DOI: 10.52950/ES.2024.13.2.005

STOCK MARKET PREDICTION USING GENERATIVE ADVERSARIAL NETWORK (GAN) – STUDY CASE GERMANY STOCK MARKET

MICHAL MEC, MIKULAS ZEMAN, KLARA CERMAKOVA

Abstract:

Using neural networks in economics time series data is a new unexplored field. Lot of companies and economics research use mostly logistic regression or statistical approach when they try to predict the movement of stock market. Neural networks became a frequent tool for prediction in recent years and this approach has been confirmed to provide more reliable and better solutions when it comes to prediction and accuracy power. Within a wider context of current debate on neural networks employment in stock market predictions, we suggest an innovative methodology based on the combination of neural networks. In our analysis we use Wasserstein Generative Adversarial Network (WGAN) on Germany stock market as an example. We present how the trading strategy could be established on the prediction of the model and how it can be compared with other models in terms of returns. Overall, the WGAN monthly prediction outperformed Random Forest by 36%, benchmark by 32% and LSTM by 26% in the testing period. Our results also suggest that the WGAN model has on average higher returns than pure investment into index. Furthermore, WGAN is less volatile, which is always the preferred option for investors. Using neural networks for stock index prediction and confirming that WGAN investment strategy brings higher returns compared to generally used models is our main contribution to the current debate.

Keywords:

Generative adversial network; Germany stock market; neural network; stock prediction

JEL Classification: G11, G17, C45

Authors:

MICHAL MEC, Prague University of Economics and Business, Prague, Czech Republic, Email: michalmec@gmail.com
MIKULAS ZEMAN, Prague University of Economics and Business, Prague, Czech Republic, Email: mikulas.zeman@vse.cz
KLARA CERMAKOVA, Prague University of Economics and Business, Prague, Czech Republic, Email: klara.cermakova@vse.cz

Citation:

MICHAL MEC, MIKULAS ZEMAN, KLARA CERMAKOVA (2024). Stock market prediction using Generative Adversarial Network (GAN) – Study case Germany stock market. International Journal of Economic Sciences, Vol. XIII(2), pp. 87-103., 10.52950/ES.2024.13.2.005

1. Introduction

It goes back to the 20th century since investors try to find various ways to predict stock market. The academic literature concluded that stock market price is nonlinear, nonstationary, nonparametric, and noisy (Abu-Mostafa & Atiya, 1996). Zhong and Enke (2017) discuss that these characteristics are caused mainly by political, economic, psychological and company specific environment. In the past, investors used to use only fundamental and technical analysis to achieve maximum profits with the lowest risk in their decision-making process (Arévalo et al., 2017). Since then, a lot of research has focused on improving the prediction of stock market prices. It can be distinguished between three categories how to predict the stock market price:

1. *Statistical approach*
2. *Pattern recognition*
3. *Machine learning*

Statistical approach is described by Zhong and Enke (2017), who discuss models such as Auto-Regressive Integrated Moving Average (ARIMA), Generalized Autoregressive Conditional Heteroskedastic (GARCH) volatility and Smooth Transition Autoregressive (STAR) model. These models do not include volatility clustering which can be considered as key limitation for stock market prediction.

Pattern recognition and machine learning focuses on recognition of patterns and trends in the data. The main difference between these statistical approaches and pattern recognition is in the methodology. Patterns in stock can be found out by Open-High-Low-Close (OHLC) model (Velay & Daniel, 2018). These technical analyses depend on the patterns found out in historical stock price data (Nesbitt & Barrass, 2004) and they can inform investors how the price will behave in the future following the same pattern. Such an approach cannot be used when the markets have anomaly behavior. This often leads to significant losses of the investors.

Machine learning can be divided into two groups of models. In unsupervised learning only unlabeled or observed data is available. Algorithm tries to find a pattern, correlation, cluster of data or any combination. In supervised learning, the model has labeled input data for training at disposal. Aim of the algorithm is to train automatic approach to map the input data to the given output data. The research is focused on many machine learning algorithms for stock market prediction. Simple techniques such as naive bayes, and single decision tree was replaced by Random Forest (RF), Logistic regression (LR) used often to estimate portfolio risk (Mohammadi et al, 2022), and neural networks (Ballings et al., 2015). This is possible due to larger computational power and better quality and quantity of the data which we have at our disposal. Neural networks have become a very popular method in stock price prediction these days. Most of the research is focused on deep nonlinear neural network (Bao et al., 2017).

As a contribution to this discourse, our paper offers a brief introduction to machine learning and specifically the use of neural networks on financial markets. Within this debate the paper contributes by presenting an application of neural networks as predictors of stock market index. Current literature generally omits developing suitable strategies that would ensure high returns on investment and is limited to statistical deviations and losses description. Papers usually focus on comparison of prediction value and real value at the same time without relating future values to present values. We consider this gap crucial in investment decision-making. In this research paper we offer, within a wider context of generally used investment strategies, an innovative methodological approach consisting of employing neural networks with multiple features in an investment strategy.

2. Literature review

Generative adversarial network (GAN) was introduced by (Goodfellow et al., 2014). It is a framework for estimating generative models through adversarial process thus two models are trained simultaneously. First, Generator (G) which estimates distribution of data. Second, Discriminator (D) compares real data and predicted data and estimates the probability if the data are real or predicted. The goal of the G is to maximize the probability of D to make a mistake in the training process. In other words, G and D are playing min-max game where the unique solution exists. Arjovsky et al. (2017) tried to solve problems of traditional GAN such as model collapse or not sufficient process of the minimizing difference between distribution of real data and fake data. They developed extension of the GAN by improving loss function of D. They use Wasserstein loss function which provides a more stable training process. This is less sensitive to hyperparameters and model architecture, thus that is how the Wasserstein GAN was introduced.

In this article, we establish Long Short-Term Memory (LSTM) as G. LSTM is a deep learning model which can learn long-term dependencies. It contains cells, input gate, output gate and forget gate. All internal units of LSTM have hidden state augmented nonlinear mechanism. This mechanism can be reset or updated using learned gradient functions. Prediction of the stock prices using LSTM was researched by Selvin et al. (2017), Nelson et al. (2017) and Roondiwala et al. (2015). All authors claim that LSTM has high accuracy of prediction in comparison with other models. Siami-Namini and Namin (2018), Rhanoui et al. (2019) and Sirisha et al. (2022) compared the results of LSTM and ARIMA performance of prediction on financial market. These authors come to conclusion that LSTM outperforms ARIMA. Results that will be presented by this paper suggest that ARIMA gives inferior results in comparison with LSTM. Many studies confirm results of LSTM superior to ARIMA thus, if the GAN outperforms LSTM, we assume that it outperforms ARIMA also. Thakkar and Chaudhari (2021) article focuses on comparison of different neural networks and their performance in prediction of direction of the financial market. LSTM shows best results when it comes to handling non-linearity data as compared to other neural networks. Fischer and Krauss (2018) compare the LSTM with Random Forest, simple logistic regression and standard deep net. They found out that LSTM outperforms mentioned models by very clear margin. Shah et al. (2018) recommends to use LSTM in favor of deep neural networks also. In conclusion, LSTM can be considered as one of the best predictors of financial time series data, and we conclude that is suitable to use it as G in our model.

Convolutional neural network (CNN) is used as D. CNN is an artificial neural network which is used mainly at visual imaginary. CNN is composed of an input layer, multiple hidden layers, and output layers where hidden layers include mostly multiple combination of convolutional layers, pooling layers, fully connected layers, and normalization layers. I decided to use this neural network as a discriminator. This allows us to recognize and distinguish long-term trends and short-term trends. Another reason is that CNN is a great model for spatial data. CNN was used in the previous GAN architecture by Zhou et al., (2018) and Tovar (2020). Thakkar and Chaudhari (2021) claims that CNN is great for identification of patterns in the data. This is very suitable when the prediction is focused on the categorical problem, respectively when we want to predict if the market moves up and down. That’s how we approach CNN in our proposed WGAN. Kirisci and Cagcag Yolcu (2022) claims that CNN is better predictor than LSTM, ANN and fuzzy-based approaches. Many authors tries to combine LSTM with CNN to achieve better results (for example Lu et al. (2020), Widiputra et al. (2021), Mehtab and Sen (2022), Wu et al. (2023)). The main difference between these articles and ours is that they are using them separately or as a combination. Our methodology forces both neural networks against each other which should in theory improve our results. Overall, when we want to predict categorical variables, the best neural network for this is CNN.

GAN methodology is often used in image processing. This process is verified in a lot of research. As GAN is trying to generate data like original data, we can apply this methodology in the time series data. In the recent year there are multiple articles which applies this methodology on the time series data (for example Vuletić et al. (2023), Sonkiya et al. (2021), Meng et al. (2023) and Diqi et al. (2022)). Most of the articles produce prediction of the target variable from the target variable and they are not using multiple features. This article includes multiple other features, which should help the model to predict the explained variable more precisely. The main disadvantage of GAN is that it is hard to train. The loss function has no close form and requires a lot of trial – error approach, which concludes optimal setup of hyperparameters. Also, as the GAN uses neural networks, it is impossible to find out how much weight is put on each variable, thus we face the problem of the black box functions. Another problem of the GAN is that it is approximating to the local optimal value. This can cause that if the explanatory variable has very large value, the model will often predict decrease of this value in the next period. This can be countered by using more explanatory variables, which can set the optimum above the current value of the explained variable.

GAN models have recently become very popular on financial markets. Investment hedge funds and pension funds focus their investment strategies into ETFs and country indexes without relying on own stock picking strategy. Most of these companies have set up default strategies from which portfolio managers can differ by few percent. GAN models may be used to manage portfolios to get higher returns. Our methodology of using WGAN with multiple features aims on offering a new approach to financial models' application, that may substitute commonly used models such as Random Forest, Logistic Regression or LSTM.

3. Methodology

This chapter offers an elaborate description of used methodology, the feature selection process and how the model can be controlled for correct behavior. Our model includes multiple (24) features distributed in 12-time horizons for prediction of 3-time horizons ahead. Previous studies and GAN applications mostly rely on lagged price and identical input and output time horizons.

Our model composes G of one LSTM layer with 12 input units, hidden units and one fully connected dense layer with 3 output units. I used LeakyReLU as our activation function and L1 regularization loss. We put training and validation dataset as an input into G. Then output of G (prediction of input dataset) serves as input data for D together with the real data. D is composed of three one dimensional conventional layers. Convolution one, two, three are created by filter 32x1, 64x1 and 128x1, respectively with strides equal to one. The activation function used for all convolution layers is LeakyReLU. Batch normalization is used only on second and third convolution layer. Another layer is a dense layer with 220 units, activation function is LeakyReLU and batch normalization is applied also. Next, another dense layer with 220 units with ReLU activation function is used. Output layer is last. When this process is done the model predicts stock price from the test dataset and the mean squared error is counted. Next, for optimization of hyperparameters we use Bayesian optimization. Optimization process choose hyperparameters which has the lowest loss core of the validation dataset. We took the average of hyperparameters values from the trials which shows the lowest score. The Bayesian optimization is considered as best hyperparameter optimization method by various research (Cho et al. (2020), Putatunda & Rama (2019), Masum et al. (2021). Optimization of the parameters has a significant impact on the results. This impact is important when it comes to level of the prediction. If the hyperparameters are set correctly, it will yield improved result as the prediction is closer to real values. The method is used for following hyperparameters:

1. *G - hidden units*
2. *G - LeakyReLU alpha*
3. *D - LeakyReLU alpha*
4. *D - RMSprop learning rate*
5. *D – Clip constraint value for estimating weights*
6. *D – Batch normalization momentum*
7. *D – Truncated Normal initializer*
8. *GAN - RMSprop learning rate*
9. *Model – Epochs for training of the GAN*
10. *Model – Epochs for training of the G and D*

Using different values of these hyperparameters we have obtained a surprising result on our dataset, indicating that hyperparameters have no significant impact on the slope of the prediction, but on level of the prediction.

4. Data and training of the model

Neural networks need a lot of data for training, and it is very difficult to find out such a data which have long time series for any market. The best time series data to use for this methodology are on daily basis and have a long history. What we have on the financial market are mostly monthly data and 30 years history, which gives us around 360 data points. To have more data points for this article, I selected most of the data which has effect on the price of the shares from the article Celebi & Hönig (2019) and Karakostas (2023). We also included the data from FRED, which has close impact. DAX index is considered as explained variable in the most articles. For this article, we did not consider this index, as this index started in the year 1988, which gives us shorter time series than we need. For this reason, we selected as explained variable Share price/broad from FRED database, which represent larger portion of the companies on the market, and it should represent movement of entire economy more precisely than DAX index in theory. Rezabek et al. (2022) chooses similar data for the V4 countries, which are closely economically connected to German market. Pavelka and Löster (2014) highlight the importance of financial market on the unemployment. On the other hand, we need to look at unemployment as one of the main predictors of stock index prices, as this is a significant indicator of the performance of the economy, how the companies can maintain the number of employees. This is more important in the short run, as labor is only variable factor of production according to economic theory.

The data considered in this model are following:

Name	**Source**
Share Prices: All Shares/Broad	FRED
Capital account	Bundesbank

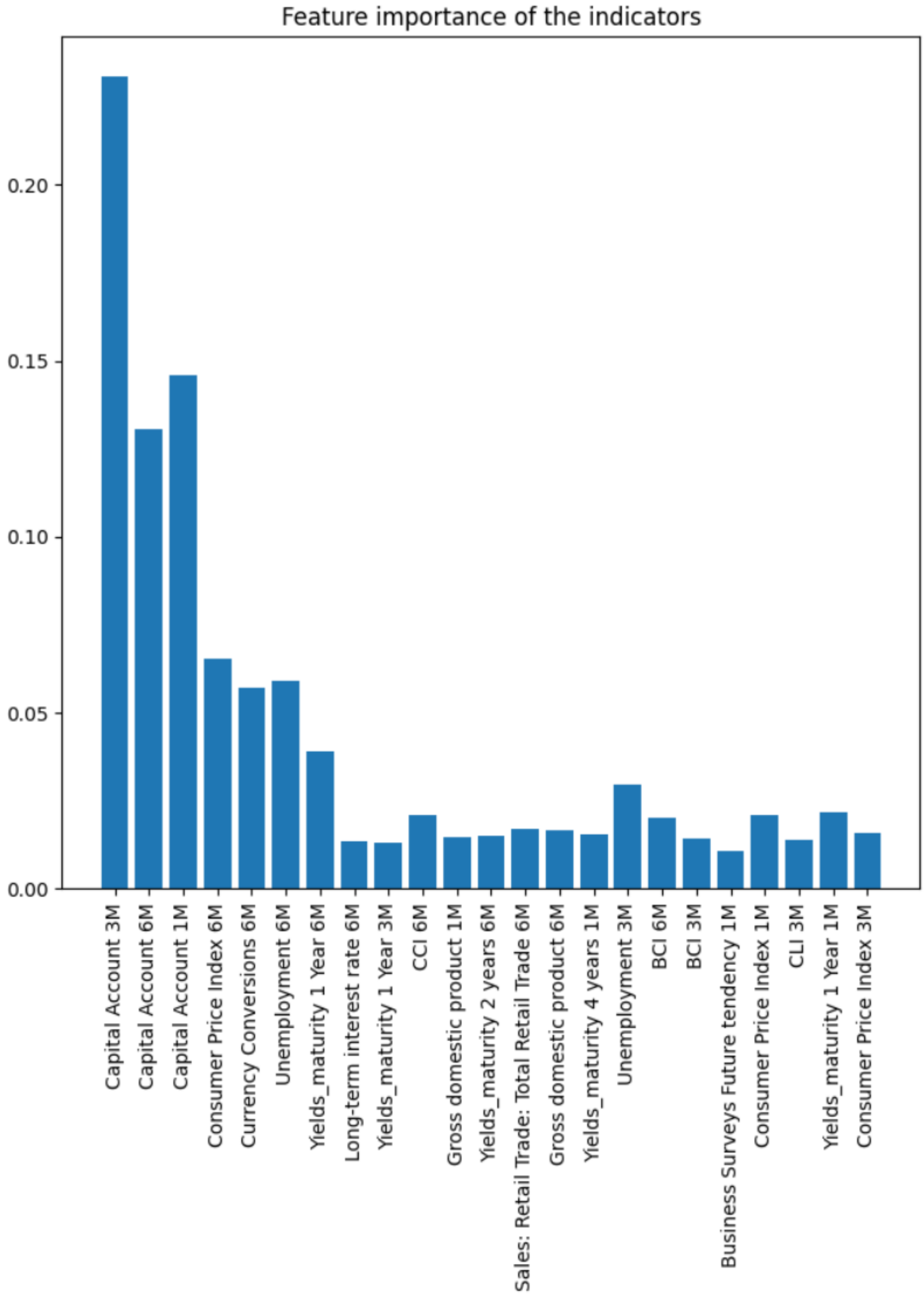
Consumer price index	FRED
Yields maturity 1 Year	Bundesbank
Yields maturity 2 Year	Bundesbank
Yields maturity 3 Year	Bundesbank
Yields maturity 4 Year	Bundesbank
Yields maturity 5 Year	Bundesbank
Consumer Confidence Index	OECD
Business Confidence Index	OECD
Composite Leading Indicator	OECD
Long term Interest rate	OECD
Business Tendency Surveys (Manufacturing): Production: Future Tendency:	FRED
Currency Conversions: US Dollar Exchange Rate: Spot, End of Period: USD: National Currency for Germany	FRED
Sales: Retail Trade: Total Retail Trade	FRED
Unemployment rate	FRED
Gross Domestic Product (real)	Bundesbank

As almost all data are not seasonally adjusted, the seasonal adjustments were done by Seasonal decomposition using LOESS build up in stats models. Also, all explanatory data were transformed on one month, three months and six months changes. Data has constant mean after the transformation. Furthermore, it gives us lagged values and more variables from which we can select the best features. Next, we use XGBoost tree methodology for selecting the most important features. The process of selection of features was done by two conditions:

1. *Importance of the variable is more than 1% and less than 30%.*
2. *The cumulative importance is less than 90%*

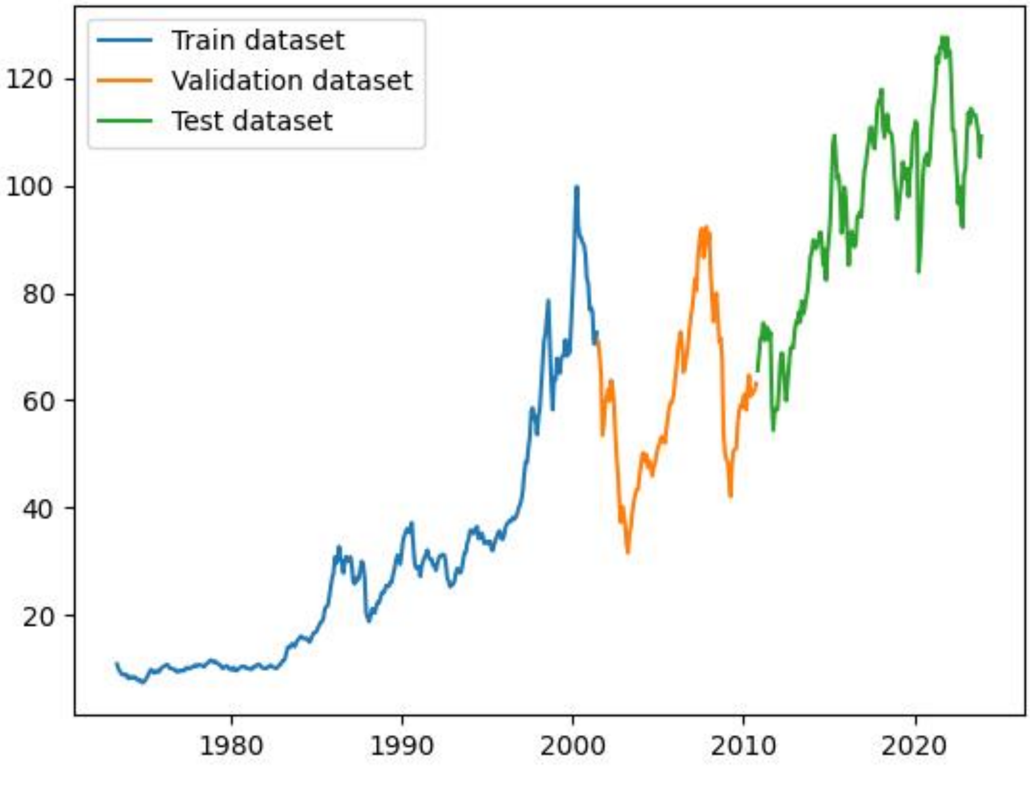
After that the algorithm was run once more to check if any feature has less than one percent of importance among themselves to eliminate the feature, which have the same information value. The important results and selected variables are shown in Figure 1.

Fig. 1. Feature importance



The data was split into three separate datasets. The data range starts at 30.10.1973 – 30.11.2023 and there are 602 data points. Training datasets consist of 50% of the data, validation dataset of 25% of the data and test dataset is the last 25%. All data are normalized by zscore methodology for each dataset separately. This should achieve more stable results, as the financial markets are more volatile in the test period and validation period in comparison with the training period.

Fig. 2. Data split



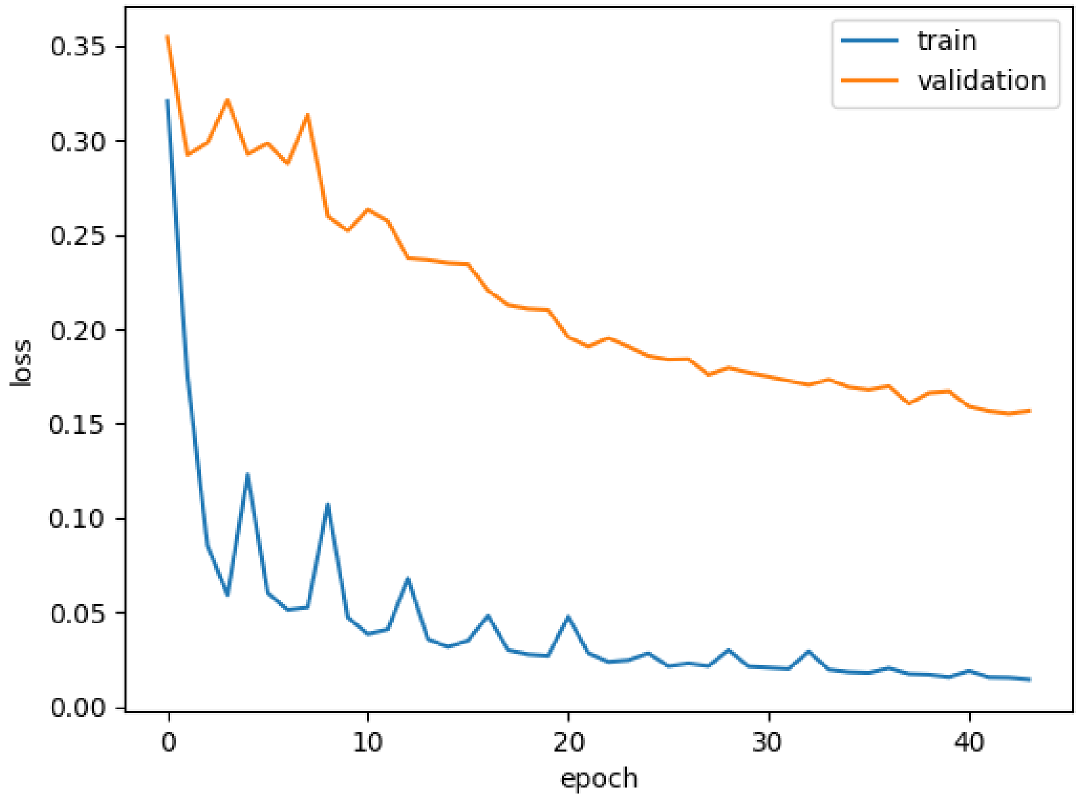
It may be observed that the performance of German stock market in training period suffered only mild downfalls in comparison with validation dataset and test dataset. Data show that in Validation dataset the trend is stable without any positive slope on average.

4.1. Visualization of WGAN results

As the GAN faces multiple problems in the training period, it is required to check that the prediction and training process is valid. The GAN has a certain randomness in predicting results due to the complexity of the model, which includes different estimated optima. Thus, every run yields a little bit different result. For this reason, is optimal to run the model multiple times independently with the same set of the data and same value of hyperparameters. In this article we run the model twenty times. Then, all results are averaged.

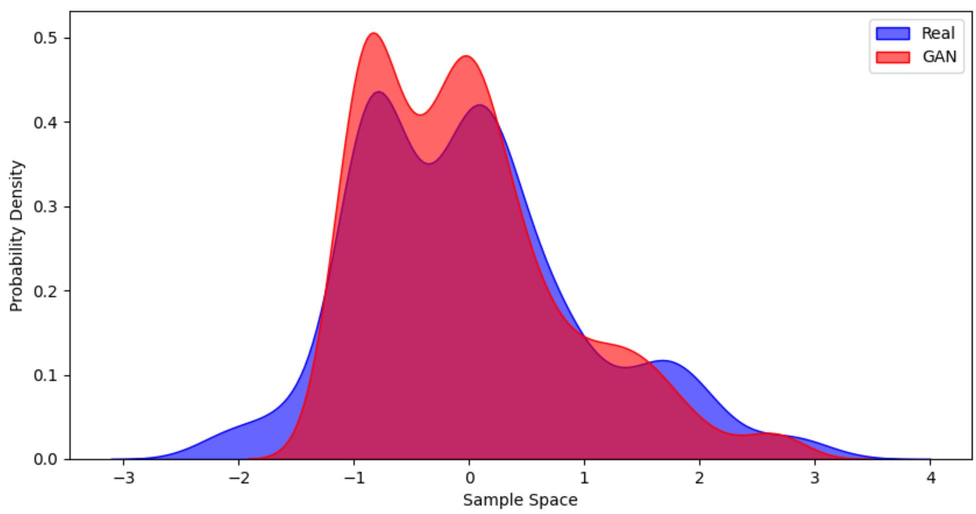
The GAN can be overfitted, when the training loss is decreasing, and validation loss is increasing. This means that the model is perfectly trained on the training data, but it is predicting validation data less precisely. Therefore, the test data will have worse results in comparison with the GAN which is trained on less epochs. Optimally we want to see training loss which is decreasing and validation loss which is stable. If the validation loss is also decreasing the model is underfitted and it is better to run more models for more epochs. As shown in Figure 3. We have optimal number of epochs for the model to be trained on.

Fig. 3. Train vs validation loss all runs



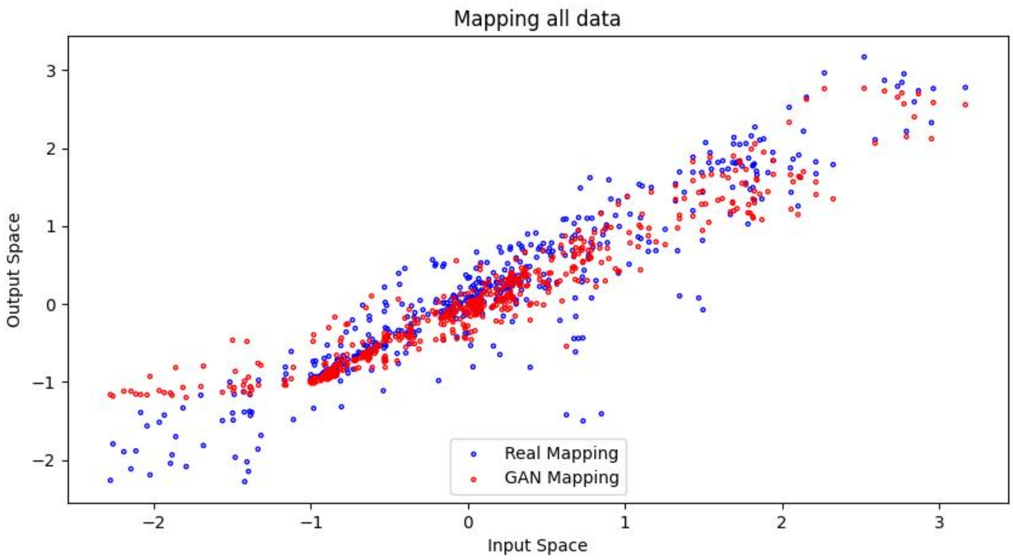
Next, we check the distribution of the predicted data and real data in the training, validation, and test period. In an optimal way the model should have the same distribution as real data. As this is an impossible task, the prediction should not look completely different from the real data and should have similar distribution. Figure 4. shows that the model is better at predicting positive periods than negative periods. In negative periods, the model is less negative and in positive periods it is more positive. This is expected to see, if we consider that financial markets in the long-term period have positive returns, the model should predict more positive results than negative ones.

Fig, 4. – probability density of the GAN and real data



The GAN mapping is the last visualization check. GAN mapping serves to check how the model predicts changes in the values in 3 months period. Again, the optimal result is that no difference between the real mapping and GAN mapping are observed. Figure 5 plots the model prediction with less volatility than is present in the real data. The only exception is the most negative results when model predict these values to be more positive. This is mostly observed in the testing and validation period when the expectation of the model after the decrease of the price increase at least at the same level in the next period. This can be due to non-occurring patterns in the training period.

Fig. 5 – Mapping of GAN prediction and real data



Overall, it can be concluded that the training process of the WGAN exhibits correct behavior as suggested by the above visualizations containing full dataset. Differences in visualization of real price values and predicted price values are negligible in all datasets. This is exactly what is expected from a correctly trained model.

5. Results

The above developed WGAN model represents an example of how the strategy for trading can be set up with comparison of the returns of the index, benchmark, cash, prediction of Random Forest and prediction of LSTM in testing period. Training and validation datasets are not used for comparison, as all used models observe the data by some level. Test dataset is invisible for them; thus, this will yield unbiased comparison.

By training the WGAN model a dataset was created and subsequently used in establishing investment strategy. The model predicts three months ahead from the previous 12 months. To compare our prediction to the benchmark, we establish benchmark as 50% invested in shares and another 50% in cash (Euribor 3M). The first model strategy is that model can have at minimum 30% of shares or maximum 70% shares. The second strategy allows the model to be fully invested in shares or cash. Each month, the position is reconsidered based on the location of real value and prediction value. To apply this trading strategy, we need to create some upper and lower bounds to established how much shares or cash we want to hold according to the recommendation of the model. Boundaries are set up as:

$$upper_{bound_{t+0......t+3}} = predicted_{price_{t+0.....t+3}} + 1{,}65 * \sigma$$

$$lower_{bound_{t+0......t+3}} = predicted_{price_{t+0.....t+3}} - 1{,}65 * \sigma$$

$$where\ \sigma = \frac{\sum_{t-12}^{t}(real_{price} - \overline{real_{price}})^2}{12}$$

Weights between index and cash are established by set of rules:

$$If\ predicted_{price_t} < predicted_{price_{t+3}}:$$

$$if\ upper_{bound} < real_{price}:$$

$$weight_{index} = 0{,}5$$

$$weight_{cash} = 0{,}5$$

$$if\ upper_{bound} > real_{price} > predicted_{price}:$$

$$weight_{index} = 0{,}75 - 0{,}25 * \frac{upper_{bound} - real_{price}}{upper_{bound} - predicted_{price}}$$

$$weight_{cash} = 1 - weight_{index}$$

$$if\ lower_{bound} < real_{price} < predicted_{price}:$$

$$weight_{index} = 0{,}75 + 0{,}25 * \frac{upper_{bound} - real_{price}}{upper_{bound} - predicted_{price}}$$

$$weight_{cash} = 1 - weight_{index}$$

$$if\ lower_{bound} > real_{price}:$$

$$weight_{index} = 0{,}7$$

$$weight_{cash} = 0{,}3$$

$$If\ predicted_{price_t} > predicted_{price_{t+3}}:$$

$$if\ upper_{bound} < real_{price}:$$

$$weight_{index} = 0{,}3$$

$$weight_{cash} = 0{,}7$$

$$if\ upper_{bound} > real_{price} > predicted_{price}:$$

$$weight_{index} = 0{,}25 - 0{,}25 * \frac{upper_{bound} - real_{price}}{upper_{bound} - predicted_{price}}$$

$$weight_{cash} = 1 - weight_{index}$$

$$if\ lower_{bound} < real_{price} < predicted_{price}:$$

$$weight_{index} = 0{,}25 + 0{,}25 * \frac{upper_{bound} - real_{price}}{upper_{bound} - predicted_{price}}$$

$$weight_{cash} = 1 - weight_{index}$$

$$if\ lower_{bound} > real_{price}:$$

$$weight_{index} = 0{,}5$$

$$weight_{cash} = 0{,}5$$

Figure 6. Comparison of returns

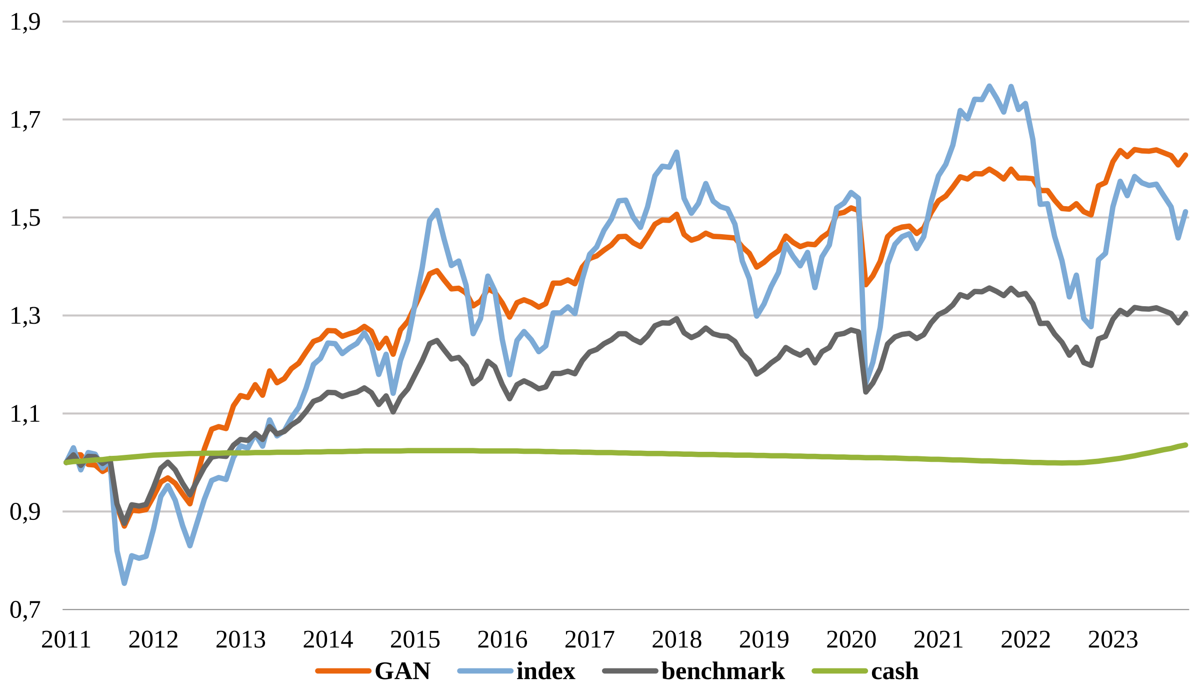
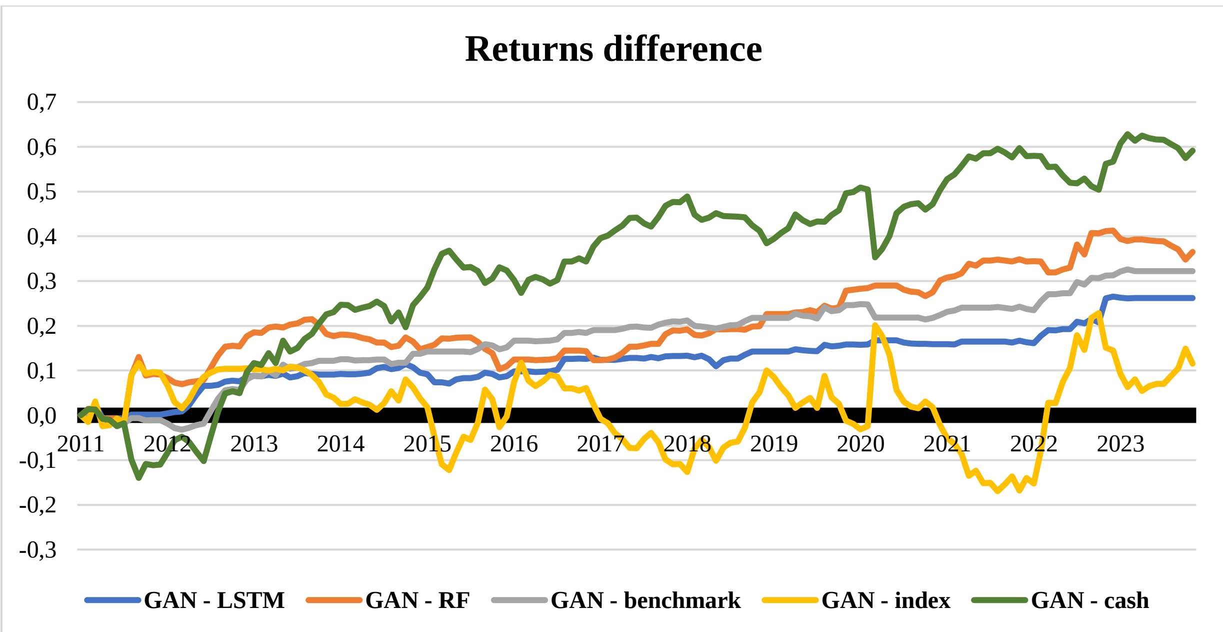


Figure 7. Comparison of all returns



When we compare our strategy with the established benchmark in testing period, our strategy based on WGAN prediction is better, as shown in Figure 6. The GAN strategy has slightly worse results only from the middle of 2011 until middle of 2012. From this period the trend value of the model is higher. The model has average returns higher by 2,69% per year in comparison with returns of the benchmark. The model outperformed the index on average mainly by the fact that it has lower volatility. When we consider that investors on financial markets are risk averse, they are willing to trade volatility for returns in the long-term period. Thus, this strategy is preferred to be fully invested in index. The WGAN is better than RF or LSTM. LSTM outperforms Random Forest also which is expected according to previous research. Another important feature of the WGAN is, that the difference between returns has positive slope and there are no sudden downfalls in performance except comparison with index. This represents, we believe, a set of stable results of the model for long-term investors but also for short-term investors.

6. Conclusion

Previous research focusses mainly on using GAN in image processing. When it comes to time series data, no explanation variables or very few are used for explanation of patterns of the data inside the model. Filling this methodological gap in current scientific and expert debate we have offered methodological change which enables us to use advanced neural network for prediction of index price with multiple variables. Based on the previous results of the separate neural networks, we use LSTM as a Generator CNN as a Discriminator. Even though it is time consuming and requires in some cases high computational power to find optimal model to use when it comes to GAN, we have obtained some robust results. The prediction outcomes confirm that WGAN is much better predictor than other comparable models. When it comes to feature selection, we have developed a methodological approach on how to deal with multiple variables and how to choose the most appropriate. This analysis eliminates any potential misjudgment in the transformation process when selecting the data. If the selected variables in the beginning of the process are selected correctly, they will have high explanatory power towards the index price. Moreover, innovative visualization techniques were presented. These can be used to check that the models behave as it is expected. This check is very important for the usage of the model, as it gives us some idea about the behavior of the model. By establishing a simple benchmark strategy to compare the results of the model we have proved superiority of WGAN as suitable investment model. Overall, the WGAN has higher, stable, and positive returns in comparison with all other options in the testing period. The only exception is when we compare it to index price, which is expected. The model is slightly better when it comes to comparison of returns, when in most of the period the returns are positive. Most importantly the model prediction is much less volatile, which is always favorable for risk averse investors.

The GAN modelling can be future of time series prediction on financial markets as this uses advantages of neural networks as much as possible. The model gets only better when we include more data, as this will help the model train on more patterns. If we have more weekly/daily data at our disposal, the GAN can prove as very strong prediction tool. This model can be enhanced if we build a structure which can identify seasonality into the account. This methodology can be used in any country if causal data are at the disposal for higher hundreds of data points. This put a limit in comparison with standard logistic regression or ARIMA, as they do not need such quantity of the data. Also, there is a need to use some feature selection process which can give us any idea about the importance of each variable.

Our results suggest that features have significant importance on slope of the prediction while model hyperparameters influence mainly the level of prediction. We believe that this important finding may improve performance of portfolio management having a direct impact on portfolio returns currently driven by RF and LSTM. Future research in GAN may be designed to reveal the importance of features selection process and their impact on prediction accuracy.

7. References

Abu-Mostafa, Y. S., & Atiya, A. F. (1996). Introduction to financial forecasting. *Applied Intelligence*, *6*(3), 205–213. https://doi.org/10.1007/BF00126626

Arévalo, R., García, J., Guijarro, F., & Peris, A. (2017). A dynamic trading rule based on filtered flag pattern recognition for stock market price forecasting. *Expert Systems with Applications*, *81*, 177–192. https://doi.org/10.1016/j.eswa.2017.03.028

Arjovsky, M., Chintala, S., & Bottou, L. (2017). *Wasserstein GAN* (arXiv:1701.07875). arXiv. http://arxiv.org/abs/1701.07875

Ballings, M., Van Den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, *42*(20), 7046–7056. https://doi.org/10.1016/j.eswa.2015.05.013

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLOS ONE*, *12*(7), e0180944. https://doi.org/10.1371/journal.pone.0180944

Celebi, K., & Hönig, M. (2019). The Impact of Macroeconomic Factors on the German Stock Market: Evidence for the Crisis, Pre- and Post-Crisis Periods. *International Journal of Financial Studies*, *7*(2), 18. https://doi.org/10.3390/ijfs7020018

Diqi, M., Hiswati, M. E., & Nur, A. S. (2022). StockGAN: Robust stock price prediction using GAN algorithm. *International Journal of Information Technology*, *14*(5), 2309–2315. https://doi.org/10.1007/s41870-022-00929-6

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). *Generative Adversarial Networks* (arXiv:1406.2661). arXiv. http://arxiv.org/abs/1406.2661

Cho, H., Kim, Y., Lee, E., Choi, D., Lee, Y., & Rhee, W. (2020). Basic Enhancement Strategies When Using Bayesian Optimization for Hyperparameter Tuning of Deep Neural Networks. *IEEE Access*, *8*, 52588–52608. https://doi.org/10.1109/ACCESS.2020.2981072

Karakostas, E. (2023). The Macroeconomic Determinants of the Stock Market Index Performance: The Case of DAX Index. *International Journal of Economics and Business Administration*, *XI*(Issue 3), 21–38. https://doi.org/10.35808/ijeba/813

Kirisci, M., & Cagcag Yolcu, O. (2022). A New CNN-Based Model for Financial Time Series: TAIEX and FTSE Stocks Forecasting. *Neural Processing Letters*, *54*(4), 3357–3374. https://doi.org/10.1007/s11063-022-10767-z

Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-Based Model to Forecast Stock Prices. *Complexity*, *2020*, 1–10. https://doi.org/10.1155/2020/6622927

Masum, M., Shahriar, H., Haddad, H., Faruk, M. J. H., Valero, M., Khan, M. A., Rahman, M. A., Adnan, M. I., Cuzzocrea, A., & Wu, F. (2021). Bayesian Hyperparameter Optimization for Deep Neural

Network-Based Network Intrusion Detection. *2021 IEEE International Conference on Big Data (Big Data)*, 5413–5419. https://doi.org/10.1109/BigData52589.2021.9671576

Mehtab, S., & Sen, J. (2022). Analysis and Forecasting of Financial Time Series Using CNN and LSTM-Based Deep Learning Models. In J. P. Sahoo, A. K. Tripathy, M. Mohanty, K.-C. Li, & A. K. Nayak (Ed.), *Advances in Distributed Computing and Machine Learning* (Roč. 302, s. 405–423). Springer Singapore. https://doi.org/10.1007/978-981-16-4807-6_39

Meng, A., Zhang, H., Yin, H., Xian, Z., Chen, S., Zhu, Z., Zhang, Z., Rong, J., Li, C., Wang, C., Wu, Z., Deng, W., Luo, J., & Wang, X. (2023). A novel multi-gradient evolutionary deep learning approach for few-shot wind power prediction using time-series GAN. *Energy*, *283*, 129139. https://doi.org/10.1016/j.energy.2023.129139

Nelson, D. M. Q., Pereira, A. C. M., & De Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. *2017 International Joint Conference on Neural Networks (IJCNN)*, 1419–1426. https://doi.org/10.1109/IJCNN.2017.7966019

Nesbitt, K. V., & Barrass, S. (2004). Finding Trading Patterns in Stock Market Data. *IEEE Computer Graphics and Applications*, *24*(5), 45–55. https://doi.org/10.1109/MCG.2004.28

Pavelka, T., & Löster, T. (2014). The use of cluster analysis for a comparison of the global financial crisis effects on the labour market of the European Union. *Forum Statisticum Slovacum*, *10*(6), 125–130.

Putatunda, S., & Rama, K. (2019). A Modified Bayesian Optimization based Hyper-Parameter Tuning Approach for Extreme Gradient Boosting. *2019 Fifteenth International Conference on Information Processing (ICINPRO)*, 1–6. https://doi.org/10.1109/ICInPro47689.2019.9092025

Rezabek, P., Marek, L., Doucek, P., & Nedomova, L. (2022). A Comparison of the Development of Selected Macroeconomic Indicators of the Regions of the V4 Countries. *Quality Innovation Prosperity*, *26*(2), 21–38. https://doi.org/10.12776/qip.v26i2.1695

Rhanoui, M., Yousfi, S., Mikram, M., & Merizak, H. (2019). Forecasting Financial Budget Time Series: ARIMA Random Walk vs LSTM Neural Network. *IAES International Journal of Artificial Intelligence (IJ-AI)*, *8*(4), 317. https://doi.org/10.11591/ijai.v8.i4.pp317-327

Roondiwala, M., Patel, H., & Varma, S. (2015). *Predicting Stock Prices Using LSTM*. *6*(4).

Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1643–1647. https://doi.org/10.1109/ICACCI.2017.8126078

Shah, D., Campbell, W., & Zulkernine, F. H. (2018). A Comparative Study of LSTM and DNN for Stock Market Forecasting. *2018 IEEE International Conference on Big Data (Big Data)*, 4148–4155. https://doi.org/10.1109/BigData.2018.8622462

Siami-Namini, S., & Namin, A. S. (2018). *Forecasting Economics and Financial Time Series: ARIMA vs. LSTM* (arXiv:1803.06386). arXiv. http://arxiv.org/abs/1803.06386

Sirisha, U. M., Belavagi, M. C., & Attigeri, G. (2022). Profit Prediction Using ARIMA, SARIMA and LSTM Models in Time Series Forecasting: A Comparison. *IEEE Access*, *10*, 124715–124727. https://doi.org/10.1109/ACCESS.2022.3224938

Sonkiya, P., Bajpai, V., & Bansal, A. (2021). *Stock price prediction using BERT and GAN* (arXiv:2107.09055). arXiv. http://arxiv.org/abs/2107.09055

Thakkar, A., & Chaudhari, K. (2021). A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. *Expert Systems with Applications*, *177*, 114800. https://doi.org/10.1016/j.eswa.2021.114800

Tovar, W. (2020). *Deep Learning Based on Generative Adversarial and Convolutional Neural Networks for Financial Time Series Predictions* (arXiv:2008.08041). arXiv. http://arxiv.org/abs/2008.08041

Velay, M., & Daniel, F. (2018). *Stock Chart Pattern recognition with Deep Learning* (arXiv:1808.00418). arXiv. http://arxiv.org/abs/1808.00418

Vuletić, M., Prenzel, F., & Cucuringu, M. (2023). Fin-GAN: Forecasting and Classifying Financial Time Series via Generative Adversarial Networks. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4328302

Widiputra, H., Mailangkay, A., & Gautama, E. (2021). Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction. *Complexity*, *2021*, 1–14. https://doi.org/10.1155/2021/9903518

Wu, J. M.-T., Li, Z., Herencsar, N., Vo, B., & Lin, J. C.-W. (2023). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. *Multimedia Systems*, *29*(3), 1751–1770. https://doi.org/10.1007/s00530-021-00758-w

Zhong, X., & Enke, D. (2017). Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications*, *67*, 126–139. https://doi.org/10.1016/j.eswa.2016.09.027

Zhou, X., Pan, Z., Hu, G., Tang, S., & Zhao, C. (2018). Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets. *Mathematical Problems in Engineering*, *2018*, 1–11. https://doi.org/10.1155/2018/4907423