



ISSN: 2543-6821 (online)

Journal homepage: <http://ceej.wne.uw.edu.pl>

Marcin Chlebus, Michał Dyczko,  
Michał Woźniak

# Nvidia's Stock Returns Prediction Using Machine Learning Techniques for Time Series Forecasting Problem

## To cite this article

Chlebus, M., Dyczko, M., Woźniak, M. (2021). Nvidia's Stock Returns Prediction Using Machine Learning Techniques for Time Series Forecasting Problem. Central European Economic Journal, 8(55), 44-62.

DOI: 10.2478/ceej-2021-0004

 To link to this article: <https://doi.org/10.2478/ceej-2021-0004>



**Marcin Chlebus** 

Faculty of Economic Sciences, Division of Quantitative Finance, University of Warsaw,  
Warszawa, Poland  
corresponding author: mchlebus@wne.uw.edu.pl

**Michał Dyczko**

Faculty of Mathematics and Computer Science, Warsaw University of Technology, Warszawa,  
Poland

**Michał Woźniak** 

Faculty of Economic Sciences, University of Warsaw, Warszawa, Poland

---

# Nvidia's Stock Returns Prediction Using Machine Learning Techniques for Time Series Forecasting Problem

---

## Abstract

Statistical learning models have profoundly changed the rules of trading on the stock exchange. Quantitative analysts try to utilise them predict potential profits and risks in a better manner. However, the available studies are mostly focused on testing the increasingly complex machine learning models on a selected sample of stocks, indexes etc. without a thorough understanding and consideration of their economic environment. Therefore, the goal of the article is to create an effective forecasting machine learning model of daily stock returns for a preselected company characterised by a wide portfolio of strategic branches influencing its valuation. We use Nvidia Corporation stock covering the period from 07/2012 to 12/2018 and apply various econometric and machine learning models, considering a diverse group of exogenous features, to analyse the research problem. The results suggest that it is possible to develop predictive machine learning models of Nvidia stock returns (based on many independent environmental variables) which outperform both simple naïve and econometric models. Our contribution to literature is twofold. First, we provide an added value to the strand of literature on the choice of model class to the stock returns prediction problem. Second, our study contributes to the thread of selecting exogenous variables and the need for their stationarity in the case of time series models.

## Keywords

machine learning | nvidia | stock returns | technical analysis | fundamental analysis

## JEL Codes

C02, C32, C40, G12

---

## 1 Introduction

In recent years, researchers have attempted to find novel and unbiased theoretical foundations that would be useful for understanding the behaviour of financial markets. The hypothesis that was considered as a breakthrough for the academic world in the above-mentioned field is 'The Adaptive Markets Hypothesis' (Lo, 2004). It means the ability to use the achievements

of fundamental analysis, technical analysis and behaviour analysis with good results. But, at the same time, the last decade was a renaissance of supervised machine learning algorithms that are used in time series prediction problems in such areas as energetics (Chou & Tran, 2018), finance (Abe & Nakayama, 2018), logistics (Laptev, Yosinski, Li & Smyl, 2017) etc. It seems that the combination of The Adaptive Markets Hypothesis and Machine Learning models for time

series can produce tangible results in stock returns forecasting.

The main purpose of this study is to forecast the daily stock returns of Nvidia Corporation company quoted on the Nasdaq Stock Market effectively, using numerous exogenous explanatory variables. The most important problems standing to be focussed by researchers are statistical specificity of rates of return i.e. as a time series they can be a white noise (Hill & Motegi, 2019), necessity of applying many atypical machine learning methods to handle the influence of the time factor (Bontempi, Taieb & Borgne, 2012) and the lack of publications constituting a benchmark in terms of regression problem. The Nvidia Corporation is selected intentionally based on economic sense because of its unique and complex cross-sectoral structure, including games, deep learning and cryptocurrency market. From a modelling perspective, it gives us the possibility to use many exogenous features that can have a significant impact on the endogenous variable – rate of return. Moreover, one can observe that Nvidia's stock price changes rapidly between October 2018 and December 2018. It is interesting to analyse whether the models are able to detect these fluctuations and predict stock returns well.

As mentioned above, the models used in this article are based on variables from various sources namely fundamental analysis of Nvidia Corporation, technical analysis of Nvidia stock prices, behaviour analysis from Google Trends data, etc. The research period spread over the years 02/07/2012–31/12/2018 and the period is divided into training set (02/07/2012–29/06/2018) and testing set (02/07/2018–31/12/2018). This is done because of the desire to test the performance of models in this particular period of time. The algorithms that will be applied to this data materialised from two types of frameworks – machine learning: SVR, KNN, XGBoost, LightGBM and LSTM Networks; and econometric methods: ARMA and ARMAX. Further, we tested ranking-based ensemble model on the above-mentioned algorithms.

The main hypothesis verified in this paper is whether it is possible to construct predictive machine learning model of Nvidia's daily stock returns, which can outperform both simple naive and econometric models. The additional research questions are (1) will models cope with the market fluctuations that began in October 2018?, (2) Do models based on stationary variables perform better than models based on stationary and non-stationary variables?, (3) Will

ranking-based ensemble models perform better than singular ones? and (4) Are the categories of variables suggested in literature significant?

The structure of this paper was composed as follows. The second part dealt with the literature review. The third part was devoted to the materials and methodology used in research. The fourth part presented the results of empirical research and answers to the hypotheses. The last part contained a summary of the work and conclusions.

## 2. Literature Review

Studies focussed to stock returns prediction in regression field using statistical learning techniques is quite skimpy, especially as far as one-day-ahead forecast is concerned. Authors who made significant contributions to this subject are Abe and Nakayama (2018). Their study examined performance of DNN, RF and SVM models in predicting one-month-ahead stock returns in the cross-section on the Japanese stock market. Moreover, they also applied weighted ensemble method. The experimental results indicate that Deep Neural Networks and weighted ensembling show great potential at stock returns forecast. Rather, Agarwal and Sastry (2015) proposed hybrid model for prediction of stock returns. The proposed model is constituted of RNN, ARMA and Holt model. The results obtained by the researchers indicated better accuracy of the hybrid model compared to single realizations of the above-mentioned approaches. Another example is provided by Pai and Lin (2005) who tried to predict stock price using ARIMA, SVR and hybrid of these models (complementary dependence: SVR was specified to predict residuals of ARIMA, and therefore minimise forecast error) on ten stocks basing on daily closing prices. They obtained promising and potential capability of the hybrid model to forecast time series. In their case, simple average of two models does not gain any significant impact. Adebisi, Adewunmi and Ayo (2014) applied ANN and ARIMA on daily stock data from NYSE and examined their performance. The empirical results presented the superiority of neural networks model over ARIMA model. It is also worth mentioning interesting works on classification problem of stock prices. Zheng and Jin (2017) analysed the results of logistic regression, Bayesian network, LSTM and SVM on daily Microsoft stock prices, along with technical indicators. The results indicated that they could achieve a correct prediction of the price trends at level of 70%. SVM

gained the best score. Chen, Zhou and Dai (2015) successfully applied LSTM to forecast changes in stock returns for China stock market in Shanghai and Shenzhen. Moreover, they showed that additional exogenous variables such as technical analysis data increase the predictive power of the model. Milosevic (2016) forecasted long-term stock price movement using RF, logistic regression and Bayesian networks. In this study, the author gathered quarterly stock prices from over 1700 stocks together with data from fundamental analysis. He showed that RF is the best one and it also provided the most appropriate set of fundamental indicators.

Based on the achievements of researchers in the field of stock returns forecasting, following econometric models will be used in this article: ARIMA developed by Whittle (1951) and popularised by Box and Jenkins (1970), ARIMAX which gives possibility to extend ARIMA model with additional exogenous variables. Nonlinear supervised machine learning models applied in this research are: SVR created by Vapnik, Drucker, Burges, Kaufman and Smola (1997) and KNN designed by Altman (1992). Models from the same category as stated above and based on boosted decision trees are XGBoost developed by Chen and Guestring (2016) and LightGBM created by Ke, Meng, Finley, Wang and Chen (2017). During the research, RNN in LSTM architecture constructed by Hochreiter and Schmidhuber (1997) was also deployed. To conduct ensembling, a Model Ranking Based Selective Ensemble Approach powered by paper of Adhikari, Verma and Khandelwal (2015) was used.

A particularly important part of the entire analysis is the collection of variables from various thematic classes i.e.: fundamental analysis, technical analysis, behavioural analysis, expert indicators of Nvidia-related markets. Undoubtedly, the best described in the literature is the selection of variables from the fundamental analysis to the problem of prediction of stock prices and stock returns. Mahmoud and Sakr (2012) showed that fundamental variables explaining issues such as profitability, solvency, liquidity and operational efficiency are helpful in creating effective investment strategies. Zeytinoglu, Akarim and Celik (2012) showed that following factors are good predictors of stock returns in their study on the Turkish insurance market: Price/Profit, Earnings per Share and Price to Book Ratio. Hatta (2012), in his research, found that Earnings Per Share has a positive effect on the rate of return and is negatively correlated with the Debt to Equity Ratio. Muhammad

and Ali (2018) divided the variables into categories so that they describe the company as a whole: liquidity (current liquidity), market value (price/profit, EPS), profitability (ROA), indebtedness (general debt). Their research for the emerging Pakistani stock exchange showed that the variables ROA (positive coefficient) and Price/Earnings (positive coefficient) were highly significant in modelling rates. The remaining variables were irrelevant or showed illogical dependencies. The use of variables from technical analysis is slightly less systematised in academic literature than variables from fundamental analysis. Stanković, Marković and Stojanović (2015) examined the effectiveness of using three technical analysis indicators: EMA (Exponential Moving Average), MACD (Moving Average Convergence-Divergence) and RSI (Relative Strength Index) to predict stock returns on selected stock exchanges in the countries of former Yugoslavia. According to their research, EMA and MACD produced the best performance, while RSI was unable to predict rates of return as expected. Hybrid models with variables from technical and fundamental analysis were also well-described in literature. Beyaz, Tekiner, Zeng and Keane (2018) conducted an experiment based on forecasting the share prices of companies from the S&P index in time windows: 126 and 252 days. Their analysis showed that hybrid models work better than single models. They also investigated the significance of variables in the hybrid approach (division into three tiers). The highest, which means the most effective tier, consists of ATR and EPS. Emir, Dincer and Timor (2012) conducted research on the specificity of the Turkish stock exchange. They created a model that ranked ten in the most profitable companies on the market. They used a hybrid approach, which turned out to be much more effective than single models. The variables that turned out to be significant in the analysis are: Price per Book Value, ROE, CCI and RSI.

In the literature, the behavioural analysis of speculators is divided into sentiment analysis and analysis of the popularity of search terms on the Internet. This paper is focussed on the second approach. Ahmed, Asif, Hina and Muzammil (2017) collected data from Google Trends to capture the relationship between online searches and political and business events. They used this knowledge to predict the ups and downs of the Pakistan Stock Exchange Index 100, quantifying the semantics of the international market. Their research shows that these variables have good prognostic properties. Preis, Moat and Stanley (2013) used Google Trends data to

predict the value of Dow Jones Industrial Average Index. They claimed that these features provide some insight into future trends in the behaviour of traders and can be good factor in the decision-making process for investors. In addition, the Nvidia Annual Financial Report (2018) legitimises the need to collect indicators linked to Nvidia related markets such as deep learning, games and crypto currency.

Yang and Shahabi (2005) show that not only model and variables selection are important, but also the feature engineering. They performed several classification experiments using correlation coefficient techniques on four datasets and investigated how data stationarity influences forecast accuracy. The results of their work indicate that one can obtain higher accuracy in prediction after differencing non-stationary data, while differencing stationary data makes the forecast less accurate. The authors suggest that the test of stationarity and possible differencing features is recommended pre-processing step.

Ultimately, we need to emphasise that Nvidia stock price dropped rapidly in 3rd quarter of 2018 by more than 30 per cent after the earning results presentation. Experts (Abazovic, 2018; Eassa, 2018) associate this fact to previous diminishment in crypto interest and mining. This has led to severe demand decrease for graphic processing units, which are Nvidia's main product line useful in cryptocurrencies mining.

### 3. Materials and Methods

#### 3.1 Dataset

Dataset preparation step was crucial in this research. After data pre-processing, dataset consists of more than 350 independent variables from various categories. The data used in this paper belongs to the period from 01/01/2012 to 31/12/2018. All information was collected in January 2019.

##### 3.1.1. Data sources

The key variables among all others used in the article are open price, close price, high price, low price and volume for stocks of companies such as Nvidia Corporation (NVIDIA), Advanced Micro Devices, Inc. (AMD), Bitcoin USD (BTC), Ubisoft Entertainment SA (UBSFY), Activision Blizzard Inc. (ATVI), Take-Two Interactive Software Inc. (TTWO)

and indexes of S&P500 and NASDAQ-100. The stocks include entities that are competitors of Nvidia and their close business partners. On the other hand, the indices show the general market situation. They were collected from Yahoo Finance.

Nvidia's fundamental analysis features origins from the balance sheet and are available on the Bloomberg Terminal. Such variables are:

- **Profitability ratios:** Return on Equity, Return on Assets, Gross Margin, Operating Margin, Return on Investment, Earnings Before Interests and Taxes Margin, Pre-Tax Profit Margin, Net Profit Margin;
- **Liquidity ratios:** Current Ratio, Operating Cash Flow per Share, Free Cash Flow per Share;
- **Debt ratios:** Long-term Debt to Capital, Debt to Equity ratio;
- **Efficiency Ratios:** Asset turnover, Inventory Turnover Ratio, Receivable Turnover, Days Sales Outstanding;
- **Market ratios:** Earning per share, Price to book value ratio, Book value per share, Price to Earnings ratio;

The behavioural analysis was based on a people's demand for information taken from the Internet, especially from the Google Search and Google News. These data are available on the Google Trends platform. It was decided to look for entries related to this study such as:

- **Google search:** Nvidia, Geforce, GTX, GPU, AMD, Intel, Deep learning, Artificial intelligence, Machine learning, Neural network, Data science, Natural language processing, Fintech, Azure, AWS, Google Cloud, Tensorflow, PyTorch, Mxnet, Blockchain, Cryptocurrency, Bitcoin, Ethereum, Bitcoin miner, Cryptocurrency miner, Gaming, E-sport, Battlefield, Just cause, Assassins Creed, Hitman, Far cry, Final fantasy, Forza motorsport, Call of duty, Witcher, Fallout, Gaming PC, Nvidia shield, GTA, Python, Twitch;
- **Google news:** Nvidia, GTX, GPU, AMD, Deep learning, Artificial intelligence, Data science, Fintech, AWS, Blockchain, Bitcoin, Ethereum, Gaming, E-sport, Battlefield, Just cause, Assassins Creed, Hitman, Far cry, Final fantasy, Forza motorsport, Call of duty, Witcher, Fallout, Gaming PC.

They were gathered from an area covering the whole world.

In order to capture how the Artificial Intelligence market is developing, it was decided to use a proxy in the form of a number of scientific publications in the field of statistics and machine learning published on the Arxiv website. These data were collected using web-scraping.

Driven by the need to use some gaming market data in this paper, the publication dates for the most demanding PC video games were scrapped from game-debate.com along with average FPS that each game scored on single set of software with GeForce GTX 1060 at 1080p resolution.

Our study also considered non-financial information published by Nvidia, which may influence the decisions of speculators. These variables are publication dates of various thematic articles on the Nvidia Newsroom website, dates of the announcement of new graphics cards and GPUs by Nvidia. Features were crawled respectively from Nvidia and Wikipedia pages.

Further, dataset was extended with such features like day of a week, day of a year, month, quarter to handle time series specificity of the research.

### 3.1.2. Feature preparation

Return ratios for: NVIDIA, BTC, UBSFY, ATVI, TTWO, BTC, S&P500, NASDAQ-100 were calculated using the formula:

$$\text{return ratio} = \frac{\text{price}_t - \text{price}_{t-1}}{\text{price}_{t-1}}. \quad (1)$$

For the same variables, a 10-period rolling variance was applied.

Technical analysis of Nvidia prices was obtained using Open Source Technical Analysis Library (TA-Lib). Depending on the technical indicator, a different set of Nvidia attributes (opening price, closing price, highest price, lowest price, volume) was used to generate a new variable. Gathered features are:

- **Overlap Studies:** Exponential Moving Average, Double Exponential Moving Average, Hilbert Transform – Instantaneous Trendline, Kaufman Adaptive Moving Average, Midpoint over period, Midpoint Price over period, Parabolic SAR,

Parabolic SAR – Extended, Triple Exponential Moving Average, Triangular Moving Average, Weighted Moving Average;

- **Momentum Indicators:** Momentum, Commodity Channel Index, Relative Strength Index, Williams' %R, Money Flow Index, Directional Movement Index, Plus Directional Movement, Percentage Price Oscillator, Aroon Oscillator, Balance Of Power, Minus Directional Movement, Ultimate Oscillator, Average Directional Movement Index, Average Directional Movement Index Rating, Absolute Price Oscillator, Chande Momentum Oscillator, Minus Directional Indicator, Plus Directional Indicator, Rate of change, Rate of change Percentage, Rate of change ratio, Rate of change ratio 100 scale;
- **Volume Indicators:** Chaikin A/D Line, On Balance Volume, Chaikin A/D Oscillator;
- **Volatility Indicators:** Average True Range, Normalised Average True Range, True Range;
- **Price Transform:** Average Price, Median Price, Typical Price, Weighted Close Price;
- **Cycle Indicators:** Hilbert Transform – Dominant Cycle Period, Hilbert Transform – Dominant Cycle Phase, Hilbert Transform – Trend vs Cycle Mode;
- **Pattern Recognition:** Two Crows, Three Black Crows, Three Inside Up/Down, Three-Line Strike, Three Outside Up/Down, Three Stars In The South, Three Advancing White Soldiers, Abandoned Baby, Advance Block, Belt-hold, Breakaway, Closing Marubozu, Concealing Baby Swallow, Counterattack, Dark Cloud Cover, Doji, Doji Star, Dragonfly Doji, Engulfing Pattern, Evening Doji Star, Evening Star, Up/Down-gap side-by-side white lines, Gravestone Doji, Hammer, Hanging Man, Harami Pattern, Harami Cross Pattern, High-Wave Candle, Hikkake Pattern, Modified Hikkake Pattern, Homing Pigeon, Identical Three Crows, In-Neck Pattern, Inverted Hammer, Kicking, Kicking - bull/bear determined by the longer marubozu, Ladder Bottom, Long Legged Doji, Long Line Candle, Marubozu, Matching Low, Mat Hold, Morning Doji Star, Morning Star, On-Neck Pattern, Piercing Pattern, Rickshaw Man, Rising/Falling Three Methods, Separating Lines, Shooting Star, Short Line Candle, Spinning Top, Stalled Pattern, Stick Sandwich, Takuri, Tasuki Gap, Thrusting Pattern, Tristar Pattern, Unique 3 River, Upside Gap Two Crows, Upside/Downside Gap Three Methods;

- **Statistic Functions:** Pearson's Correlation Coefficient, Linear Regression, Linear Regression Angle, Linear Regression Intercept, Linear Regression Slope, Standard Deviation, Time Series Forecast, Variance;
- **Math Transform Functions:** Vector Trigonometric ATan, Vector Ceil, Vector Trigonometric Cos, Vector Trigonometric Cosh, Vector Arithmetic Exp, Vector Floor, Vector Log Natural, Vector Log10, Vector Trigonometric Sin, Vector Trigonometric Sinh, Vector Square Root, Vector Trigonometric Tan, Vector Trigonometric Tanh;
- **Math Operator Functions:** CAPM Beta, Highest value over a specified period, Index of highest value over a specified period, Lowest value over a specified period, Index of lowest value over a specified period, Vector Arithmetic Mult, Vector Arithmetic Substraction, Summation.

Factors mentioned above were calculated on the assumption of default values of parameters in the software.

The Min-Max Scaler was applied to all variables from Google Trends and Google News, and newly generated data was included in the base dataset. Hence, scaled and non-scaled Google variables were taken into account.

Data from Google Trends, Google News and game release dates had non-daily frequency, so they were locally interpolated by exponential smoothing and incorporated into dataset.

The non-financial information published by Nvidia, which were mentioned in previous subsection was discretised using univariate decision trees to determine best bins. Then one-hot-encoding was applied to these features.

After exploratory data analysis, it was decided to extend dataset with some stationary variables that were obtained by differencing selected variables that seemed to react similar to Nvidia stock prices. These were, for instance: Google searches for Artificial Intelligence, Deep Learning; UBSFY index or AVGPRICE.

In order to obtain a convenient data form for supervised machine learning analysis in a time series environment, most of the explanatory factors were shifted to one or more periods. This approach allowed introducing an element of time dependence for models which, according to their architecture, are unable

to take this element into consideration. Moreover, scientists put strong emphasis on preventing data leaks. Therefore, they analysed possible problems for each group of variables, e.g. the date of publication of financial statements (fundamental variables) or data scaling (Google Trends) and prepared the input data accordingly.

### 3.2. General methodology of research

In this paper, we tested performances of models based on features from two categories namely all variables (stationary and non-stationary) and only stationary variables. Thus, each model was built twice, but methodology remains the same.

The performance of each model was interpreted as Root Mean Squared Error (RMSE) score between its predicted values and empirical values of stock returns. The choice of this metric is implied by the fact that it emphasises the importance of large individual errors. What's more, RMSE does not generate computational problems due to lack of  $y$  in the denominator that could lead to a zero-division error in the stock return problem. In addition, the metrics Mean Absolute Error (MAE) and Median Absolute Error (MedAE) were considered to better understand and analyse the final results.

Moreover, we defined a simple naive model, which is one of the benchmarks during the study. It is formulated as:

$$\hat{y}_t = y_{t-1} \quad (2)$$

This approach is appropriate and commonly used for financial problems (Shim and Siegel, 2007).

Dataset was divided into five periods: train set: 02/07/2012 –29/12/2017; first validation set: 02/01/2018–28/02/2018; second validation set: 01/03/2018–30/04/2018; third validation set: 01/05/2018–29/06/2018 and test set: 02/07/2018–31/12/2018.

Single model building process was divided into four steps: feature pre-selection, feature selection, parameters/hyperparameters tuning, generation of predictions on concatenated validation sets and test set (Figure 1).

The above algorithm was applied to the models like: ARIMA, ARIMAX, KNN, SVR, XGBoost, LightGBM and LSTM.

### 3.3. General feature selection, hyperparameters tuning and model building methodology

The feature preselection process was based on Mutual Information algorithm for a continuous target variable, which was conducted on each feature category in this research (Kozachenko and Leonenko, 1987). It builds a decent idea and knowledge for further selection of variables.

In fact, feature selection methodology depends on machine learning model (details in 3.4 subsection), but in general, high-level approach was presented here: hyperparameters were chosen randomly from the range proposed by experts and model authors; a dynamic forecast was used due to the specificity of time series modelling; models during feature selection were trained on train set and validated on the concatenated validation sets. After this operation, final lists of the best variables were obtained for each model.

The choice of model hyperparameters is crucial. Fine-tuning them, due to the specificity of time series prediction, requires a lot of attention to deal with bias-variance trade-off satisfactory. In this research, it is important to obtain hyperparameters that ensure the variability of forecasted values (to prevent underfitting). Our hyperparameters are selected in 3 steps as described in Algorithm 1.

---

#### Algorithm 1. Hyperparameters tuning algorithm.

---

1. **For each** pair of sets  $(X_i, Y_i) \in S = \{(train, validation_1), (train \cup validation_1, validation_2), (train \cup validation_1 \cup validation_2, validation_3)\}$  next operations will be performed:
  - a. the possibly largest group of hyperparameters will be selected according to best practice mentioned in literature,
  - b. one-step-ahead prediction will be done, providing  $X_i$  as train set and  $Y_i$  as test set, and then one model with the lowest RMSE will be chosen, with parameters  $H_i$ .

As a result, set  $\{H_1, H_2, H_3\}$  is obtained.

2. **For**  $H_i$  will be executed three predictions on each pair from  $S$ . In effect 3 RMSE will be received, from which the average will be calculated –  $A_i$ . As a result, set  $\{A_1, A_2, A_3\}$  is obtained.
  3.  $H_j$  will be chosen, where  $A_j = \min\{A_1, A_2, A_3\}$ . It is the best set of hyperparameters, which is believed to assure stable fit in future forecasts.
- 

At this point, when the best set of variables and the best set of hyperparameters are collected, two predictions will be made in one-step-ahead approach

on concatenated validation set and test set. The forecasts on the validation chunk were used to prepare ensembling.

### 3.4. Ensembling procedure

The ensemble algorithm that was performed in this paper is an implementation of a Model Ranking Based Selective Ensemble Approach (Adhikari et al., 2015). Its methodology is based on weighted average. The algorithm is specially created for the time series problem.

Let  $MSE_i$  be mean squared error of forecast of  $i$ -th model on validation set, so  $i$ -th weight is expressed by:

$$\omega_i = \frac{1 / MSE_i}{\sum_{j=1}^n 1 / MSE_j} \quad (3)$$

Then formula for ensembling model will be:

$$\hat{y} = \sum_{i=1}^n \omega_i \cdot M_i(X_i) \quad (4)$$

where  $M_i(X_i)$  is forecast on test set provided by  $i$ -th individual model, given matrix of regressors as  $X_i$ .

The selection of the parameter  $n$  is arbitrary. However, the methodologists recommend careful choice of  $n$  e.g. in the iteration process. In this research, various values of  $n$  parameter up to the predefined threshold were tested.

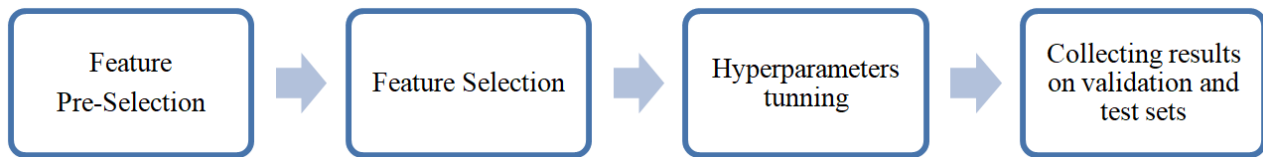
Let's assume that  $S$  is a set of models based on stationary variables,  $A$  is a set of models based on stationary and non-stationary variables and  $M = S \cup A$ . There were three types of ensembling models i.e. established on models from  $S$ ,  $A$  or  $M$ .

## 4. Empirical Results

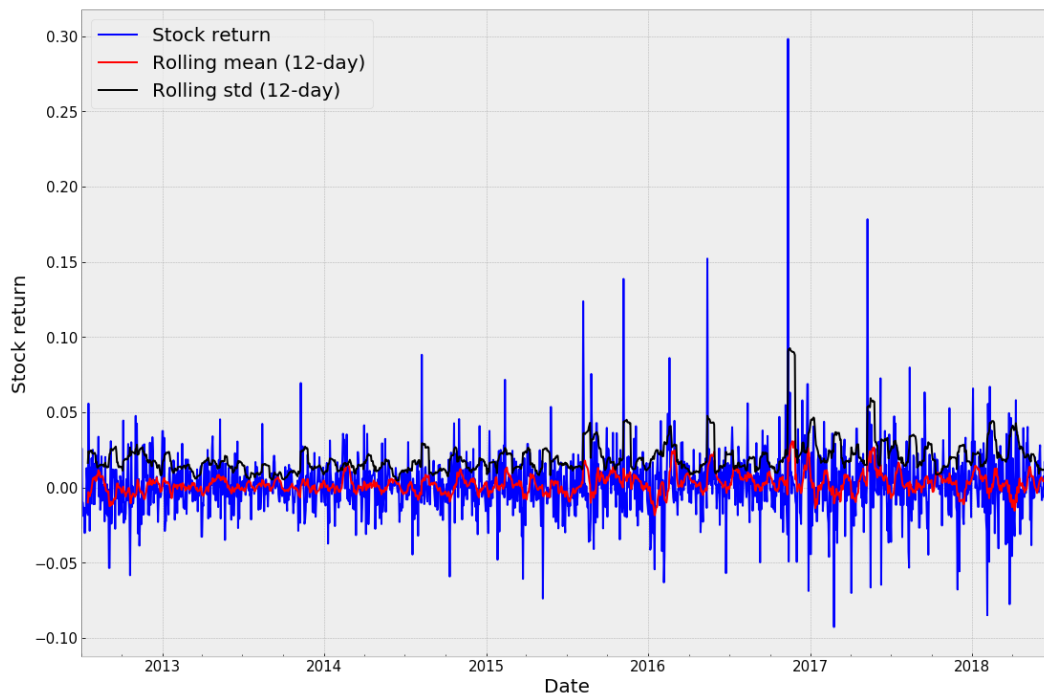
### 4.1. Exploratory data analysis of target variable

Studying the properties of the dependent variable in the time series analysis is very important because it should pass strict mathematical assumptions before beginning of further analysis. Stock returns variable by its very nature is often problematic, because it might occur to be a white noise (Hill and Motegi, 2019). In econometric theory, it is not suitable for univariate forecast. Moreover, the statistical approach





**Fig. 1.** Algorithm of model building  
Source: Author



**Fig. 2.** Nvidia stock returns on in-sample set  
Source: Authors calculations

requires stationarity of endogenous feature (Davies and Newbold, 1979). These issues are examined in this subsection.

Initially, the stationarity of the target variable on in-sample and out-of-sample sets was inspected using Figures 2 and 3. Both plots, especially statistics of rolling mean and variance, suggest that time series is stationary.

The Augmented Dickey-Fuller test was performed to check stationarity of the time series on in-sample and out-of-sample data. The algorithm implemented in the Statsmodels package automatically selects number of lags to assure that residuals are not autocorrelated. Table 1 confirms the impression that stock returns are stationary in both periods.

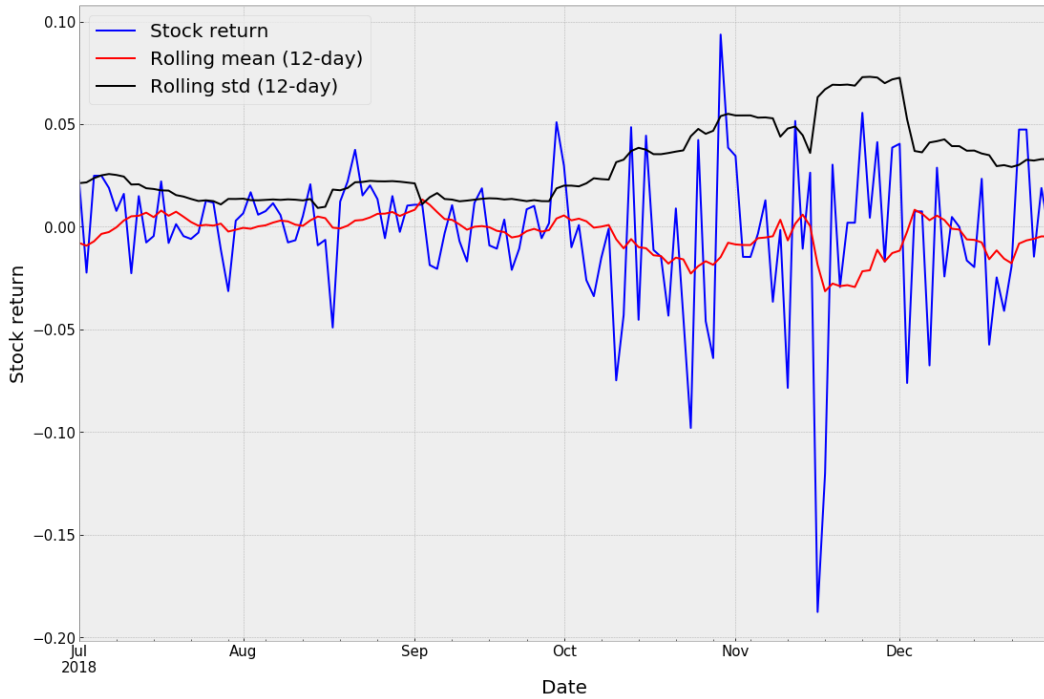
The Ljung-Box test was performed to verify whether time series on in-sample and out-of-sample chunk is a white noise. Based on the Figure 4 for the

in-sample set, the hypothesis of white noise is rejected. On the other hand, for out-of-the sample period, it is not possible to reject the white noise hypothesis.

The autocorrelation and partial correlation plots, presented in Figure 5, give a deeper insight into lag significance. For in-sample set, the smallest lags that are correlated are the third and the eighth. It justifies the need to consider these lags in model specification. According to out-of-sample chunk, all lags up to twentieth are not correlated, and it is another result confirming that the time series during the testing period is a white noise.

#### 4.2. Singular models

In Table 2, one can see results of all singular models: SVR, KNN, XGBoost, LightGBM and LSTM,



**Fig. 3.** Nvidia stock returns on out-of-sample set  
Source: Authors calculations

**Tab. 1.** Results of Augmented Dickey-Fuller test for in-sample set and out-of-sample set

| Test statistic (in sample) | p-value (in sample) | Test statistic (out of sample) | p-value (out of sample) |
|----------------------------|---------------------|--------------------------------|-------------------------|
| -11.01                     | <0.0001             | -11.09                         | <0.0001                 |

Source: Authors calculations.

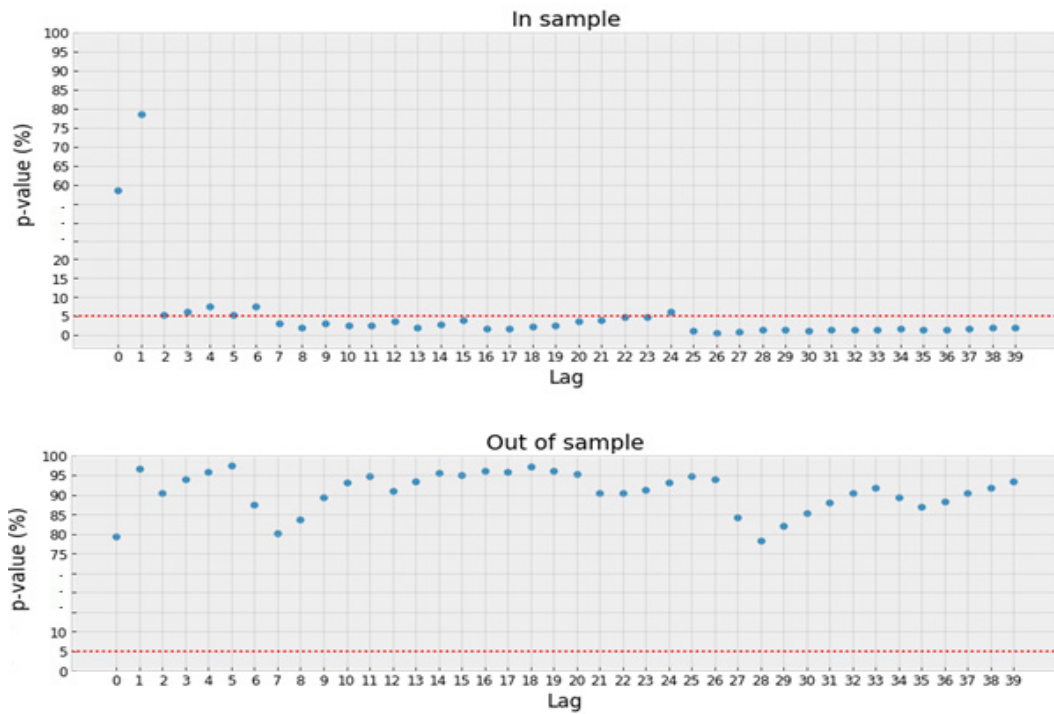
based on stationary variables. It represents values of 3 metrics of estimation quality: RMSE; MAE; MedAE. And each model gained on validation and test set, number of attributes used and values of hyperparameters. Analogically, Table 3 consists of results of these models based on stationary and non-stationary variables. In addition, we must report that the results for the ARIMA and ARIMAX models are not published. During the test period, both models converge to the average value of stock returns. In the case of ARIMA, the forecast was equal to zero, while the obtained variance was negligible for ARIMAX, and consequently to insignificant from zero. This state makes it impossible to compare them with other approaches. We can unequivocally ascertain that they are not suitable for forecasting Nvidia stock returns.

For all of the models in Table 2, RMSE on the test set was about 0.01 higher than on the validation set. The lowest value of Root Mean Squared Errors

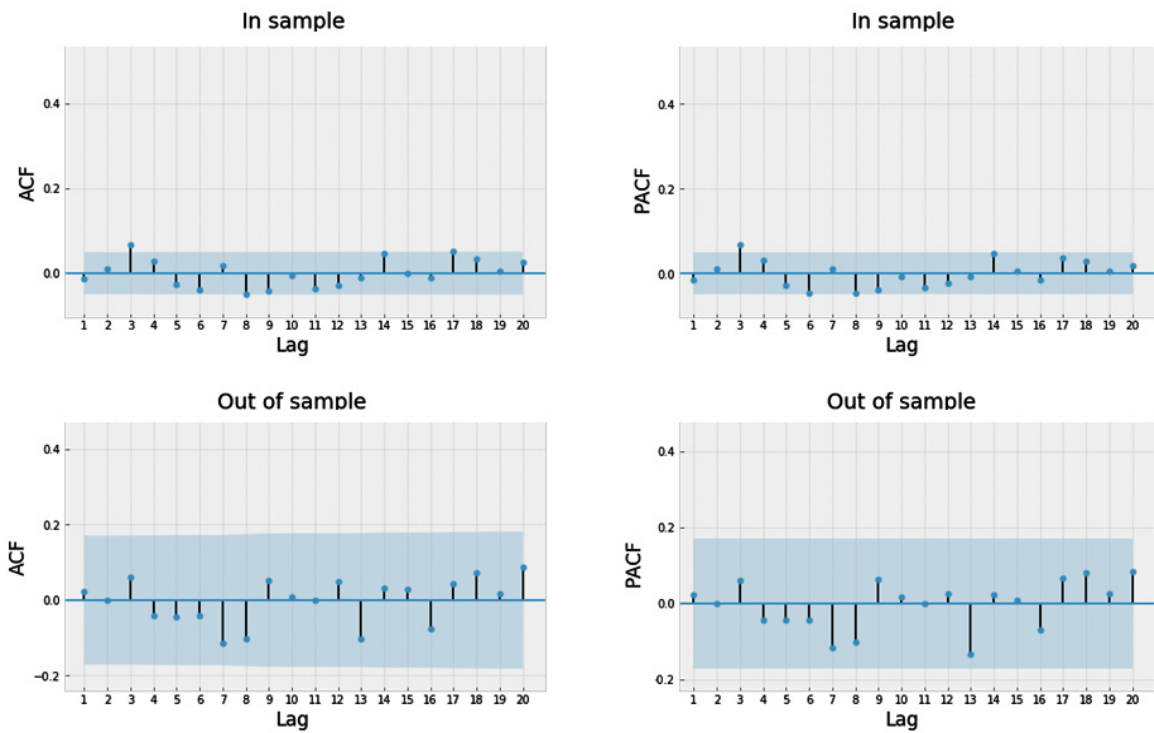
metric growth was noted on SVR model (~0.09) based on 20 stationary variables with C=0.005206 and epsilon=0.087308. Scores of RMSE on the test set when compared with the validation set, they are much worse, which can be caused by difficulty of that period (white noise). It implies the lack of model’s stability.

All the models in Table 3 have RMSE on the test set over 0.011 higher than on the validation set. The lowest value of Root Mean Squared Errors metric growth was noted on LSTM model (~0.011) based on 20 stationary and non-stationary variables. Values of RMSE on the test set on comparing with the validation set are significantly worse, which can be caused by white noise nature. It implies the lack of model’s ability to forecast stock returns.

For major of the models a stationarity of variables didn’t have impact of prediction quality, except KNN



**Fig. 4.** Results of Ljung-Box test for in-sample set and out-of-sample set. Notes: Figure presents results of Ljung-Box test of white-noise hypothesis of Nvidia’s stock returns on in-sample set and out-of-sample set  
Source: Authors calculations



**Fig. 5.** ACF and PACF for in-sample and out-of-sample sets. Notes: Figure presents autocorrelation and partial autocorrelation plots of Nvidia’s stock returns on in-sample and out-of-sample sets  
Source: Authors calculations

**Tab. 2.** Results of singular models on validation and test set (based on stationary variables)

| Model (number of attributes) | Set        | Hyperparameters   | RMSE     | MAE      | MedAE    |
|------------------------------|------------|---|----------|----------|----------|
| SVR (20)                     | Validation | C=0.005206<br>Epsilon=0.087308  | 0.026924 | 0.019478 | 0.014985 |
| SVR (20)                     | Test       | C=0.005206<br>epsilon=0.087308  | 0.036014 | 0.024916 | 0.016682 |
| KNN (20)                     | Validation | Power of Minkowski metric=2<br>k=7<br>Weight function=uniform   | 0.026328 | 0.020331 | 0.016199 |
| KNN (20)                     | Test       | Power of Minkowski metric=2<br>k=7<br>Weight function=uniform   | 0.039305 | 0.025935 | 0.017202 |
| XGBoost (27)                 | Validation | Max depth:7<br>Subsample: 0.760762<br>Colsample by tree: 0.199892<br>Lambda: 0.345263<br>Gamma: 0.000233<br>Learning rate: 0.2            | 0.027622 | 0.020678 | 0.016553 |
| XGBoost (27)                 | Test       | Max depth:7<br>Subsample: 0.760762<br>Colsample by tree: 0.199892<br>Lambda: 0.345263<br>Gamma: 0.000233<br>Learning rate: 0.2            | 0.038848 | 0.027218 | 0.019782 |
| LGBM (43)                    | Validation | Number of leaves:58<br>Min data in leaf:21<br>ETA: 0.067318<br>Max drop: 52<br>L1 regularization: 0.059938<br>L2 regularization: 0.050305 | 0.025905 | 0.018803 | 0.014339 |
| LGBM (43)                    | Test       | Number of leaves:58<br>Min data in leaf:21<br>ETA: 0.067318<br>Max drop: 52<br>L1 regularization: 0.059938<br>L2 regularization: 0.050305 | 0.038870 | 0.026283 | 0.016467 |
| LSTM (20)                    | Validation | $H_1$   | 0.026565 | 0.019741 | 0.014537 |
| LSTM (20)                    | Test       | $H_1$   | 0.036705 | 0.024918 | 0.016772 |

Notes: Table represents values of 3 metrics of estimation quality: RMSE; MAE; MedAE that SVR, KNN, XGBoost, LightGBM, LSTM (based on stationary variables) gained on validation and test set, number of attributes and values of hyperparameters.

$H_1$ : number of hidden layers: 1 LSTM layer with dense layer at the end; number of units on each layer: first layer with 20; number of epochs: 100; activation functions: sigmoid on first layer and linear on dense layer; optimiser function: Adam; batch size: 32; loss function: MSE.

Source: Author calculations.

**Tab. 3.** Results of singular models on validation and test set (based on stationary and non-stationary variables)

| Model (number of attributes) | Set        | Hyperparameters   | RMSE     | MAE      | MedAE    |
|------------------------------|------------|---|----------|----------|----------|
| SVR (27)                     | Validation | C=0.005317, epsilon=0.092179  | 0.025632 | 0.019126 | 0.015488 |
| SVR (27)                     | Test       | C=0.005317, epsilon=0.092179  | 0.041904 | 0.025875 | 0.017279 |
| KNN (40)                     | Validation | Power of Minkowski metric=1<br>k=6<br>Weight function=uniform   | 0.027021 | 0.020110 | 0.013813 |
| KNN (40)                     | Test       | Power of Minkowski metric=1<br>k=6<br>Weight function=uniform   | 0.039313 | 0.026863 | 0.018946 |
| XGBoost (74)                 | Validation | Max depth:3<br>Subsample: 0.840403<br>Colsample by tree: 0.605006<br>Lambda: 4.461698<br>Gamma: 0.000808<br>Learning rate: 0.105          | 0.028021 | 0.021604 | 0.020396 |
| XGBoost (74)                 | Test       | Max depth:3<br>Subsample: 0.840403<br>Colsample by tree: 0.605006<br>Lambda: 4.461698<br>Gamma: 0.000808<br>Learning rate: 0.105          | 0.040685 | 0.026906 | 0.016939 |
| LGBM (80)                    | Validation | Number of leaves:32<br>Min data in leaf:38<br>ETA: 0.099519<br>Max drop: 51<br>L1 regularization: 0.060221<br>L2 regularization: 0.050423 | 0.025840 | 0.019361 | 0.014083 |
| LGBM (80)                    | Test       | Number of leaves:32<br>Min data in leaf:38<br>ETA: 0.099519<br>Max drop: 51<br>L1 regularization: 0.060221<br>L2 regularization: 0.050423 | 0.037284 | 0.026295 | 0.017959 |
| LSTM (20)                    | Validation | H <sub>2</sub>  | 0.028334 | 0.021702 | 0.018201 |
| LSTM (20)                    | Test       | H <sub>2</sub>  | 0.039593 | 0.028891 | 0.020576 |

Notes: Table represents values of 3 metrics of estimation quality: RMSE; MAE; MedAE that SVR, KNN, XGBoost, LightGBM, LSTM (based on stationary and non-stationary variables) gained on validation and test set, number of attributes and values of hyperparameters.

H<sub>2</sub>: number of hidden layers: 2 LSTM layers with batch normalisation after both of them and dense layer at the end; number of units on each layer: first layer with 20 units and second with 32 units; number of epochs: 600; activation functions: sigmoid on first layer and linear on dense layer; optimiser function: SGD; regularization: bias regularizer at first layer and activity regularizer (L2) on second LSTM layer; dropout: 0.3 after first LSTM layer; batch size: 16; loss function: MSE.

Source: Authors calculations.

**Tab. 4.** Performance of ensemble models on test set (models based on stationary variables)

| Number of models | Models (weight)   | RMSE            | MAE             | MedAE           |
|------------------|---|-----------------|-----------------|-----------------|
| 2                | LightGBM (0.508099), KNN (0.491901)   | 0.038571        | 0.025784        | 0.017147        |
| 3                | LightGBM (0.342575), KNN (0.331655), LSTM (0.32577)   | 0.037403        | 0.025111        | 0.015704        |
| 4                | LightGBM (0.260092), KNN (0.251801), LSTM (0.247333), SVR (0.240775)                          | 0.036181        | 0.024366        | 0.01599         |
| <b>5</b>         | <b>LightGBM (0.211671), KNN (0.204923), LSTM (0.201287), SVR (0.19595), XGBoost (0.18617)</b> | <b>0.036106</b> | <b>0.024094</b> | <b>0.015307</b> |

Note: Table represents performance of ensemble models on test set built of models based on stationary variables. Bold rows respond to the model with the lowest RMSE in that category of ensembling.

Source: Authors calculations.

and LSTM, where usage of non-stationary variables in model estimation resulted in decreased performance.

#### 4.3. Ensemble models

Due to variety of machine learning models in this paper, it is very interesting to study the ensembling approach. Ensemble models were expected to outperform singular models. So, performances of ensembling models from three categories mentioned in methodology chapter were analysed to confirm the expectations. Table 4 represents results of ensembling of models, which are based on stationary variables. As one can see, combining five best singular models LightGBM, KNN, LSTM, SVR, and XGboost provides the best score of 0.036106 for RMSE in that category.

Results of ensembling models, which are based on stationary and non-stationary features, are collected in the Table 5. Integration of SVR, LightGBM, KNN, XGBoost and LSTM provides the best results (RMSE: 0.038053). Models from Table 4 are better than those from Table 5 in the mean sense. It is caused by huge impact of SVR performance (based on stationary and non-stationary variables) on validation set in contrast to test set, where its prognostic capabilities are weak.

Table 6 aggregates performances of ensemble models based on all appropriate, available models in research repository. The best model connects: SVR (stationary and non-stationary variables), LightGBM (stationary and non-stationary variables), LightGBM (stationary variables), KNN (stationary variables), LSTM (stationary variables), SVR (stationary variables), KNN (stationary and non-stationary

variables and XGboost (stationary variables). It gains score on RMSE: 0.036746. This result is worse than performance achieved by ensembling established only on stationary models.

To sum up, weights of singular models were calculated based on its RMSE on validation set. It was believed that it will improve forecasts of ensembling models. However, surprisingly it is just opposite. Singular models perform poorly on test set and indirectly influence ensembling model's accuracy (in RMSE meaning). Moreover, it has been observed that the results converge to the average value (weights are split almost equally) when there is an increase in number of models in ensembling algorithm and thus the variance decreases. Though the models are getting better (RMSE), but in the context of research, analysing them loses sense and does not serve any purpose.

#### 4.4. Tabular summary

To summarise the obtained models and compare its performances in a systematised way, Tables 7 and 8 were prepared. Additionally, we used them to examine two hypotheses. Best ensemble model, which is based on stationary variables, is composed of: LightGBM, KNN, LSTM, SVR and XGBoost. The same models are a part of best ensemble model based on stationary and non-stationary features. Moreover, naive model results were attached into tables. As one can observe in Table 7, according to Root Mean Squared Error metric, SVR performed the best on test set among models based on stationary variables. LSTM score

**Table 5.** Performance of ensemble models on test set (models based on stationary and non-stationary variables)

| Number of models | Models (weight)   | RMSE            | MAE             | MedAE           |
|------------------|---|-----------------|-----------------|-----------------|
| 2                | SVR (0.504057), LightGBM (0.495943)   | 0.038730        | 0.025882        | 0.017714        |
| 3                | SVR (0.346773), LightGBM (0.341191), KNN (0.312036)   | 0.038314        | 0.026003        | 0.016682        |
| 4                | SVR (0.268786), LightGBM (0.264459), KNN (0.241861), XGBoost (0.224895)                         | 0.038301        | 0.025793        | 0.016876        |
| <b>5</b>         | <b>SVR (0.220323), LightGBM (0.216777), KNN (0.198253), XGBoost (0.184346), LSTM (0.180301)</b> | <b>0.038053</b> | <b>0.026068</b> | <b>0.016645</b> |

Notes: Table represents performance of ensemble models on test set built of models based on stationary and non-stationary variables. Bold rows respond to the model with the lowest RMSE in that category of ensembling.

Source: Author calculations.

**Tab. 6.** Performance of ensemble models on test set (based on all models)

| Number of models | Models (weight)   | RMSE            | MAE             | MedAE          |
|------------------|---|-----------------|-----------------|----------------|
| 2                | S+NS SVR (0.504057), S+NS LightGBM (0.495943)   | 0.03873         | 0.025882        | 0.017714       |
| 3                | S+NS SVR (0.337508), S+NS LightGBM (0.332075), S LightGBM (0.330417)  | 0.038593        | 0.025959        | 0.017321       |
| 4                | S+NS SVR (0.255711), S+NS LightGBM (0.251594), S LightGBM (0.250338), S KNN (0.242357)  | 0.038436        | 0.025665        | 0.017152       |
| 5                | S+NS SVR (0.206542), S+NS LightGBM (0.203217), S LightGBM (0.202202), S KNN (0.195756), S LSTM (0.192283)   | 0.037734        | 0.025267        | 0.016599       |
| 6                | S+NS SVR (0.173976), S+NS LightGBM (0.171176), S LightGBM (0.170321), S KNN (0.164891), S LSTM (0.161965), S SVR (0.157671)   | 0.03681         | 0.024751        | 0.01743        |
| 7                | S+NS SVR (0.150427), S+NS LightGBM (0.148006), S LightGBM (0.147266), S KNN (0.142572), S LSTM (0.140042), S SVR (0.136329), S+NS KNN (0.135359)  | 0.036871        | 0.024897        | 0.016953       |
| <b>8</b>         | <b>S+NS SVR (0.133177), S+NS LightGBM (0.131034), S LightGBM (0.130379), S KNN (0.126223), S LSTM (0.123983), S SVR (0.120696), S+NS KNN (0.119837), S XGBoost (0.114672)</b>   | <b>0.036746</b> | <b>0.024681</b> | <b>0.01647</b> |
| 9                | S+NS SVR (0.119825), S+NS LightGBM (0.117896), S LightGBM (0.117307), S KNN (0.113568), S LSTM (0.111553), S SVR (0.108595), S+NS KNN (0.107822), S XGBoost (0.103175), S+NS XGBoost (0.100259)                       | 0.036898        | 0.024757        | 0.01645        |
| 10               | S+NS SVR (0.109125), S+NS LightGBM (0.107368), S LightGBM (0.106832), S KNN (0.103426), S LSTM (0.101591), S SVR (0.098897), S+NS KNN (0.098193), S XGBoost (0.093961), S+NS XGBoost (0.091305), S+NS LSTM (0.089302) | 0.036899        | 0.024915        | 0.016466       |

Note: Table represents performance of ensemble models on test set built of all models. Bold rows respond to the model with the lowest RMSE in that category of ensembling.

Source: Author calculations.

**Table 7.** Performance of models on test set (based on stationary variables)

| Metric | SVR             | KNN      | XGBoost  | LSTM     | LGBM     | Best ensemble   | Naive model     |
|--------|-----------------|----------|----------|----------|----------|-----------------|-----------------|
| RMSE   | <b>0.036014</b> | 0.039305 | 0.038848 | 0.036705 | 0.038870 | <b>0.036106</b> | <b>0.050244</b> |
| MAE    | <b>0.024916</b> | 0.025935 | 0.027218 | 0.024918 | 0.026283 | <b>0.024094</b> | <b>0.034908</b> |
| MedAE  | <b>0.016682</b> | 0.017202 | 0.019780 | 0.016772 | 0.016467 | <b>0.015307</b> | <b>0.022378</b> |

Note: Table represents performance of all models on test set based on stationary variables.

Source: Authors calculations.

**Tab. 8.** Performance of models on test set (based on stationary and non-stationary variables)

| Metric | SVR             | KNN             | XGBoost         | LSTM            | LGBM            | Best ensemble model | Naive model     |
|--------|-----------------|-----------------|-----------------|-----------------|-----------------|---------------------|-----------------|
| RMSE   | <b>0.041904</b> | <b>0.039313</b> | <b>0.040685</b> | <b>0.039593</b> | <b>0.037284</b> | <b>0.038053</b>     | <b>0.050244</b> |
| MAE    | <b>0.025875</b> | <b>0.026863</b> | <b>0.026906</b> | <b>0.028891</b> | <b>0.026295</b> | <b>0.026068</b>     | <b>0.034908</b> |
| MedAE  | <b>0.017279</b> | <b>0.018946</b> | <b>0.016939</b> | <b>0.020576</b> | <b>0.017959</b> | <b>0.016645</b>     | <b>0.022378</b> |

Note: Table represents performance of all models on test set based on stationary and non-stationary variables.

Source: Authors calculations.

is also satisfactory for the aim of the research. In this case, ensemble model cannot overpass singular models, particularly SVR. All the established models (from Table 7) were able to surpass naïve model.

As presented in Table 8, LightGBM gained the best results on the test set. Other models performed noticeably worse on RMSE metric. Interestingly SVR, which has the highest RMSE, gained the lowest Mean Absolute Error. It is due to the fact that RMSE is sensitive to outliers. Ensemble approach fails to introduce an improvement in prediction. As before, naïve model has the worst outcome on test set.

For further deeper insight into models efficiency, best ensemble models from 3 categories were compared in Table 9 with singular models that perform the best against stationary variables and all variables. Naive model was also included as a benchmark. Among models based on stationary features, SVR occurred to surpass other models. RMSE obtained by Support Vector Regression was 0.036014. In the group of models built from stationary and non-stationary variables, LightGBM was the best one with score of RMSE at level equal to 0.037284. Therefore, models based on stationary features give higher precision of forecast than models estimated on stationary and non-stationary variables. All in all, SVR model (based

on stationary features) outperformed other models based on Table 9. Figure 6 presents its results on the test set. This plot suggests that, during the first period of testing (before a bear market), model has a low and almost constant variance, however during second period fitted values deviates significantly, but model can detect arising spikes. Comparing three types of ensemble models, the one based on the stationary approach returns the best forecasts. The third ensemble model, which is based on all available singular models, does not gain additional impact on this study.

## 5. Conclusions

Note that that all hypotheses mentioned have been analysed and answered in this research. The main purpose of this study was to effectively forecast the daily stock returns of Nvidia Corporation company quoted on the Nasdaq Stock Market, using numerous exogenous explanatory variables. Considering these variables, the major hypothesis that has been verified in this paper is that it is possible to construct prediction model of Nvidia daily return ratios, which can outperform both simple naïve and econometric models. According to results, outperforming simple

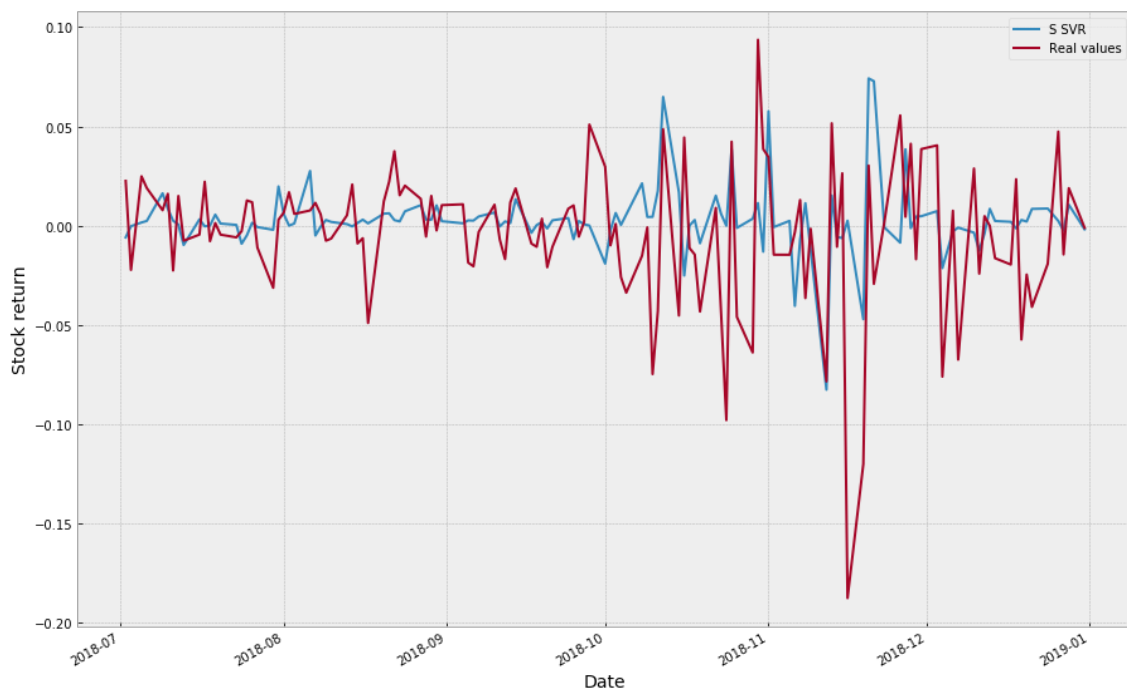


**Tab. 9.** Performance of best models on test set (ensemble and primary models)

| Metric | Best stationary ensemble model | Best stationary + non-stationary ensemble model | Best ensemble model based on all models | Best stationary model - SVR | Best stationary + non-stationary model - LGBM | Naive model |
|--------|--------------------------------|---|---|-----------------------------|---|-------------|
| RMSE   | 0.036106                       | 0.038053  | 0.036746                                | <b>0.036014</b>             | 0.037284                                      | 0.050244    |
| MAE    | 0.024094                       | 0.026068  | 0.024681                                | <b>0.024916</b>             | 0.026295                                      | 0.034908    |
| MedAE  | 0.015307                       | 0.016645  | 0.01647                                 | <b>0.016682</b>             | 0.017959                                      | 0.022378    |

Note: Table represents compof performances of best singular and ensemble models from each category and naive model on test set.

Source: Authors calculations.

**Fig. 6.** Performance on test set of the best model in research – SVR (based on stationary variables)

Source: Authors calculations

naive model is not a challenging task because every final model had much lower RMSE of forecast residuals. It is found that econometric models such as ARIMA and ARIMAX are not able to rationally forecast during the white noise period (they converge to 0). Thus machine learning models are more appropriate than traditional statistical-econometric methods in case of stock returns prediction. The best score was obtained by SVR based on stationary variables, whereas Abe and Nakayama (2018) show that DNN has much better prognostic properties than SVR. The reason for that might be forecasting window (one-day-ahead

*vs* one-month-ahead approach). Prepared models were not able to cope with market fluctuations that began in October 2018. It comes from specificity of testing set, where variance increased drastically on an unprecedented scale. Models based on stationary variables perform better than models based on stationary and non-stationary variables (RMSE: SVR with 0.036014 *vs* LightGBM with 0.037284). It is consistent with results obtained by Yang and Shahabi (2005) in classification problem, in which correlation among features is essential. Adebisi, Adewumi and Ayo (2014) have shown the superiority of neural

networks model over ARIMA model in prediction of profits from shares. Surprisingly ranking-based ensemble models did not perform better than singular ones, which is contrary to conclusions drawn by Adhikari et al. (2015). Categories of variables which are suggested in literature are significant in final models. The combination of technical analysis and fundamental analysis, as proposed by Beyaz et al. (2018), turned out to be a good idea. Features selection algorithms extracted many features based on Google Trends entries and this fact is consistent with the discovery of Ahmed et al. (2017). It is worth noticing that singular variables recommended by literature e.g. Mahmoud and Sakr (2012) were always rejected in feature extraction.

We state that our contribution to literature is twofold. First, we provide added value to the strand of literature on the choice of model class to the stock returns prediction problem. Second, our study contributes to the thread of selecting exogenous variables and the need for their stationarity in the case of time series models.

There exist many noticeable challenges on this field that should be investigated in the future. Simple naive model performed poorly on test set. Hence, other models should be considered as a benchmark. Furthermore, this model should not have a variance converging to zero, which is an important problem. Due to the specifics of the stock exchange, models degrade very quickly. Perhaps a reasonable approach would be to build and calibrate models (based on new variables) in a quarterly period. Additionally, Nested Cross Validation algorithm might be applied. As ARIMAX failed, GARCH model could be examined alternately. Another improvement could be obtained using different algorithms of ensembling (blending and stacking). As the part of study is connected with variables, sentimental analysis should be taken into account.

## References

Abazovic, F. (2018). Nvidia stock dropped for the wrong reason. Retrieved from <https://www.fudzilla.com/news/ai/47637-nvidia-stock-dropped-for-the-wrong-reason>

Abe, M., & Nakayama, H. (2018). Deep learning for forecasting stock returns in the cross-section. Pacific-Asia Conference on Knowledge Discovery and Data Mining. 273–284. [https://doi.org/10.1007/978-3-319-93034-3\\_22](https://doi.org/10.1007/978-3-319-93034-3_22)

Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of ARIMA and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics* 2014. <https://doi.org/10.1155/2014/614342>

Adhikari, R., Verma, G., & Khandelwal, I. (2015). A model ranking based selective ensemble approach for time series forecasting. *Procedia Computer Science*, 48, 14–21. <https://doi.org/10.1016/j.procs.2015.04.104>

Ahmed, F., Asif, R., Hina, S., & Muzammil, M. (2017). Financial market prediction using Google Trends. *International Journal of Advanced Computer Science and Applications*, 8(7), 388–391. <https://doi.org/10.14569/IJACSA.2017.080752>

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), 175–185. <https://doi.org/10.2307/2685209>

Beyaz, E., Tekiner, F., Zeng, X. J., & Keane, J. (2018). Stock Price Forecasting Incorporating Market State. 2018 IEEE 20th International Conference on High Performance Computing and Communications. <https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00263>

Bontempi, G., Taieb, S., & Borgne, Y. A. (2012). Machine learning strategies for time series forecasting. In *European business intelligence summer school* (pp. 62–77). Berlin, Germany: Springer. [https://doi.org/10.1007/978-3-642-36318-4\\_3](https://doi.org/10.1007/978-3-642-36318-4_3)

Box, G., & Jenkins, G. (1970). *Time Series Analysis: Forecasting and Control*. San Francisco, CA: Holden-Day.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). <https://doi.org/10.1145/2939672.2939785>

Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE International Conference on Big Data (Big Data)*. <https://doi.org/10.1109/bigdata.2015.7364089>

Chou, J. S., & Tran, D. S. (2018). Forecasting energy consumption time series using machine learning techniques based on usage patterns of residential householders. *Energy*, 165, 709–726. <https://doi.org/10.1016/j.energy.2018.09.144>

Vapnik, V., Drucker, H., Burges, C. J., Kaufman, L., & Smola, A. (1997). Support vector regression

machines. *Advances in Neural Information Processing Systems*, 9, 155–161.

Eassa, A. (2018). Why NVIDIA's Stock Crashed. Retrieved from <https://finance.yahoo.com/news/why-nvidia-apos-stock-crashed-122400380.html>

Emir, S., Dincer, H., & Timor, M. (2012). A stock selection model based on fundamental and technical analysis variables by using artificial neural networks and support vector machines. *Review of Economics & Finance*, 2(3), 106–122.

Hatta, A. J. (2012). The company fundamental factors and systematic risk in increasing stock price. *Journal of Economics, Business and Accountancy*, 15(2), 245–256. <http://doi.org/10.14414/jebav.v15i2.78>

Hill, J. B., & Motegi, K. (2019). Testing the white noise hypothesis of stock returns. *Economic Modelling*, 76, 231–242. <https://doi.org/10.1016/j.econmod.2018.08.003>

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>

Ke, G., Meng, Q., Finley, T., Wang, T., & Chen, W. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, 3146–3154.

Kozachenko, L. F., & Leonenko, N. N. (1987). Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2), 9–16.

Laptev, N., Yosinski, J., Li, L. E., & Smyl, S. (2017). Time-series extreme event forecasting with neural networks at Uber. *International Conference on Machine Learning*, 34, 1–5.

Lo, A. W. (2004). The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *The Journal of Portfolio Management*, 30(5), 15–29. <https://doi.org/10.3905/jpm.2004.442611>

Mahmoud, A., & Sakr, S. (2012). The predictive power of fundamental analysis in terms of stock return and future profitability performance in Egyptian Stock Market: Empirical Study. *International Research Journal of Finance & Economics*, 92(1), 43–58.

Milosevic, N. (2016). Equity forecast: Predicting long term stock price movement using machine learning. *Journal of Economics Library*, 3(2), 288–294. <http://doi.org/10.1453/jel.v3i2.750>

Muhammad, S., & Ali, G. (2018). The relationship between fundamental analysis and stock returns based

on the panel data analysis; evidence from Karachi Stock exchange (KSE). *Research Journal of Finance and Accounting*, 9(3), 84–96.

Nvidia Corporation. (2018). Nvidia Corporation Annual Review. Retrieved from [https://s22.q4cdn.com/364334381/files/doc\\_financials/annual/2018/NVIDIA2018\\_AnnualReview-\(new\).pdf](https://s22.q4cdn.com/364334381/files/doc_financials/annual/2018/NVIDIA2018_AnnualReview-(new).pdf).

Pai, P. F., & Lin, C. S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega*, 33, 497–505. <https://doi.org/10.1016/j.omega.2004.07.024>

Preis, T., Moat, H. S., & Stanley, E. H. (2013). Quantifying trading behavior in financial markets using Google Trends. *Scientific Reports*, 3(1684), 1–6. <https://doi.org/10.1038/srep01684>

Rather, A. M., Agarwal, A., & Sastry, V. N. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications*, 42(6), 3234–3241. <https://doi.org/10.1016/j.eswa.2014.12.003>

Shim, J. K., & Siegel, J. G. (2007). *Handbook of Financial Analysis, Forecasting, and Modeling* (p. 255). Chicago, USA: CCH.

Stanković, J., Marković, J., & Stojanović, M. (2015). Investment strategy optimization using technical analysis and predictive modeling in emerging markets. *Procedia Economics and Finance*, 19, 51–62. [https://doi.org/10.1016/S2212-5671\(15\)00007-6](https://doi.org/10.1016/S2212-5671(15)00007-6)

Whittle, P. (1951). *Hypothesis Testing in Time Series Analysis*. Uppsala, Sweden: Almqvist & Wiksells boktr.

Yang, K., & Shahabi, C. (2005). On the stationarity of multivariate time series for correlation-based data analysis. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, Houston. <https://doi.org/10.1109/ICDM.2005.109>

Zeytinoglu, E., Akarim, Y. D., & Celik, S. (2012). The impact of market-based ratios on stock returns: The evidence from insurance sector in Turkey. *International Research Journal of Finance and Economics*, 84, 41–48.

Zheng, A., & Jin, J. (2017). Using AI to make predictions on stock MARKET. Stanford University. Retrieved from <http://cs229.stanford.edu/proj2017/final-reports/5212256.pdf>