. *Journal of Hydrology*, *144*(1-4), 193-211. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/0022169493901726

Burton, M. G. (1973). A Random Walk down Wall Street. *Financial Analysts Journal*, *29*(6), 15-20.

Cortez, P. & Rocha, M. (2004). Evolving Time Series Forecasting ARMA Models. *Journal of Heuristics* 10, 415-429. Retrieved from https://doi.org/10.1023/B:HEUR.0000034714.09838.1e

Darman, H., Ho, K. M. & Musa, S. (2021). Stock Price Prediction Using ARIMA, Neural Network and LSTM Models. Journal of Physics: Conference Series 1988, 012041. Retrieved from https://iopscience.iop.org/article/10.1088/1742-6596/1988/1/012041/meta

Durusu-Ciftci, D., Ispir, M. S. & Kok, D. (2019). Do Stock Markets Follow a Random Walk? New Evidence For An Old Question. *International Review of Economics & Finance*, 64, 165-175. Retrieved from https://www.sciencedirect.com/science/article/pii/S1059056018310979

Dziechciarz, J. & Błaczkowska, A. (2003). *Ekonometria – metody, przykłady, zadania*. Wydawnictwo Akademii Ekonomicznej im. Oskara Langego we Wrocławiu. Retrieved from https://www.researchgate.net/publication/301231419_Ekonometria_Metody_Przyklady_Zadania

Fama, E. F., & French, K. R. (1988). Dividend Yields and Expected Stock Returns. *Journal of Financial Economics*, *22*(1), 3-25. Retrieved from https://www.jstor.org/stable/pdf/2328716.pdf?casa_token=t3gFWJobnsEAAAAA:eoysSzvvyZBepqYbntRPBNuik0kIwjxf24KH7e1N67ZKVBRxKnQyDkuHjprV0yxCpmlLtn-1jZ_E0sNNYG5n5lxBHM6TKHRuYr_IhjBEYOySPMdrQd8

Fama, E. F., & French, K. R. (1992). The Cross Section of Expected Stock Returns. *Journal of Finance*, *47*(2), 427-465.

Huang, Y. (2022). Prediction of the Stock Price of Shanghai Securities Composite Index by Using Time-series Model. *In ICEEG '22: Proceedings of the 6th International Conference on E-Commerce, E-Business and E-Government* (223-228). Retrieved from https://dl.acm.org/doi/pdf/10.1145/3537693.3537729?casa_token=mPwJbaSuh-8AAAAA:8p1wNWHYrYM4sUyzFMu9wdW5bANCAmtR_RO5mdB_DOgmEkDe5CJcyYnr-L_WnOk1S91CScIBeX26

Johnson, J., Fei-Fei, L., Yeung, S. (2017). *Lecture 10. Recurrent Neural Networks.* Stanford University.

Kobiela, D., Krefta, D., Król, W., Weichbroth, P. (2022). ARIMA vs LSTM on NASDAQ stock exchange data. Procedia Computer Science, 207, 3836-3845. Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050922013382

Król, A. (2004). Rozkłady α-stabilne w analizie rynków papierów wartościowych: problemy estymacji. Monografie i Opracowania / Szkoła Główna Handlowa, 533, 147-157. Retrieved from http://bazekon.icm.edu.pl/bazekon/element/bwmeta1.element.ekon-element-000061794311

Maknickienė, N., Rutkauskas, A. V. & Maknickas, A. (2011). Investigation Of Financial Market Prediction By Recurrent Neural Network. Innovative Infotechnologies for Science, Business and Education, 2(11), 3-8. Retrieved from http://journal.kolegija.lt/iitsbe/2011/Maknickiene-investigation-IITSBE-2011-2(11)-3-8.pdf

Murillo, A. R. A. (2018). High-Frequency Trading Strategy Based on Deep Neural Networks. *Journal of Financial Technology*, *12*(3), 45-60.

Narayan, P. K., Sharma, S. S., & Thuraisamy, K. S. (2015). Can governance quality predict stock market returns? New global evidence. *Pacific-Basin Finance Journal*, 35, 367-380. Retrieved from https://www.academia.edu/21344954/Can_governance_quality_predict_stock_market_returns_New_global_evidence

Nian Wei, T. (2022). *Explaining negative R-squared*. Towards Data Science. Retrieved from https://towardsdatascience.com/explaining-negative-r-squared-17894ca26321

Pang, X., Zhou, Y., Wang, P., Lin, W., Chang, V. (2018). An Innovative Neural Network App-Roach For Stock Market Prediction. The Journal of Supercomput 76, 2098–2118 (2020). https://doi.org/10.1007/s11227-017-2228-y

Rounaghi, M. M., Zadeh, F. N. (2016). Investigation Of Market Efficiency And Financial Stability Between S&P 500 And London Stock Exchange: Monthly And Yearly Forecasting Of Time Series Stock Returns Using Arma Model. Physica A: Statistical Mechanics and its Applications, 456, 10-21. https://www.sciencedirect.com/science/article/abs/pii/S0378437116002776

SAFE Data Room and Bloomberg Terminal. (2024). [Financial dataset]. Leibniz Institute for Financial Research SAFE, Goethe University Frankfurt.

Sin, K. W., Sheng, H. H., Zhi, W. Ch., Ling, L. P., Dass, C. (2020). Forecasting Stock Price using ARMA Model. *INTI Journal*, *59*, http://eprints.intimal.edu.my/1480/1/ij2020_59.pdf

Stanford University School of Engineering. (b.d.). *Lecture 10 – Recurrent Neural Networks*. YouTube.

Zhang, K. (2023). LSTM Neural Network in Stock Price Prediction. In Proceedings of the 2nd International *Academic Conference on Blockchain, Information Technology and Smart Finance* (ICBIS 2023). Retrieved from https://www.atlantis-press.com/proceedings/icbis-23/125989740

Analiza porównawcza modeli predykcyjnych w prognozowaniu cen na rynku akcji

Streszczenie: W artykule autorzy testują efektywność modeli ARMA, ARIMA oraz sieci neuronowej LSTM na danych giełdowych z jednominutowymi interwałami. Przeprowadzono również symulację błądzenia losowego. Modele zostały dostosowane i/lub przeszkolone na danych S&P500, podzielonych w stosunku 80:20. Testy przeprowadzono na ostatnich 20% danych S&P500 oraz na akcjach takich firm jak AAPL, 3M, GM. Sprawdzono korelacje, aby wyciągnąć poprawne wnioski. Spośród wszystkich modeli najlepiej wypadł model ARIMA, osiągając w niektórych przypadkach współczynnik determinacji R^2 na poziomie 0,99996. Wszystkie modele osiągnęły satysfakcjonujące wyniki, przy czym symulacja Random Walk sprawdziła się najgorzej.

Słowa kluczowe: ARMA, ARIMA, sieci neuronowe, predykcja rynku akcji