University of Warsaw
Faculty of Economic Sciences

Working Papers
No. 22/2020 (328)

# NVIDIA'S STOCK RETURNS PREDICTION USING MACHINE LEARNING TECHNIQUES FOR TIME SERIES FORECASTING PROBLEM

Marcin Chlebus
Michał Dyczko
Michał Woźniak

Warsaw 2020

UNIVERSITY OF WARSAW
FACULTY OF ECONOMIC SCIENCES
WORKING PAPERS

# Nvidia's stock returns prediction using machine learning techniques for time series forecasting problem

**Marcin Chlebus[a] , Michał Dyczko[b], Michał Woźniak[a]***

*[a] Faculty of Economic Sciences, University of Warsaw*
*b Faculty of Mathematics and Computer Science, Warsaw University of Technology*
** Corresponding author: mj.wozniak9@student.uw.edu.pl*

**Abstract:** The main aim of this paper was to predict daily stock returns of Nvidia Corporation company quoted on Nasdaq Stock Market. The most important problems in this research are: statistical specificity of return ratios i.e. time series might occur to be a white noise and the fact of necessity of applying many atypical machine learning methods to handle time factor influence. The period of study covered 07/2012 - 12/2018. Models used in this paper were: SVR, KNN, XGBoost, LightGBM, LSTM, ARIMA, ARIMAX. Features which, were used in models comes from such classes like: technical analysis, fundamental analysis, Google Trends entries, markets related to Nvidia. It was empirically proved that there is a possibility to construct prediction model of Nvidia daily return ratios which can outperform simple naive model. The best performance was obtained by SVR based on stationary attributes. Generally, it was shown that models based on stationary variables perform better than models based on stationary and non-stationary variables. Ensemble approach designed especially for time series failed to make an improvement in forecast precision. It seems that usage of machine learning models for the problem of time series with various explanatory variable classes brings good results.

**Keywords:** nvidia, stock returns, machine learning, technical analysis, fundamental analysis, google trends, stationarity, ensembling

**JEL codes:** C02, C32, C40, G12

## 1. Introduction

Some analysts and researchers claim that stock market prediction is like doing astrology. In spite of that fact, in the past years researchers have attempted to find novel and unbiased theoretical background which is useful to understand stock behaviour. Hypothesis that was a breakthrough for academic world in case of financial market modelling is The Adaptive Markets Hypothesis (Andrew Lo 2004). It implicates possibility of making use of achievements from: fundamental analysis, technical analysis and behaviour analysis with decent results. At the same time, last decade was a renaissance of supervised machine learning algorithms which are used in time series prediction problems in such fields like: energetics (Chou and Tran 2018), finance (Abe and Nakayama 2018), logistics (Laptev et al. 2017) etc. These two states are drivers for development in this area, especially in stock return ratio forecasting.

The main aim of this study is to predict daily stock returns of Nvidia Corporation company quoted on Nasdaq Stock Market. The most important problems standing in front of researchers in that field are: statistical specificity of return ratios i.e. time series might occur to be a white noise (Hill and Motegi 2017), necessity of applying many atypical machine learning methods to handle time factor influence (Bontempi et al. 2013) and lack of researches based on regressive attitude to benchmark with.

Nvidia Corp. choice is intentional in economic sense because of its unique and complex cross-sectoral structure, including games, deep learning and cryptocurrency market. From modelling perspective it makes possibility to use many features that could have significant impact on response variable. Moreover, one can observe rapid Nvidia`s stock price changes over the last months (October 2018 – December 2018) period connected with bessa on american stock exchanges. It is interesting if models are able to detect these fluctuations and predict stock ratios well.

As mentioned before, in this paper models will be based on variables from various categories: fundamental analysis of Nvidia, technical analysis of Nvidia stock prices, behaviour analysis from Google Trends data and Nvidia marketing news, analysis of markets related to graphic card manufacturers. Time period of research is 2012/07/02 - 2018/12/31 divided into training (2012/07/02 - 2018/06/29) and testing set (2018/07/02- 2018/12/31). It is driven by the will of examining models performance in that peculiar time chunk. Algorithms which will be applied on this data origins from two types of approaches - machine learning: SVR, KNN, XGBoost, LightGBM, LSTM Networks; and econometric methods: ARMA, ARMAX. What's more, ranking based ensemble model on previously mentioned algorithms will be provided.

The major hypothesis verified in this paper is whether it is possible to construct prediction model of Nvidia's daily return ratios which can outperform simple naive model. The additional research questions are: will models cope with market fluctuations that began in October 2018?; do models based on stationary variables perform better than models based on stationary and non-stationary variables?; will machine learning models be more appropriate than traditional statistical-econometric methods?; will ranking based ensemble models perform better than singular ones?; will categories of variables which are suggested in literature be significant?

The structure of this paper was composed as follows. The second part contained the literature review. The third part was devoted to materials and methodology used in research. In the fourth part empirical results and answers on hypothesis are presented. The summary of paper and conclusions are included in the last part.

## 2. Literature review

Studies devoted to stock returns prediction in regression problems using machine learning techniques is quite skimpy, especially as far as one-day-ahead forecast is concerned. Authors that made a significant contribution in that subject are Abe and Nakayama (2018). Their study examined performance of DNN, RF and SVM in predicting one-month-ahead stock returns in the cross-section in the Japanese stock market. What's more, weighted ensemble method was also conducted in above-mentioned analysis. The experimental results indicate that deep neural networks and weighted ensemble method show promise at stock returns forecast. Another example is provided by Pai and Lin (2004) who attempted 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 capability of hybrid model to forecast time series. In their case, simple average of two models doesn't gain impact. Adebiyi et al. (2014) applied ANN and ARIMA to examine their performance on daily stock data from NYSE. The empirical results present superiority of neural networks model over ARIMA model. It is worth to mention that there are also interesting works on classification problem of stock prices. Zheng and Jin (2017) analyzed performance of Logistic Regression, Bayesian Network, LSTM, and SVM on daily Microsoft stock prices, extended by technical indicators. As a result, they can achieve a correct prediction of the price trends at level of 70%. SVM gained the best score. Milosevic (2016) forecasted long term stock price movement using i.e. RF, Logistic Regression, Bayesian Networks. In this research author gathered quarterly stock prices from over 1700 stocks combined with data from fundamental analysis. He showed that RF performed the best and he also provided the most appropriate set of fundamental indicators.

According to hitherto achievements of researchers regarding stock return forecasting, in this paper following econometric models will be used: ARIMA developed by Whittle (1951) and popularized by Box and Jenkins (1970), ARIMAX which provides possibility of ARIMA model extension with additional exogenous variables. Nonlinear supervised machine learning models applied in this research are: SVR created by Vapnik (1997), KNN designed by Altman (1990). Models from the same category as above, based on boosted decision trees are XGBoost developed by Chen and Guestring (2016) and LightGBM created by Ke et al. (2017). During studies RNN in LSTM architecture constructed by Hochreiter (1997) was also deployed. To conduct ensembling a Model Ranking Based Selective Ensemble Approach powered by paper of Adhikari et al. (2014) was used.

A very important part of the whole analysis is to collect variables from diverse thematic classes i.e.: fundamental analysis, technical analysis, behavioral analysis, expert indicators of markets related to Nvidia. Undoubtedly the best described in the literature is the selection of variables from the fundamental analysis for the problem of prediction of stock prices and stock returns. Mahmoud and Sakr (2012) proved that fundamental variables explaining such issues as: profitability, solvency, liquidity and operational efficiency are helpful in creating effective investment strategies. Dhatt et al. (1999) analyzed the relationship between the stock returns and the variables from the fundamental analysis for the Korean stock exchange in 1982-1992. They found, among others, a positive correlation between the return ratio and the Price to Book Ratio and Debt-Equity Ratio. Akarim et al. (2012) showed in their work dedicated to the Turkish insurance market that good predictors of stock returns are mostly: Price/Profit, Earnings per Share and Price to Book Ratio. Titman (1993) emphasized the need to pay attention to the variable Price to Book Ratio. Hatta (2012) in his research discovered that Earnings Per Share has a positive impact on the rate of return and it is negatively correlated with the Debt to Equity Ratio. Shakeel and Gohar (2018) divided the variables into categories that describe the company as a whole: liquidity (current liquidity), market value (price / profit, EPS), profitability (ROA), indebtedness (general debt). Their research conducted for the emerging Pakistani exchange shows that the variables ROA (positive coefficient) and Price/Earnings (positive coefficient) were highly significant in the modeling of the return rates. The remaining variables were irrelevant or showed unreasonable dependencies. The use of variables from technical analysis is slightly less systematized in academic literature than variables from fundamental analysis. Marković et al. (2015) investigated the effectiveness of using three indicators of technical analysis: EMA (Exponential Moving Average), MACD (Moving Average Convergence-Divergence), RSI (Relative Strength Index) to predict the stock returns on selected stock exchanges in the countries of former Yugoslavia. According to their research, EMA and MACD perform best, while RSI was not able to predict rates of return well. Hybrid models with variables from technical and fundamental analysis is also well-described in literature. Beyaz et al. (2018) performed an experiment involving the prediction of share prices of companies from the S&P index in time windows: 126 days and 252 days. Their analysis showed that hybrid models work better than single models. They also examined the significance of variables in the hybrid approach (division into three tiers). In the highest, i.e. the most effective tier there were, for instance: ATR and EPS. Dincer et al. (2012) conducted a research for the specifics of the Turkish stock exchange. They modeled the ranking of ten the most profitable companies on the market. In the analysis, they used a hybrid approach that turned out to be much more effective than individual models. Variables that occurred to be significant in the analysis are eg Price per Book Value, ROE, CCI, RSI. Behavioral analysis of speculators is divided in the literature into: sentiment analysis and analysis of the popularity of search terms on the Internet. This paper is focused on the second approach. Asif et al. (2017) collected data from Google Trends to capture the dependencies between online searches and political and business events. They used this knowledge to predict the ups and downs of the Pakistan stock exchange 100 index by quantifying the semantics of the international market. Their research shows that these variables have good prognostic properties. Moat et al. (2013) utilized Google Trends data to predict value of Dow Jones Industrial Average Index. They claim that this features provide some insight into future trends in the behavior of economic actors and may be decent factor in the decision making process for investors. Moreover the Nvidia Annual Financial Report (2018) legitimise the need to collect indicators connected with Nvidia related markets like: deep learning, gaming and crypto currency.

Yang and Shahabi (2005) show that not only model and variables choice are important, but also feature engineering. They performed several classification experiments using techniques which are based on correlation coefficient on four data sets and investigated how stationarity of data influences forecast accuracy. Results of their work imply that one can obtain higher accuracy in prediction after differencing non-stationary data while differencing stationary data makes forecast less accurate. Authors suggests that test of stationarity and differencing features is recommended pre-processing step.

Nvidia stock price dropped rapidly in 3rd quarter of 2018 by more than 30 percent after earning results. Experts (Abazovic 2018; Eassa 2018) connect that fact with previous diminishment of crypto interest and mining. This led to severe demand decrease for graphic processing units which are Nvidia's main product line useful in cryptocurrencies mining. It is not a secret: gaming market is an important rock solid component of Nvidia's business and it also followed sharp crypto falloff.

## 3. Materials and methods

### *3.1 Dataset*

Dataset preparation step was crucial in this research. After data pre-processing it contains over 350 variables from various categories. Data covered in this paper comes from 2012/01/01 to 2018/12/31. All information was collected in January 2019.

#### *3.1.1. Data sources*

The key variables used in the article are among others: opening price, closing price, highest price, lowest price and volume for shares like: 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: S&P500, NASDAQ-100. The shares include entities that are: competition for Nvidia and their close business partners (sometimes unintentionally). On the other hand, the indices show the overall market situation. They were collected from Yahoo Finance.

Nvidia fundamental analysis features origins from balance sheet are available on Macrotrends.net. What's more this web page provide formulas for all fundamental indicators which were used in this research. 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**: Currrent 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, Receiveable Turnover, Days Sales Outstanding;
- **Market ratios**: Earning per share, Price to book value ratio, Book value per share, Price to Earnings ratio;

Behavioral analysis was based on people's demand for information taken from the Internet, especially from Google Search and Google News. This data are available on Google Trends platform. Regarding study it was decided to look for entries like:

- **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 the area covering the whole world.

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. The data were collected using web-scraping.

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

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

What's more dataset was extended with such features like: day of a week, day of a year, month, quarter to handle time series specificity of research.

### *3.1.2. Feature preparation*

Return ratios for: NVIDIA, BTC, UBSFY, ATVI, TTWO, BTC, S&P500, NASDAQ-100 were calculated using formula $return\ ratio = \frac{price_t - price_{t-1}}{price_{t-1}}$. On the same variables 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 this new variable. Gathered features are:

- **Overlap Studies**: Exponential Moving Average (EMA), Double Exponential Moving Average (DEMA), Hilbert Transform - Instantaneous Trendline (HT TRENDLINE), Kaufman Adaptive Moving Average (KAMA), Midpoint over period (MIDPOINT), Midpoint Price over period (MIDPRICE), Parabolic SAR (SAR), Parabolic SAR - Extended (SAREXT), Triple Exponential Moving Average (TEMA), Triangular Moving Average (TRIMA), Weighted Moving Average (WMA);
- **Momentum Indicators**: Momentum (MOM), Commodity Channel Index (CCI), Relative Strength Index (RSI), Williams' %R (Will R), Money Flow Index (MFI), Directional Movement Index (DX), Plus Directional Movement (PLUS DM), Percentage Price Oscillator (PPO), Aroon Oscillator (AROONOSC), Balance Of Power (BOP), Minus Directional Movement (MINUS DM), Ultimate Oscillator (ULOTSC), Average Directional Movement Index (ADX), Average Directional Movement Index Rating (ADXR), Absolute Price Oscillator (APO), Chande Momentum Oscillator (CMO), Minus Directional Indicator (MINUS DI), Plus Directional Indicator (PLUS DI), Rate of change (ROC), Rate of change Percentage (ROCP), Rate of change ratio (ROCR), Rate of change ratio 100 scale (ROCR100);
- **Volume Indicators**: Chaikin A/D Line (AD), On Balance Volume (OBV), Chaikin A/D Oscillator (ADOSC);
- **Volatility Indicators**: Average True Range (ATR), Normalized Average True Range (NATR), True Range (TRANGE);
- **Price Transform**: Average Price (AVGPRICE), Median Price (MEDPRICE), Typical Price (TYPPRICE), Weighted Close Price (WCLPRICE);
- **Cycle Indicators**: Hilbert Transform - Dominant Cycle Period (HT DCPERIOD), Hilbert Transform - Dominant Cycle Phase (HT DCPHASE), Hilbert Transform - Trend vs Cycle Mode (HT TRENDMODE);
- **Pattern Recognition**: Two Crows (CDL2CROWS), Three Black Crows (CDL3BLACKCROWS), Three Inside Up/Down (CDL3INSIDE), Three-Line Strike (CDL3LINESTRIKE), Three Outside Up/Down (CDL3OUTSIDE), Three Stars In The South (CDL3STARSINSOUTH), Three Advancing White Soldiers (CDL3WHITESOLDIERS), Abandoned Baby (CDLABANDONEDBABY),

Advance Block (CDLADVANCEBLOCK), Belt-hold (CDLBELTHOLD), Breakaway (CDLBREAKAWAY), Closing Marubozu (CDLCLOSINGMARUBOZU), Concealing Baby Swallow (CDLCONCEALBABYSWALL), Counterattack (CDLCOUNTERATTACK), Dark Cloud Cover (CDLDARKCLOUDCOVER), Doji (CDLDOJI), Doji Star (CDLDOJISTAR), Dragonfly Doji (CDLDRAGONFLYDOJI), Engulfing Pattern (CDLENGULFING), Evening Doji Star (CDLEVENINGDOJISTAR), Evening Star (CDLEVENINGSTAR), Up/Down-gap side-by-side white lines (CDLGAPSIDESIDEWHITE), Gravestone Doji (CDLGRAVESTONEDOJI), Hammer (CDLHAMMER), Hanging Man (CDLHANGINGMAN), Harami Pattern (CDLHARAMI), Harami Cross Pattern (CDLHARAMICROSS), High-Wave Candle (CDLHIGHWAVE), Hikkake Pattern (CDLHIKKAKE), Modified Hikkake Pattern (CDLHIKKAKEMOD), Homing Pigeon (CDLHOMINGPIGEON), Identical Three Crows (CDLIDENTICAL3CROWS), In-Neck Pattern (CDLINNECK), Inverted Hammer (CDLINVERTEDHAMMER), Kicking (CDLKICKING), Kicking - bull/bear determined by the longer marubozu (CDLKICKINGBYLENGTH), Ladder Bottom (CDLLADDERBOTTOM), Long Legged Doji (CDLLONGLEGGEDDOJI), Long Line Candle (CDLLONGLINE), Marubozu (CDLMARUBOZU), Matching Low (CDLMATCHINGLOW), Mat Hold (CDLMATHOLD), Morning Doji Star (CDLMORNINGDOJISTAR), Morning Star (CDLMORNINGSTAR), On-Neck Pattern (CDLONNECK), Piercing Pattern (CDLPIERCING), Rickshaw Man (CDLRICKSHAWMAN), Rising/Falling Three Methods (CDLRISEFALL3METHODS), Separating Lines (CDLSEPARATINGLINES), Shooting Star (CDLSHOOTINGSTAR), Short Line Candle (CDLSHORTLINE), Spinning Top (CDLSPINNINGTOP), Stalled Pattern (CDLSTALLEDPATTERN), Stick Sandwich (CDLSTICKSANDWICH), Takuri (Dragonfly Doji with very long lower shadow) (CDLTAKURI), Tasuki Gap (CDLTASUKIGAP), Thrusting Pattern (CDLTHRUSTING), Tristar Pattern (CDLTRISTAR), Unique 3 River (CDLUNIQUE3RIVER), Upside Gap Two Crows (CDLUPSIDEGAP2CROWS), Upside/Downside Gap Three Methods (CDLXSIDEGAP3METHODS);

- **Statistic Functions**: Pearson's Correlation Coefficient (CORREL), Linear Regression (LINEARREG), Linear Regression Angle (LINEARREG ANGLE), Linear Regression Intercept (LINEARREG_INTERCEPT), Linear Regression Slope

(LINEARREG SLOPE ), Standard Deviation (STDDEV), Time Series Forecast (TSF), Variance (VAR);

- **Math Transform Functions:** Vector Trigonometric ATan (ATAN), Vector Ceil (CEIL), Vector Trigonometric Cos (COS), Vector Trigonometric Cosh (COSH), Vector Arithmetic Exp (EXP), Vector Floor (FLOOR), Vector Log Natural (LN), Vector Log10 (LOG10), Vector Trigonometric Sin (SIN), Vector Trigonometric Sinh (SINH), Vector Square Root (SQRT), Vector Trigonometric Tan (TAN), Vector Trigonometric Tanh (TANH);
- **Math Operator Functions:** CAPM Beta (BETA), Highest value over a specified period (MAX), Index of highest value over a specified period (MAXINDEX), Lowest value over a specified period (MIN), Index of lowest value over a specified period (MININDEX), Vector Arithmetic Mult (MULT), Vector Arithmetic Substraction (SUB), Summation (SUM).

Factors mentioned above were calculated on the assumption of default values of parameters in the software. Mathematical description of these variables is available at tadoc.org web-page (http://tadoc.org/). The source code of technical features, which were used, is published in Go programming language by Mark Chenoweth on GitHub (https://github.com/markcheno/go-talib/blob/master/talib.go).

Min-Max Scaler was applied on all variables from Google Trends and Google News and newly generated data were appended to primary data set. Regarding that scaled and non-scaled Google variables were taken into account.

Data from Google Trends and Google News, reciprocal of FPS and release dates of games had non-daily frequency, so they were locally interpolated by exponential smoothing and included into dataset.

Non-financial information published by Nvidia which were mentioned in previous subsection were discretized using univariate decision trees to specify best bins. Then binarization (one-hot-encoding) was applied on these features.

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

To obtain convenient form of data for supervised machine learning analysis, most of explanatory factors were shifted. In a majority of them 1-day lag were used, but for differenced variables following procedure was performed: for each variable lags from 1 to 30 were tested, and chosen one with the smallest RMSE between it and stock return time series. If lag was less than 30, then it was used instead of 1. As a result, mostly Google Trends attributes were shifted with lag not equal to 1. Regarding Nvidia's stock returns variable, lags from 1 to 8 days were considered.

*3.2. General methodology of research*

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

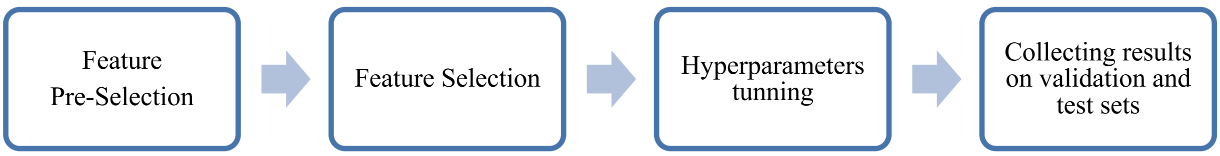
Performance of each model was interpreted as Root Mean Squared Error (RMSE) score between its predicted values and real values of stock return. The choice of this metric is implied by the fact that it emphasizes the importance of large individual errors. What's more, RMSE does not generate computational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

As a benchmark for every model, simple naive model will be used, which formula is: $\hat{y}_t = y_{t-1}$. This approach is appropriate and commonly used for financial problems (Shim and Siegel, 2007). Furthermore, there is no benchmark from literature for comparison.

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

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

**Figure 1. Algorithm of model building**




Sources: Own preparation.

Above mentioned algorithm will be proceeded on models like: ARIMA, ARIMAX, KNN, SVR, XGBoost, LightGBM, LSTM.

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

Feature preselection process will be based on Mutual Information algorithm for a continuous target variable, which will be conducted on each category of features from this research (Kozachenko and Leonenko, 1987). It builds a decent intuition for further choice of variables.

As a matter of fact, feature selection methodology depends on machine learning model (details in 3.4 subsection), but generally high-level approach will be provided as follows: hyperparameters will be chosen randomly from range proposed by experts and authors of models; dynamic forecast will be used because of time series modeling specificity; models during feature selection will be trained on train set and validated on concatenated validation sets. After that operation final lists of the best variables will be obtained for every model.

Model hyperparameters choice is crucial. 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 which provide variability of forecasted values (to prevent underfitting). Hyperparameters are selected in 3 steps as described in Algorithm 1.

**Algorithm 1**. Hyperparameters tuning algorithm.

1. **For each** pair of sets $(X_i, Y_i) \in S=\{(\text{train}, \text{validation}_1), (\text{train} \cup \text{validation}_1, \text{validation}_2), (\text{train} \cup \text{validation}_1 \cup \text{validation}_2, \text{validation}_3)\}$ 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 that 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. Forecasts on validation chunk will be used to prepare ensembling.

*3.4. Ensembling procedure.*

Ensemble algorithm that will be performed in this paper is an implementation of a Model Ranking Based Selective Ensemble Approach (Adhikari et al., 2014). Its methodology is based on weighted average. The algorithm is specially created for the problem of time series.
Let $MSE_i$ be mean squared error of forecast of i-th model on validation set, so i-th weight is expressed by:

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

Then formula for ensembling model will be:

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

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 that research will be tested various values of n parameter up to the predefined treshold.
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 will be three types of ensembling models i.e. established on models from $S$, $A$ or $M$.

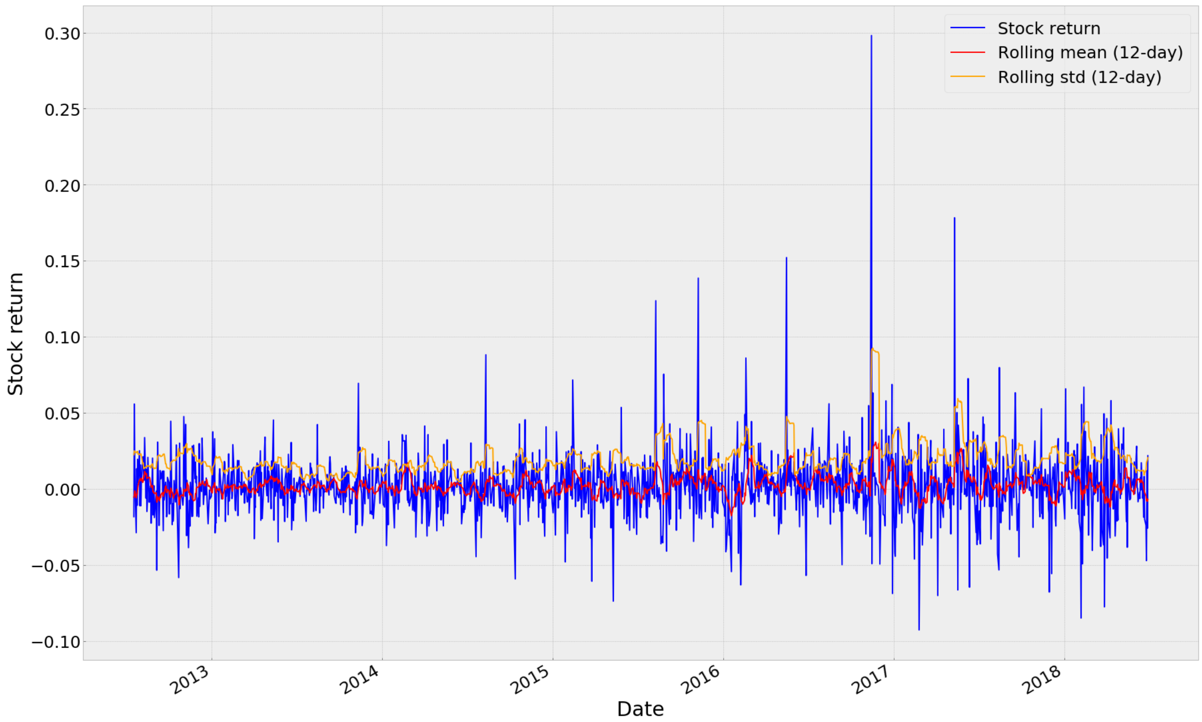
**4. Empirical results**

*4.1. Exploratory data analysis of target variable*

Studying properties of a dependent variable in a 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, cause it might occur to be a white noise (Hill and Motegi, 2017). In econometric theory it is not suitable for univariate forecast. Moreover statistical approach requires stationarity of endogenous feature (Davies and Newbold, 1979). These issues will be examined in this subsection.

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

**Figure 2. Nvidia stock returns on in-sample set**




Sources: Own calculations.

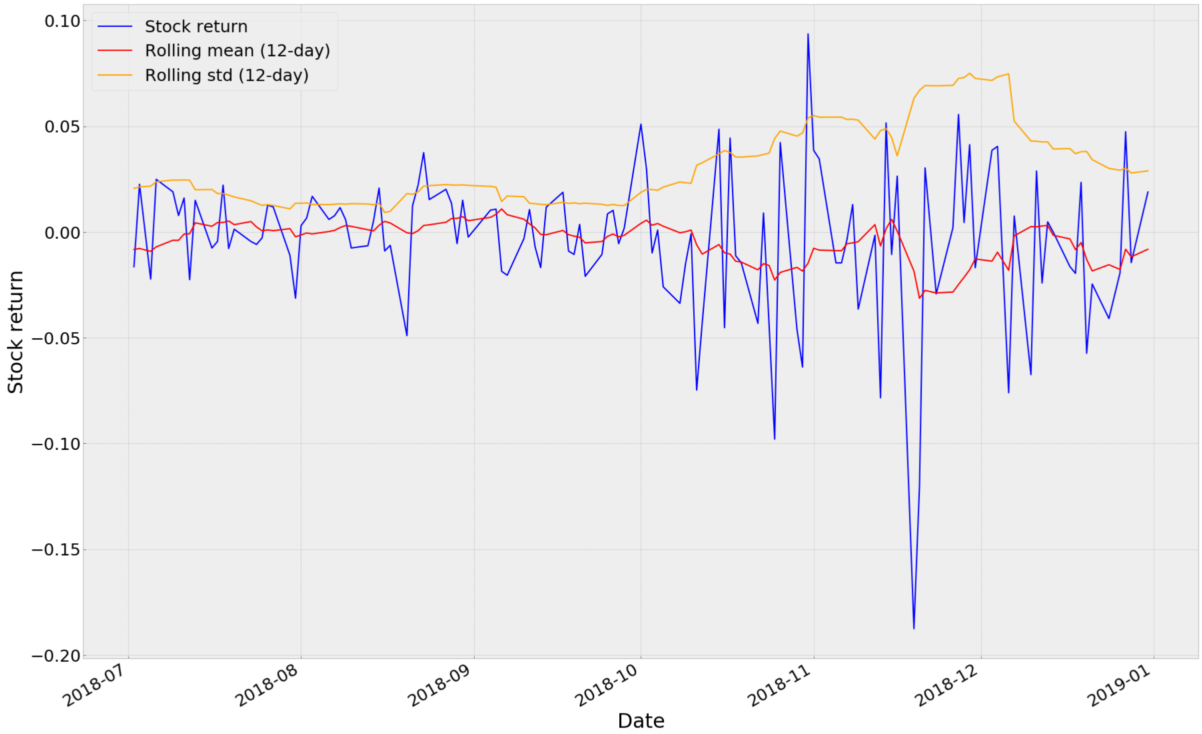
To check stationarity of time series on in-sample and out-of-sample chunk formal Augmented Dickey-Fuller test was performed. Algorithm implemented in Statsmodels selects number of lags automatically to assure that residuals are not autocorrelated. Table 1 confirms in a suspicion that stock returns in both periods are stationary.

**Table 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 |

Sources: Own calculations.

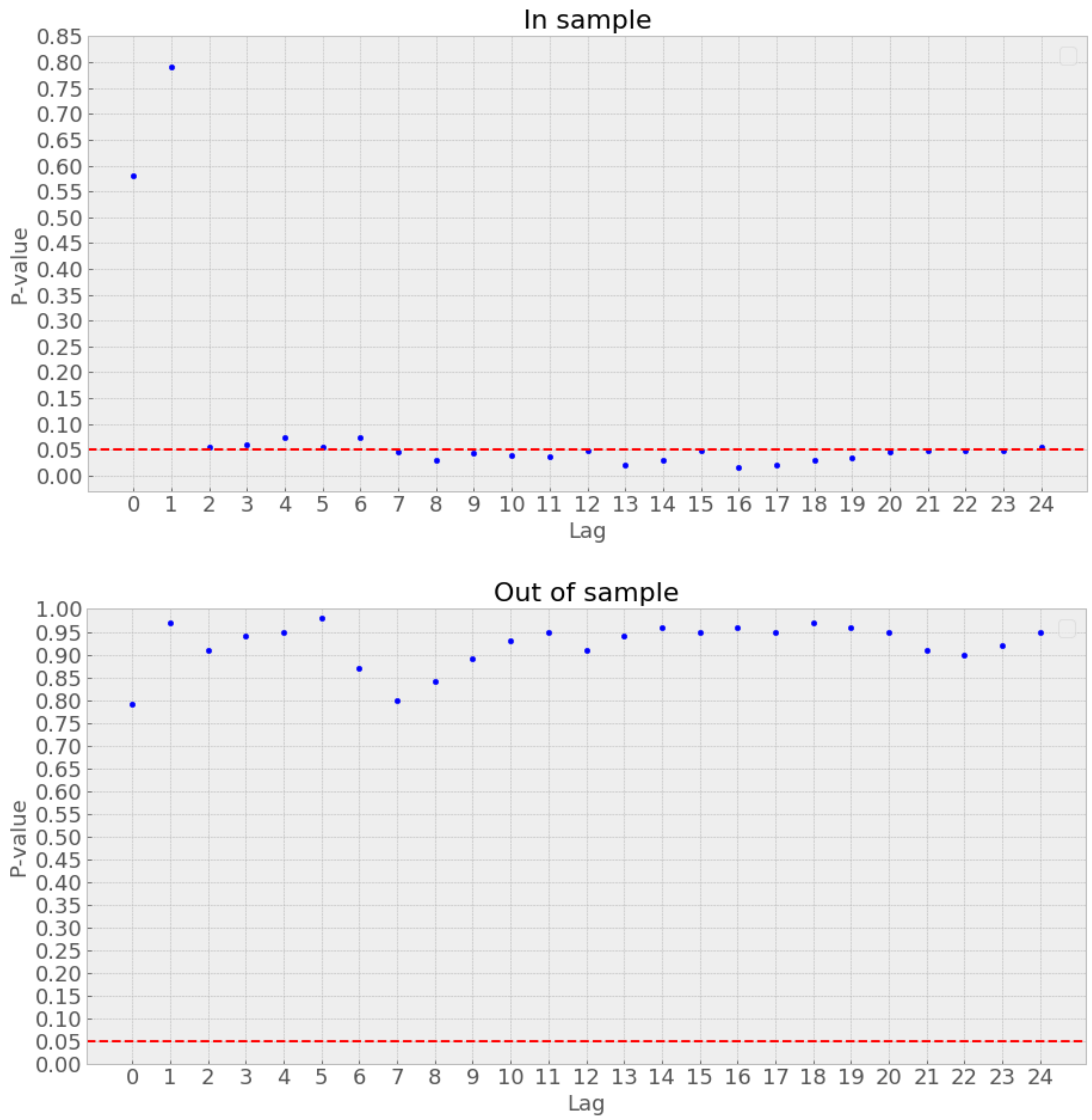
**Figure 3. Nvidia stock returns on out-of-sample set**




Sources: Own calculations.

To check if time series on in-sample and out-of-sample chunk is a white noise formal Ljung- Box test was done. As Figure 4 shows on in-sample set, the hypothesis of white noise is rejected. On the other hand, on out-of-sample chunk it is not possible to reject the white noise hypothesis.

**Figure 4. Results of Ljung-Box test for in-sample set and out-of-sample set.**



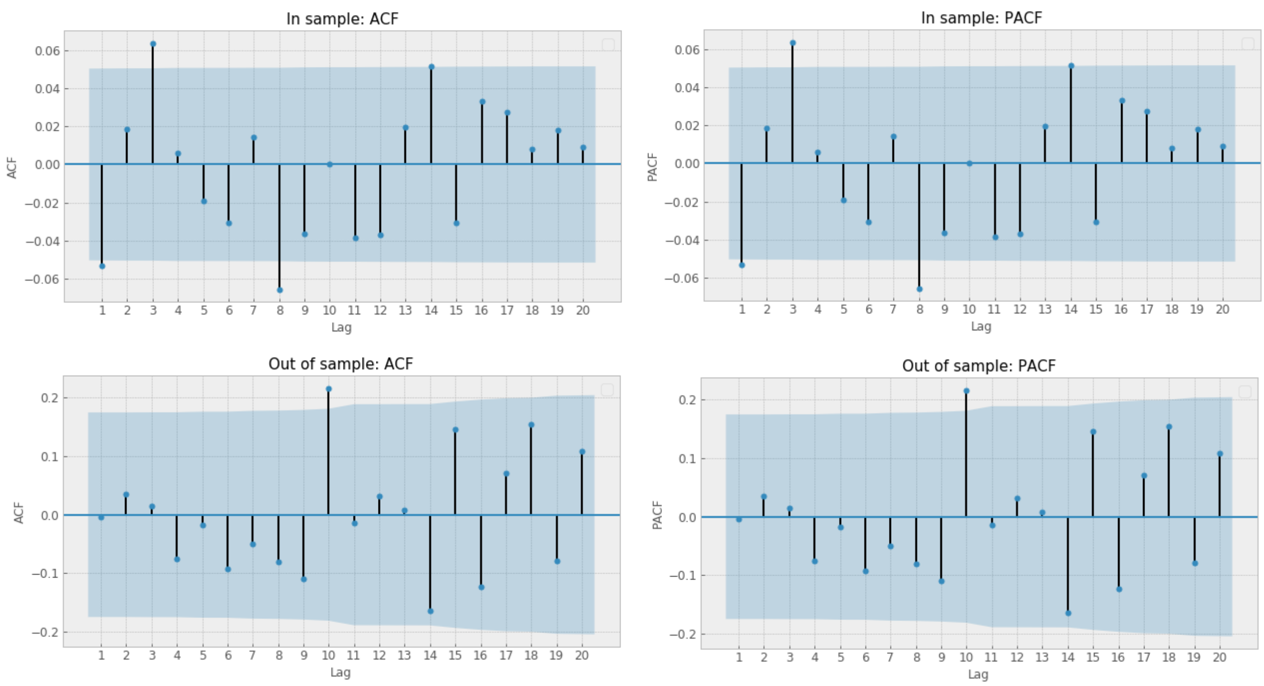

Notes: Figure presents results of Ljung-Box test of white-noise hypothesis of Nvidia's stock returns on in-sample set and out-of-sample set.
Sources: Own calculations.

Autocorrelation and partial correlation plots, presented on Figure 5, give deeper insight on lag significance. For in-sample set one of the smallest lags that are correlated are the first, the third and the eighth. It justifies the need to consider these lags in model building. According to out-of-sample chunk, all lags to ninth are not correlated which is another proof for testing period to be a white noise.

To sum up, the stock return on out-of-sample chunk is problematic for univariate prediction. Thus, it clearly corresponds with one of research hypothesis which is about model forecasting properties during bessa on the technology market in the second half of 2018.

Outliers captured on Figure 4 are consequence of Nvidia’s events like: presentation of quarterly financial results, premiere of a new graphics card or investors’ expectations mismatch.

**Figure 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.
Sources: Own calculations.

### *4.2. Singular models*

In Table 2. one can see results of all singular models: SVR; KNN; XGBoost; LightGBM and LSTM, based on stationary variables. It represents values of 3 metrics of estimation quality: RMSE; MAE; MedAE that each model gained on validation and test set, but also number of attributes and values of hyperparameters. Analogically, Table 3. contains results of these models based on stationary and non-stationary variables. There is no results for ARIMA/ARIMAX because assumptions of model were not satisfied.

For all of the models in Table 2. RMSE on test set was about 0.01 higher than on 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 test set, comparing with validation set, are much worse, which can be caused by difficulty of that period (white noise). It implies the lack of model’s stability.

**Table 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; optimizer function: Adam; batch size: 32; loss function: MSE.
Sources: Own calculations.

**Table 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<br>epsilon=0.092179 | 0.025632 | 0.019126 | 0.015488 |
| SVR (27) | Test | C=0.005317<br>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 | 0.039313 | 0.026863 | 0.018946 |

| | | | | | |
|---|---|---|---|---|---|
| | | k=6<br>weight function=uniform | | | |
| 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 | $H_2$ | 0.028334 | 0.021702 | 0.018201 |
| LSTM (20) | Test | $H_2$ | 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. $H_2$: number of hidden layers: 2 LSTM layers with batch normalization 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; optimizer function: SGD; regularization: bias regularizer at first layer and activity relularizer (L2) on second LSTM layer; dropout: 0.3 after first LSTM layer; batch size: 16; loss function: MSE.
Sources: Own calculations.

All of the models in Table 3. has RMSE on test set over 0.011 higher than on 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 test set, comparing with validation set, are significantly worse, which can be caused by white noise nature of that. It implies the lack of model's ability of stock returns forecast.

For major of models stationarity of variables didn't have impact of prediction quality, except KNN and LSTM, where usage of non-stationary variables in model estimation resulted in performance drop.

*4.3. Ensemble models*

Due to variety of machine learning models in this paper, it seems to be interesting to study the ensembling approach. Ensemble models were expected to outperform singular models. To check that, performances of ensembling models from three categories mentioned in methodology chapter were analyzed. 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, XGboost provides the best score of 0.036106 for RMSE in that category.

**Table 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 |
| **5** | **LightGBM (0.211671), KNN (0.204923), LSTM (0.201287), SVR (0.19595), XGBoost (0.18617)** | **0.036106** | **0.024094** | **0.015307** |

Note: Table represents performance of ensemble models on test set built of models based on stationary variables. Bolded row responds to the model with the lowest RMSE in that category of ensembling.
Sources: Own calculations.

Results of ensembling models, which are based on stationary and non-stationary features are collected in Table 5. Integration of SVR, LightGBM, KNN, XGBoost and LSTM provides the best results (RMSE: 0.038053). Models from table 4 are better than these 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 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 |
| **5** | **SVR (0.220323), LightGBM (0.216777), KNN (0.198253), XGBoost (0.184346), LSTM (0.180301)** | **0.038053** | **0.026068** | **0.016645** |

Notes: Table represents performance of ensemble models on test set built of models based on stationary and non-stationary variables. Bolded row responds to the model with the lowest RMSE in that category of ensembling.
Sources: Own calculations.

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

**Table 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 |
| **8** | **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)** | **0.036746** | **0.024681** | **0.01647** |
| 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. Bolded row responds to the model with the lowest RMSE in that category of ensembling.
Sources: Own calculations.

To sum up, weights of singular models were calculated basing on its RMSE on validation set. It was believed that it will improve forecasts of ensembling models. However, surprisingly it is an opposite. Singular models perform poorer on test set and indirectly it influences ensembling models accuracy (in RMSE meaning). What's more, as it has been observed, along with increase of number of models in ensembling algorithm, the results converge to the average (weights are splitted almost equally) and thus the variance decreases. Models are getting better (RMSE) but in the context of research, analysis of them lose sense.

### *4.4. Tabular summary*

To summarize the obtained models and compare its performances in a systematized way table 7 and 8 were prepared. Additionally using them two hypothesis were examined. Best ensemble model which is based on stationary variables is composed of: LightGBM, KNN, LSTM, SVR, 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 see 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 is also satisfactory for the aim of the research. In this case ensemble model is not able to overpass singular models, in particular SVR. All established models (from table 7) were able to outperform naive model.

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

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

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

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

**Table 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 | 0.041904 | 0.039313 | 0.040685 | 0.039593 | **0.037284** | **0.038053** | **0.050244** |
| MAE | 0.025875 | 0.026863 | 0.026906 | 0.028891 | **0.026295** | **0.026068** | **0.034908** |
| MedAE | 0.017279 | 0.018946 | 0.016939 | 0.020576 | **0.017959** | **0.016645** | **0.022378** |

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

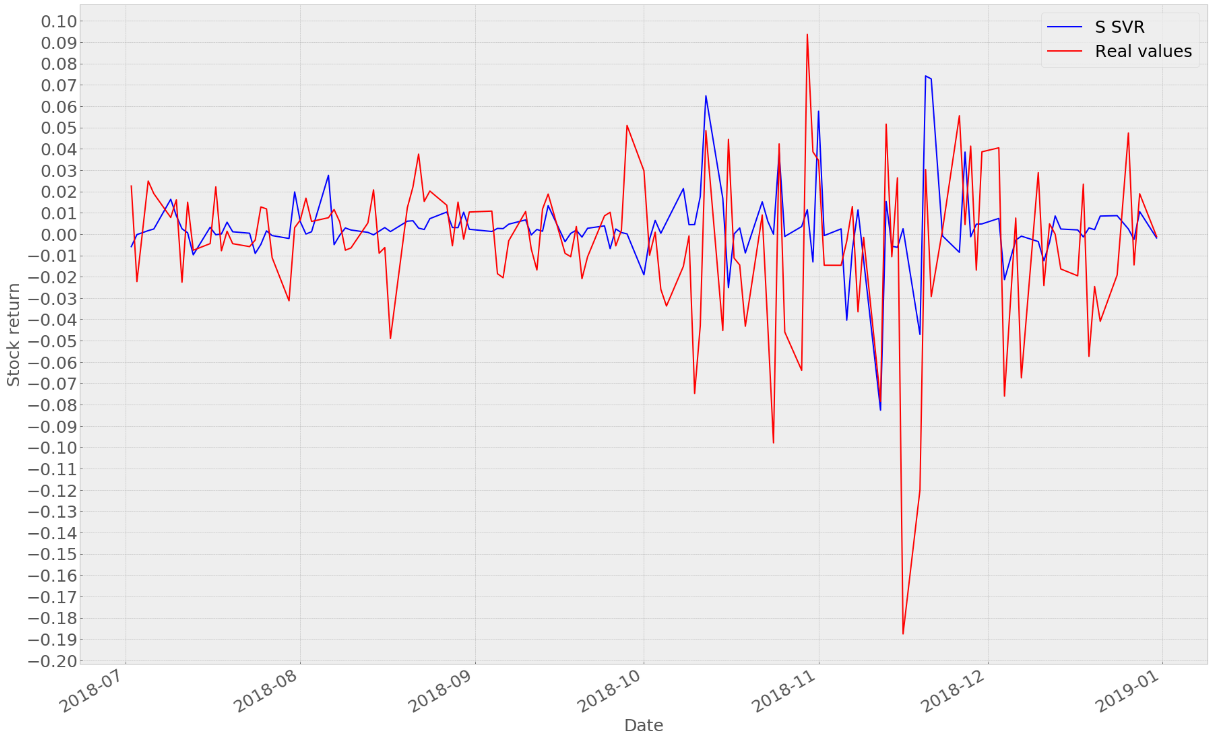
To have deeper insight into models efficiency, best ensemble models from 3 categories were compared in Table 9 with models that performs the best on stationary variables and all variables. Naive model was also included as a benchmark. Among models based on stationary features SVR occured to surpass other singular models and ensembling. 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 gives higher precision in forecast than models estimated on stationary and non-stationary variables. All in all, SVR model (based on stationary features) outperformed other models from table 9. Figure 9 presents its results on test set. This plot suggests that in the first period of testing (before bessa) model has a low and almost constant variance, however during second period fitted values deviates noticeably more, but model is able to 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.

**Table 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 | **0.036014** | 0.037284 | 0.050244 |
| MAE | 0.024094 | 0.026068 | 0.024681 | **0.024916** | 0.026295 | 0.034908 |
| MedAE | 0.015307 | 0.016645 | 0.01647 | **0.016682** | 0.017959 | 0.022378 |

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

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




Sources: Own calculations.

## 5. Conclusions

In this research all of hypothesis stated in the beginning have been analysed and answered. The main aim of this study was to predict daily stock returns of Nvidia Corporation company quoted on Nasdaq Stock Market.

Taking that into account, the major hypothesis verified in this paper is that it is possible to construct prediction model of Nvidia daily return ratios which can outperform simple naive model. According to results, outperforming simple naive model is not a challenging task, cause every final model had much lower RMSE of forecast residuals. The best score was obtained by SVR in version 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 where correlation among features is essential. Machine learning models were more appropriate than traditional statistical-econometric methods. Adebiyi et al. (2014) has shown superiority of neural networks model over ARIMA model in prediction profits from shares. ARIMA(X) failed to satisfy all assumptions imposed on them in that research, especially normality of residuals and non-zero variance, and that's why it was rejected and considered not to deal with stock returns prediction. Surprisingly ranking based ensemble models didn't perform better than singular ones, in contrary with conclusions drawn by Adhikari et al. (2014). Categories of variables which are suggested in literature are significant in final models. The combination of technical analysis and fundamental analysis as well as Beyaz et al. (2018) proposed, 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 Asif et al. (2017). It is worth to notice that singular variables recommended by literature e.g. Mahmoud and Sakr (2012) were always rejected in feature extraction.

There exist many noticeable challenges on this field that should be investigated in the future. Simple naive model occurred to perform poorly on test set. Thus, other models should be considered as a benchmark. What's important, and problematic this model should not have a variance converging to zero. 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 instead. Another improvement could be obtained using different algorithms of ensembling (blending and stacking). Regarding part of study connected with variables, sentimental analysis should be taken into account.

**References**

Abazovic, F. 21.11.2018. Nvidia stock dropped for the wrong reason. https://www.fudzilla.com/news/ai/47637-nvidia-stock-dropped-for-the-wrong-reason (Accessed 18.04.2019).

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.

Adebiyi, A., Adewumi, A., Ayo, Ch. 2014. Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction. Journal of Applied Mathematics 2014.

Adhikari, R., Verma, G., Khandelwal, I. 2015. A model ranking based selective ensemble approach for time series forecasting. Procedia Computer Science 48, 14-21.

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.

Altman N. 1992. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. The American Statistician 46 (3), 175-185.

Beyaz, E., Tekiner, F., Zeng, X., Keane, J. 2018. Stock Price Forecasting Incorporating Market State. 2018 IEEE 20th International Conference on High Performance Computing and Communications.

Bontempi, G., Taieb, S., Borgne, Y. 2012. Machine learning strategies for time series forecasting. European business intelligence summer school, 62-77.

Box, G., Jenkins, G. 1970. Time Series Analysis: Forecasting and Control. Holden-Day.

Chen, T., Guestrin, C. 2016. Xgboost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.

Chou, J., Tran, D. 2018. Forecasting energy consumption time series using machine learning techniques based on usage patterns of residential householders. Energy 165, 709-726.

Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems 2 (4), 303-314.

Davies, N., Newbold, P. 1979. Some Power Studies of a Portmanteau Test of Time Series Model Specification. Biometrika 66(1), 153-155.

Dhatt, M., Kim, H., Mukherji, S. 1999. Relations Between Stock Returns and Fundamental Variables: Evidence from a Segmented Market. Asia-Pacific Financial Markets 6(3), 221-233.

Drucker, H., Burges, Ch., Kaufman, L., Smola, A., Vapnik, V. 1997. Support vector regression machines. Advances in neural information processing systems, 155-161.

Eassa, A. 17.11.2018. Why NVIDIA's Stock Crashed. https://finance.yahoo.com/news/why-nvidia-apos-stock-crashed-122400380.html (Accessed 18.04.2019).

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.

Fonti, V., Belitser, E. 2017. Feature selection using LASSO. https://beta.vu.nl/nl/Images/werkstuk-fonti_tcm235-836234.pdf (Accessed 18.04.2019).

Guyon, I., Weston, J., Stephen, B., Vapnik, V. 2002. Gene Selection for Cancer Classification using Support Vector Machines. Machine Learning 46 (1-3), 389-422.

Hatta, J. 2012. The company fundamental factors and systematic risk in increasing stock price. Journal of Economics, Business and Accountancy 15 (2).

Hill, Jonathan B., Motegi, K. 2019. Testing the white noise hypothesis of stock returns. Economic Modelling 76, 231-242.

Hochreiter, S., Schmidhuber, J. 1997. Long short-term memory. Neural computation, MIT Press 9 (8), 1735-1780.

Jegadeesh, N., Titman, S. 1993. Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency. The Journal of Finance 48 (1), 65-91.

James G., Witten D., Hastie T., Tibshirani R. 2013. An Introduction to Statistical Learning with Applications in R. Springer,

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.

Kira, K., Rendell, L. 1992. A Practical Approach to Feature Selection, in: Sleeman, D., Edwards, P.(Eds.), Machine Learning Proceedings 1992. Morgan Kaufmann, pp. 249-256.

Kozachenko, N., Leonenko, N. 1987. Sample Estimate of the Entropy of a Random Vector. Probl. Peredachi Inf. 23(2), 9-16.

Laptev, N., Yosinski, J., Li, L., Smyl, S. 2017. Time-series extreme event forecasting with neural networks at uber. International Conference on Machine Learning 34, 1-5.

Lo, Andrew W. 2004. The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective. The Journal of Portfolio Management 30, 15-29.

Mahmoud, A., Sakr, Sameh. 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, 43.

Milosevic, N. 2016. Equity Forecast: Predicting Long Term Stock Price Movement using Machine Learning. Journal of Economics Library 3 (2), 288-294.

Nvidia Corporation. 2018. https://s22.q4cdn.com/364334381/files/doc_financials/annual/2018/NVIDIA2018_AnnualReview-(new).pdf. Nvidia Corporation Annual Review, (Accessed 18.04.2019).

Pai, P., Lin, Ch. 2005. A hybrid ARIMA and support vector machines model in stock price forecasting. Omega 33, 497-505.

Preis, T., Moat, H.S., Stanley, H.E. 2013. Quantifying Trading Behavior in Financial Markets Using Google Trends. Scientific Reports 3 (1684), 1-6.

Shakeel, M., Gohar, Ali. 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.

Shim, J., Siegel, J. 2007. Handbook of Financial Analysis, Forecasting, and Modeling. CCH. 255.

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.

Tibshirani, R. 1996. Regression Shrinkage and Selection Via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological) 58 (1), 1-300.

Whittle, P. 1951. Hypothesis testing in time series analysis. Uppsala, 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, -.

Zeytinoglu, E., Akarim 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, 2012. 41-48.

Zheng, A., Jin, J. 2017. Using AI to Make Predictions on Stock Market.

University of Warsaw
Faculty of Economic Sciences
44/50 Długa St.
00-241 Warsaw
www.wne.uw.edu.pl