Forecasting High-Frequency Excess Stock Returns via Data Analytics and Machine Learning*

Erdinc Akyildirim[1,2], Duc Khuong Nguyen[3,4,*], Ahmet Sensoy[5], Mario Šikić[2]

[1]*Mehmet Akif Ersoy University, Department of Banking and Finance, Burdur, Turkey*
[2]*University of Zurich, Department of Banking and Finance, Zurich, Switzerland*
[3]*IPAG Business School, Paris, France*
[4]*International School, Vietnam National University, Hanoi, Vietnam*
[5]*Bilkent University, Faculty of Business Administration, Ankara, Turkey*

Abstract

Borsa Istanbul introduced data analytics to present additional information about its market conditions. We examine whether this product can be utilized via various machine learning methods to predict intraday excess returns. Accordingly, these analytics provide significant prediction ratios above 50% with ideal profit ratios that can reach up to 33%. Among all the methods considered, XGBoost (logistic regression) performs better in predicting excess returns in the long-term analysis (short-term analysis). Results provide evidence for the benefits of both the analytics and the machine learning methods and raise further discussion on the semi-strong market efficiency.

Keywords: Big data, data analytics, machine learning, forecasting, efficient market hypothesis

JEL: C52, C53, D81, G14, G17

*The authors would like to thank an anonymous referee and the Editor John A. Doukas for their constructive comments that helped us significantly improve our paper. Ahmet Sensoy gratefully acknowledges support from the Turkish Academy of Sciences under its Outstanding Young Scientist Program.

*Corresponding author. Tel: +33 1 5363 3600 / Fax: +33 1 4544 4046 / Postal address: 184, bd Saint-Germain, 75006 Paris, France.

Email addresses: erdinc.akyildirim@bf.uzh.ch (Erdinc Akyildirim), duc.nguyen@ipag.fr (Duc Khuong Nguyen), ahmet.sensoy@bilkent.edu.tr (Ahmet Sensoy), mario.sikic@bf.uzh.ch (Mario Šikić)

1. Introduction

Big data is now seen as the future of many businesses and industrial sectors. Defined as the datasets whose size exceeds the ability of typical database software tools to capture, store, manage, and analyze, big data has now spread through every part of life (Manyika et al., 2011). Organizations are becoming increasingly digital day after day and, as a result, a large volume of data is being generated in their business operations (Sheng et al., 2017). Accordingly, the world has witnessed the explosive growth of data in terms of larger volume, higher velocity, and more diverse variety in various industries (Chen et al., 2012),[1] and big data has become one of the most valuable assets for many organizations in recent years (Albergaria and Jabbour, 2020). In parallel to these developments, finance industry has also evolved into a status where big data is an essential part of the day-to-day operations. In this study, we attempt to understand the benefits of analytics derived from a specific type of big data in the stock markets. We particularly aim to find out whether data analytics (derived from order and trade data) that are available as a commercial product in a leading emerging market can help day traders to forecast excess returns if they are to be combined with cutting-edge machine learning methods.

It is first opportune to note that unlike capital, big data has no value without the tools by which deeper insights can be extracted from it (Aydiner et al., 2019). Therefore, big data is often associated with the concept of analytics which refers to the ability to obtain useful information from big data by applying various quantitative methods to support decision making processes (Wang et al., 2016). Even though the positive and negative aspects of big data analytics usage are still debated (Ghasemaghaei, 2020), 80% of industry-leading firms have started big data analytics initiatives to bring greater accuracy to their decision-making, according to recent world-wide reports.[2]

According to Sivarajah et al. (2017), analytics can improve the decision-making process by making sense of big data for different types of analytic problems namely: (i) descriptive analytics which explains what is happening at the present; (ii) prescriptive analytics which plans what to do in the future; and (iii) predictive analytics which analyzes the likelihood of the future. Although all these types of analytics are important for various organizations, the predictive analytics is of uttermost importance in financial markets since it is

[1] As predicted by the International Data Corporation, it is projected by 2025 that there will be 175 zettabytes of data increased from 41 zettabytes in 2019 and only 2 zettabytes in 2010 (Reinsel et al., 2018).

[2] https://www.statista.com/statistics/742935/worldwidesurvey-corporate-big-data-initiatives-and-success-rate/

directly related to one of the most popular and controversial theories in finance, namely the efficient market hypothesis (EMH) introduced by Fama (1970).

EMH postulates that any past information should already be reflected into current prices so that they might be effected by only future events. To the extent that the future is unknown, prices should be consistently unpredictable. However, several works have shown so far that some market patterns can help generate abnormal returns thus violating the EMH, particularly the semi-strong EMH (e.g., Bernard and Thomas (1990); Lakonishok et al. (1994)), according to which abnormal returns cannot be earned by the help of all the available public information including a company's fundamentals, its historical stock market performance, or any other variables that may affect stock prices, such as economic factors.

Even though the concept of (semi-strong) EMH remains the same, the coverage of public information relevant to financial markets has changed drastically over the last three decades. With the availability of large amounts of high-quality data, faster and faster processors crunch ever more data such as macroeconomic announcements and monetary policy decisions, earnings statements, competitors' performance, consumer sentiment, and economic policy uncertainty (Begenau et al., 2018). In essence, financial markets are the place where analytics are most needed as they have the potential to forecast future returns. To meet this demand in the industry, various companies started to generate and sell big financial data analytics to help investors in financial decision making.

With that in mind, Borsa Istanbul (BIST - formerly known as Istanbul Stock Exchange) introduced a product called 'data analytics' on June 1st, 2016, covering 39 different analytics. These analytics are derived from the order and trade data of Borsa Istanbul's equity market and aim to present vital information about equity market conditions and trends to various types of market participants such as market makers, algorithmic traders, retail investors, smart order routing applications, analysts and risk managers.[3] Another important feature regarding the BIST analytics is their real-time dissemination which allows for reaching the meaningful data in a timely manner.

In this paper, we aim to examine whether the analytics product sold by Borsa Istanbul can be used to identify future stock price patterns at the high-frequency level using six different machine learning (ML)

[3]Another exchange that provides a similar product is the Deutsche Borse. To the best of our knowledge, no other exchange provides a similar product. The main reason is argued to be privacy concerns by algorithmic traders who fear that their trading strategies might be exposed with the help of these analytics.

algorithms. The answers to this question allow us to assess not only the utility of BIST data analytics for forecasting purposes but also the validity of the EMH based on the combination of high-quality publicly available data and advanced quantitative techniques. Note that the ML approach is highly promising in finance since the nature of the forecasting relation between predictive variables and stock returns can change over time. It has been successfully applied to various types of financial time-series forecasting such as stock prices (Patel et al., 2015) and stock volatility (Kim and Won, 2018), exchange rates (Colombo and Pelagatti, 2020), derivatives (Wang and Wang, 2019), commodities (Li et al., 2019) and even cryptocurrencies (Atsalakis et al., 2019). Its application in our current setup is particularly relevant because big data samples with complex nature representing the majority or the entire population may imply the invalidity of conventional statistical methods (Ngo et al., 2020).

Accordingly, our results reveal that data analytics can indeed help investors in predicting high-frequency future excess returns. For various selection of train and test partitions within the whole sample, we observe an average prediction accuracy rates of 53% across all stocks and methods, which can reach up to 58% for specific stock and methodologies. Among all methods considered, random forest has the highest forecasting accuracy in general. With regard to capturing potential gains, ideal profit ratio analysis reveals that ML methods can capture up to 42% of the all excess returns. In this case, extreme gradient boosting and random forest techniques are superior to others.

Next, we implement a dynamic setup via a sliding window approach for different selection of window size, train and test partitions. In this setup, each window that slides from beginning to the end of the sample generates a success and ideal profit ratio which eventually creates a sample of success ratios for each stock. Consequently, we test whether the mean of this sample is greater than 50% or not for these stocks individually and also investigate the average capability of capturing potential returns, represented in percentage terms, in excess of the benchmark index. Accordingly, we can forecast intraday excess returns with an accuracy significantly greater than 50%, independent of the ML method, window size, train and test partitions. Among all methods, the support vector machine generates the most consistent results with an average successful prediction ratio of 54%, with a maximum value of 57% in some specific cases. Average ideal profit ratio varies between 12% to 14%, where the random forest algorithm and the logistic regression classifier show slightly better performance compared to other ML methods.

In sum, our study makes three important contributions to the related literature. First, past studies have shown that high-frequency information regarding order and trade data can improve forecasting intraday and

daily returns (Chordia and Subrahmanyam, 2004; Hvidkjaer, 2008; Yamamoto, 2012; Cont et al., 2014; Cai and Zhang, 2016; Chelley-Steeley et al., 2019; Klein, 2020). However, these studies either (i) perform ex-post analyses since the main explanatory variables of order and trade data is not disseminated in real time but obtained later and examined in a back-testing environment, which makes the value of data questionable from practitioners' perspective[4] or (ii) use various classification algorithms (e.g., Lee and Ready (1991)) to make inference regarding the fundamental properties of the data, which is subject to specification error. By contrast, in our case, analytics are precise and disseminated in real time by the exchange itself, which prevents us from the abovementioned critics/problems.

Second, while applications of ML in forecasting financial time-series are mainly focused on textual-linguistic analysis or efficiently utilizing indicators obtained from historical prices of the asset itself[5], we implement ML methods on analytics derived from order and trade data of company stocks, providing important and direct implications for investment strategies of market participants.

Finally, unlike previous studies with information models where public information is taken as an input to test the market efficiency, data in our study is an actual asset that can be bought by the investors in the market. This would allow us to evaluate whether the analytics product introduced by BIST creates value for investors.

The rest of the study is organized as follows. Section 2 provides a brief literature review on testing semi-strong EMH in recent years and empirical applications of ML methods in finance. Section 3 explains the market under study and the analytics data in detail along with the features derived from analytics to be used in ML methods. Section 4 introduces the main ML approaches to be implemented. Section 5 reports and discusses the main results. Section 6 concludes the paper.

2. Literature Review

As argued earlier, modern financial markets have turned into a platform where almost anything and everything available publicly can influence asset prices. This highly motivates research in testing the semi-strong form of the efficient market hypothesis, in particular using public big data. For instance, Tetlock (2007)

[4]For example, Boehmer et al. (2008) and Diether et al. (2009) find that heavily shorted stocks underperform lightly shorted stocks. But, authors of both papers emphasize that their results do not contradict the semi-strong form of market efficiency because investors do not have access to their data in real time.

[5]See Henrique et al. (2019) for an excellent survey on this matter.

and Tetlock et al. (2008) perform textual analysis on newspapers to examine how content of the news affects stock returns and company earnings respectively. They find that market overreacts to extremely negative news and firm-specific negative news can predict firm earnings. Da et al. (2011) test whether company-related Google search volume predicts stock returns and find that a one standard deviation cross-sectional increase in search volume drives up stock returns in the subsequent week by 18 basis points. Internet search of terms such as "recession" and "unemployment" is also found to predict short-term reversals (Da et al., 2014). Jiang et al. (2019) construct a managerial sentiment index via text analysis of corporate disclosures and show that this index strongly predicts aggregate market returns. Ding et al. (2020) utilize a dataset from Seeking Alpha, the largest crowd-sourced social media platform that provides third-party generated financial analysis in US, and show a negative association between social media coverage and comovement as social media facilitates the incorporation of firm-specific information into stock price. Using partial least squares, Cepni et al. (2020) construct an investment sentiment index using the six main constituents that were also used in the sentiment index of Baker and Wurgler (2006). Authors show that the predictive power of their sentiment index is statistically significant, especially for bond premia with shorter maturities, even after controlling for a large number of financial and macro factors.

Studies testing the semi-strong EMH often differ in terms of both the content of the public information and its speed. The study of Chordia et al. (2005) finds, for example, that sophisticated investors react to order imbalances (imbalance between buyer and seller initiated trades) within more than five minutes but less than sixty minutes, which removes the serial-correlation for the daily returns of stocks listed on the NYSE and thus makes the market more efficient. Chung and Hrazdil (2012) extend this study to a broad panel of NYSE stocks to examine the relation between electronic communication networks and the corresponding informational efficiency of prices. Overall, they confirm the earlier results and further show that it takes about 20 minutes longer for smaller firms to react to order imbalances. Cont et al. (2014) document that, over short time intervals within the day, price changes of NYSE stocks are mainly driven by the order flow imbalance, defined as the imbalance between supply and demand at the best bid and ask prices. Bernile et al. (2016) show that abnormal order imbalances in the S&P500 futures contracts are in the direction of subsequent macroeconomic policy surprises and contain information that predicts the market's immediate reaction to the policy announcements. Klein (2020) uses high-frequency trading aggressiveness after earnings releases as a measure of crowding by sophisticated traders. Author finds that the prices of aggressively traded stocks overreact after good news, but not after bad news, except during the financial crisis.

While the content of the public information has seriously expanded over time and its diffusion speed increased dramatically, the methodological approaches have also improved. In recent years, we have witnessed several applications of ML techniques in testing the efficiency of financial markets. An earlier study of Antweiler (2004) uses naïve Bayes algorithms to perform computational linguistic analyses of internet bulletin board posts and shows that the increased disagreement in messages and the number of postings lead to increased volatility. Li (2010) finds that managerial disclosures predict accounting fundamentals by training a machine on a manually classified set of managerial disclosures. More recently, Amat et al. (2018) use ML methods to show that fundamentals from simple exchange rate models or Taylor-rule based models lead to improved exchange rate forecasts for major currencies over the floating period era of 1973–2014. Wang et al. (2018) construct extreme learning machine models with variant forecasting schemes and show that internet searching helps quantifying investor attention, improving the prediction of short-run price fluctuations in the oil market. Risse (2019) combines the discrete wavelet transform with support vector regression for forecasting gold price and shows evidence that disentangling the predictors with respect to their time and frequency domains leads to improved forecasting performance. Gu et al. (2020) perform a comparative analysis of ML methods to forecast asset risk premiums and demonstrate that ML methods can double the performance of leading regression-based strategies.[6]

The forecasting abilities of ML methods have been also utilized for developing prudential tools to have a better and healthy functioning financial system. To date, Lin et al. (2008) develop a hybrid causal model for predicting the occurrence of currency crises by using the neuro-fuzzy modeling approach. Their findings reveal that the new model leads to a better prediction of crisis. Khandani et al. (2010) apply ML techniques to construct nonlinear nonparametric forecasting models of consumer credit risk and find that their out-of-sample forecasts significantly improve the classification rates of credit-card-holder delinquencies and defaults. Finally, Gogas et al. (2018) and Beutel et al. (2019) work on early warning systems and forecasting models of bank failures based on machine-learning. While the former study reports a 99.22% overall forecasting accuracy of bank failures, the latter states that ML algorithms attain a very high in-sample fit, but they are outperformed by the logit approach in recursive out-of-sample evaluations.

Our current study joins the previous literature on forecasting stock returns, by exploiting the forecasting

[6]For further studies on the importance of ML based approaches in financial forecasting, see Kearney and Shang (2020); Aziz et al. (2021); Gonzalez et al. (2021).

abilities of ML techniques and using a big data set at high frequency from Borsa Istanbul Stock Exchange's real-time data analytics. Compared to the existing literature, we go one step further by considering order- and trade-based analytics which are publicly available in real time as a commercial product and show that excess returns are still predictable. Next, while previous studies usually focus on 5-minute or longer time intervals as the forecasting target, we use the 1-minute ahead returns which are argued to be prone to microstructure noise and find that return predictability continues to hold. Furthermore, we contribute to the literature on the benefits of using ML methods in predicting financial time series. Unlike the earlier studies, we show that the features used in these selected models do not have to be complicated or hard to obtain (such as web scraping, twitter feeds, google search, etc.) but they can be very simple variables solely depending on buy- and sell-side imbalance in the stock markets.

3. Market details, data and feature selection

Our empirical investigation focuses on the emerging equity market of Turkey (Borsa Istanbul) which attained, as of 2018, a market capitalization of $151 billion. With a daily average trading value of $1.7 billion and an annual share turnover velocity of 242%, this market is ranked 21th and 2nd in the world in terms of liquidity. It is also an attractive venue for investors worldwide given a a foreign ownership level of 65.01%.[7]

In terms of regulatory framework and legal aspect, although Turkey is not a member of the European Union (EU), Borsa Istanbul tries to follow the EU framework. In this regard, the legal landscape in the field of capital markets in Turkey underwent a major change. Capital Markets Law No. 2499 and Decree-Law No. 91 Regarding Securities Exchanges were repealed by the Capital Markets Law No. 6362 (CML) in 2012. The aim of the new CML is to harmonize the Turkish capital markets regulation with the EU acquis, to improve the integration of the Turkish capital markets with the European markets, and to enhance competitiveness thereof. This law not only brought the liberalization of the activity of running organized markets, but also re-structured Borsa İstanbul in 2013 as a joint-stock company subject to private law, having market stakeholders as shareholders, and thus allowing the realization of good governance principles. As a result, the exchange is a member of the World Federation of Exchanges, the Federation of Euro-Asian Stock Exchanges, the

[7] See the annual report of Borsa Istanbul for more detailed statistics via https://borsaistanbul.com/files/borsa-2018-annual-report.pdf.

International Securities Services Association, the International Capital Market Association, the European Capital Markets Institute, the World Economic Forum, and the Federation of European Securities Exchanges. This shows a highly integrated exchange to the global financial markets, in particular European ones.

In Borsa Istanbul, trading occurs every business day. As in various European equity markets, market opens in the morning with a call auction followed by a matching (price formation) period. Then, continuous trading begins and goes on until lunch break. During this break, another call auction takes place which is followed by price formation and then continuous trading again in the afternoon. Finally, market closes with the last call auction period followed by the matching period. Since analytics are disseminated only during the continuous trading sessions, our forecasting analysis is also performed only at this stage.

All data used in this study (intraday stock prices, benchmark index level and the data analytics) comes from Borsa Istanbul Group's database. Our sample covers a period of 649 days in 31 months starting from 1 June 2016 (the first day of analytics dissemination) until 31 December 2018. Since raw datasets are on the tick level, we aggregate the data to one-minute level in order to apply the machine learning algorithms on the same time scale. This yields 244918 one-minute time intervals in total. The whole dataset takes about 700 GB of memory but we are sill able to use a desktop computer for the machine learning analysis, thanks to the sliding window approach for prediction. The sliding window approach additionally gives us information about robustness of the classifiers. Table 1 provides descriptive statistics for both the data analytics and features which are constructed from these raw data analytics. Even though the analytics are calculated and distributed for the stocks listed in the BIST 100 Index, we limit ourselves to stocks listed only in the flagship index BIST 30 in order to exclude the effects of the heterogeneity in firm size (measured by market capitalization) among the top 100 companies and to work on a more manageable dataset. To protect ourselves from index inclusion and exclusion effects, we consider the company stocks that were included in the BIST30 index for the whole sample period. This leaves us with 28 stocks (see Table A.1 in Appendix for the whole list).

3.1. Feature selection

Borsa Istanbul's data analytics product covers 39 different analytics (see Table A.2 in Appendix for the whole list), however we do not use all of these analytics together for an important reason: The core part of the success of ML models are the selected features, which makes those unrelated to forecasting to cause underperformance of models. Features should depend on a common insight and also be generalizable. To the extent that our objective is to predict future price patterns and that asset price formation follow the demand

(buy) and supply (sell) theory as analogously in the market for goods and services, we limit our investigation to the analytics that help understand the imbalance between supply and demand in the stock market.

In the microstructure literature, according to Chordia and Subrahmanyam (2004), there are at least two reasons why such buy-sell imbalances can provide additional power beyond the ordinary trading activity measures such as volume in explaining asset returns. First, a high absolute buy-sell imbalance can alter returns as liquidity providers and market makers struggle to re-adjust their inventory. Second, these imbalances can signal excessive investor interest in an asset, and if this interest is persistent, then buy-sell imbalances could be related to future returns. Based on these arguments, imbalance between demand and supply is clearly an important descriptor that allows us to understand the general sentiment and direction the market is headed.

Using all available data analytics, we try to construct variables that reflect different aspects of the buy-sell imbalance in the market. For example, in Table 4, features NO_1 and QO_1 focus on the imbalance between all buy and sell orders coming to the market from the perspective of number and volume, respectively. These orders display the imbalance between the overall willingness of investors to buy or sell the corresponding asset in the market. The differentiation between number and volume based imbalance is essential which can be understood via the following example. Consider a scenario where in the last minute only one investor submitted a single buy order with the size of 100 shares and similarly, another investor submitted a single sell order with the size of 100 shares. This gives us a net of zero imbalance in terms of QO_1 AND NO_1. However, keeping the buy side the same, now consider that in the last minute 100 different investors submitted a sell order of size 1 share. While the QO_1 measure remains the same, NO_1 measure takes the value of -99, which might be an indicator of an overall displeasure of investors regarding the corresponding stock, in turn which might turn into a sell off in the upcoming minutes. Furthermore, since cancelled orders also directly effect the buy-sell imbalance, we introduce the features NCO_1 (number based) and QCO_3 (volume based) as well.[8]

On the other hand, the feature Vol_{O_1} reflects the instability in the buy-sell imbalance which can effect the consistency in the predictive power of the original features NO_1 and QO_1. Finally, the measures NT_1 and QT_1 reflect the imbalance in buyer-initiated trades against seller-initiated trades which can be an essential

[8]By the "number" (using the definitions of the analytics provided by the exchange), we refer to the number of orders with different identification numbers. Specifically, in the case of trades, "number" refers to the number of matched orders with different identification numbers which result in a trade. By "quantity", we refer to the number of shares for each order or trade. Hence quantity does not refer to monetary volume in our context.

predictor of the asset price. In particular, anyone who initiates a trade bears the minimum cost of the bid-ask spread, and therefore we might expect that the trade initiating side is at least partially informed.

While the abovementioned features focus on the raw difference in the buy- and sell-side of the market, rest of the features (NO_3, QO_3, NCO_3, QCO_3, Vol_{o_3}, NT_3, and QT_3) introduce a normalization factor. We introduce such a normalization concept at this stage since we believe that normalized features have the potential to reflect the buy-sell imbalance from a different perspective. For example, suppose that in a given minute, we observe 2 mn shares of buyer-initiated trades and 1 mn shares of seller-initiated trades. In such a scenario, our trade volume based imbalance measures take the value of $QT_1 = 1mn$ and $QT_3 = 1/3$. Now, consider another scenario where in a given minute we observe 2 shares of buyer-initiated trades and 1 share of seller-initiated trades. In this case, while the analytic QT_3 takes the same value with the previous case, the value of the QT_1 drastically drops to 1 from 1 mn.

In essence, many of our proposed features depend on some of the earlier and highly influential studies in the microstructure literature on the price impact of intraday trading activity. For example, using normalized version of the order imbalance measure that depends on both the number and the volume of trades (referring to features NT_3 and QT_3 respectively), Bailey et al. (2009) show that order imbalances can explain as much as 21.8% of the fluctuation in daily open-to-close returns in Chinese stocks. In another study, Chordia et al. (2002) show that raw order imbalances based on the volume and the number of trades (referring to features QO_1 and NO_1 respectively) has partial predictive power on returns within a day, and the authors suggest that price pressures caused by imbalances in inventory are an issue not just for individual stocks, but for the aggregate market as well in the US. Chan and Fong (2000) show that raw order imbalance can explain a substantial portion of daily price movements for Nasdaq stocks with a similar set of measures.

On the other hand, similar variables to our features (that focus on orders that are not realized as trades) such as NO_1, NO_3, QO_1, QO_3, NCO_1, NCO_3, QCO_1 and QCO_3 have been also used in some of the popular studies in the literature. Specifically, Amaya et al. (2018) show that imbalance in the unmatched orders between the bid versus ask side of the order book has considerable impact on intraday returns of US stocks. Similarly, Cont et al. (2014) show that, over short time intervals, price changes in US stocks are mainly driven by the order flow imbalance (OFI), defined as the imbalance between waiting supply and demand on the bid and ask side, respectively.

While many of our features have already been popular in the microstructure literature, some of them such as Vol_{O_1} and Vol_{O_3} have not appeared in earlier studies, and we believe that they would contribute to the

understanding on the forecasting of intraday price movements. In this study, since we focus on the imbalance between buy side and sell side from various aspects, we try to use all variables that (i) can be extracted from the analytics product and (ii) contain information regarding such imbalance. All in all, 14 features given in Table 4 are used in our ML methods. These features are either some of the analytics themselves or obtained from a simple combination of such analytics. Additional features that are not related to the analytics are not used because we attempt to determine whether investors can gain returns in excess of the benchmark index BIST 30 using just the analytics product.

4. Methodology

4.1. Machine learning models

We apply six different machine learning algorithms (k-Nearest Neighbors, Logistic Regression, Naive Bayes, Random Forest, Support Vector Machine, Extreme Gradient Boosting) to classify the target variable (it refers to the excess return of an individual stock against the benchmark index BIST30 in our study) as “up" or “down" at one-minute frequency. These methods are selected due their popularity and fast implementation, and they are performed with the Python’s well-known scikit-learn package. In what follows, we briefly describe how each of these classification algorithms helps to forecast the sign of the target variable.

4.1.1. k-Nearest Neighbors Classifier

k-nearest neighbors algorithm (kNN) is commonly used, simple yet successful classification method which has been applied in a large number of classification and regression problems such as handwritten digits and satellite image scenes. kNN is a supervised machine learning model where the model learns from the labeled data how to map the inputs to the desired output so that it can make predictions on test data. It is a non-parametric algorithm as it does not make any assumption about the data such as normality. kNN model picks an entry in the database and then looks at the ‘k’ entries in the database which are closest to the chosen point. Then, the data point is assigned the label of the majority of the ‘k’ closest points. For instance, if $k = 6$ with 4 of points being as ‘up’ and 2 as ‘down’, the data point in question would be labeled ‘up’, since ‘up’ is the majority class.

More generally, kNN algorithm works as follows: For a given value of k, it computes the distance between the test data and each row of the training data by using a distance metric like Euclidean metric (some of the other metrics that can also be used are cityblock, Chebychev, correlation, and cosine). The distance values are sorted in an ascending order and then top k elements are extracted from the sorted array.

It finds the most frequent class among these k elements and returns as the predicted class. In our application of kNN, we find the optimal value of k as 200.

4.1.2. Naive Bayes Classifier

Naive Bayes is another widely used classification algorithm as it is easy to build and particularly useful for very large data sets. This method is a supervised learning algorithm based on the application of the Bayes' theorem, and also called a probabilistic machine learning algorithm. It makes the "naive" assumption that the input features are conditionally independent of each other given the classification. If this assumption holds then naive Bayes classifier may perform even better than more complicated models. However, in real life, most of the time it is not possible to get a set of predictors which are completely independent.

The naive Bayes classifier assigns observations to the most probable class by first estimating the densities of the predictors within each class. As a second step, it computes the posterior probabilities according to Bayes' rule:

$$\widehat{P}(Y = k \mid X_1, ..., X_P) = \frac{\pi(Y = k) \prod_{j=1}^{P} P(X_j \mid Y = k)}{\sum_{k=1}^{K} \pi(Y = k) \prod_{j=1}^{P} P(X_j \mid Y = k)} \tag{1}$$

where Y is the random variable corresponding to the class index of an observation, $X_1, ..., X_P$ are the random predictors of an observation, and $\pi(Y = k)$ is the prior probability that a class index is k. Finally, it classifies an observation by estimating the posterior probability for each class, and then assigns the observation to the class yielding the maximum posterior probability. In this study, we implement the Gaussian Naive Bayes algorithm in which the likelihood of the features is assumed to be Gaussian.

4.1.3. Logistic Regression Classifier

Logistic Regression is a machine learning classification algorithm that is used to forecast the probability of a categorical dependent variable. In logistic regression, the outcome of target variable is dichotomous (i.e., there are only two possible classes). The classification algorithm forecasts the probability of occurrence of a binary event utilizing a logit function. More explicitly, logistic regression outputs a probability value by using the logistic sigmoid function and then this probability value is mapped to two discrete classes.

In our case, we have a binary classification problem of identifying the next time excess return as up or down. Logistic regression assigns probabilities to each row of the features matrix X. Let us denote the

sample size of the dataset with N and thus we have N rows of the input vector. Given the set of d features, i.e. $x = (x_1, ..., x_d)$, and parameter vector w, the logistic regression with the penalty term minimizes the following optimization problem:

$$\min_{w,c} \frac{w^T w}{2} + C \sum_{i=1}^{N} \log(\exp(-y_i(x_i^T w + c)) + 1) \tag{2}$$

where we find the optimal value of C by making a grid search over a set of reasonable values for C. In our empirical testing, we use the default parameters for LogisticRegression function from Python's scikit-learn package together with setting and max_iter = 10000.

4.1.4. Random Forest Classifier

Random Forest Classifier is an ensemble algorithm such that it combines more than one algorithm of the same or different kind for classifying objects. Decision trees are the building blocks of the random forest model. In other words, random forest consists of a large number of individual decision trees that function as an ensemble. Random forest classifier creates a set of decision trees from randomly selected subset of training set and each individual tree makes a class prediction. It then sums the votes from different decision trees to decide the final class of the test object. For instance, assume that there are 5 points in our training set that is $(x_1, x_2, ..., x_5)$ with corresponding labels $(y_1, y_2, ..., y_5)$ then random forest may create four decision trees taking the input of subset such as (x_1, x_2, x_3, x_4), (x_1, x_2, x_3, x_5), (x_1, x_2, x_4, x_5), (x_2, x_3, x_4, x_5). If three of the decision trees vote for "up" against "down" then random forest predicts "up". This works efficiently because a single decision tree may be producing noise but a large number of relatively uncorrelated trees operating as a choir will reduce the effect of noise which would result in more accurate results.

More generally, in random forest method as proposed by Breiman (2001), a random vector θ_k is generated, independent of the past random vectors $\theta_1, ..., \theta_{k-1}$ but with the same distribution; and a tree is grown using the training set and θ_k resulting in a classifier $h(x, \theta_k)$ where x is an input vector. In random selection, θ consists of a number of independent random integers between 1 and K. The nature and dimension of θ depend on its use in tree construction. After a large number of trees are generated, they vote for the most popular class. This procedure is called as the random forest. In this work, we apply the random forest method by using the RandomForestClassifier function from Python's scikit-learn package. We set max_leaf_nodes to 16 and all the other parameters are chosen as the default values.

4.1.5. Support Vector Machine Classifier

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both regression and classification tasks. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space where N is the number of features which distinctly classifies the data points. Hyperplanes can be thought as decision boundaries which classify the data points. Data points falling on different sides of the hyperplane can be assigned to different classes. Support vectors are described as the data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. The margin of the classifier is maximized using these support vectors. In more technical terms, the above process can be summarized as follows. Given the training vectors x_i for $i = 1, 2, ..., N$ with a sample size of N observations, the support vector machine classification algorithm solves the following problem given by

$$\min_{w,h,\xi} \frac{w^T w}{2} + C \sum_{i=1}^{N} \xi_i \tag{3}$$

subject to $y_i(w^T \phi(x_i)) \geq 1 - \xi_i$ and $\xi_i \geq 0, i = 1, 2, ..., N$. The dual of the above problem is given by

$$\min_{\alpha} \frac{\alpha^T Q \alpha}{2} - e^T \alpha \tag{4}$$

subject to $y^T \alpha = 0$ and $0 \leq \alpha_i \leq C$ for $i = 1, 2, ..., N$, where e is the vector of all ones, $C > 0$ is the upper bound. Q is an n by n positive semi-definite matrix. $Q_{ij} = y_i y_j K(x_i, x_j)$, where $K(x_i, x_j) = \phi(x_i)^T \phi(x)$ is the kernel. Here training vectors are implicitly mapped into higher dimensional space by the function ϕ. The decision function in the support vector machines classification is given by

$$sign\left(\sum_{i=1}^{N} y_i \alpha_i K(x_i, x) + \rho\right). \tag{5}$$

The optimization problem in Equation 3 can be solved globally using the Karush-Kuhn-Tucker (KKT) conditions. Clearly, this optimization problem depends on the choice of the Kernel functions. Our study employs the Gaussian (rbf) kernel which is denoted by $\exp(-\gamma \|x - x'\|^2)$ where γ must be greater than 0. When SVM is implemented, by using a grid search we find the optimal values of C and γ as 0.09 and 0.03, respectively.

4.1.6. Extreme Gradient Boosting Classifier

Extreme Gradient Boosting (XGBoost) is a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework. As we said before, an ensemble method is a machine learning

technique that combines several base models in order to produce one optimal predictive model. An algorithm is called boosting if it works by adding models on top of each other iteratively, the errors of the previous model are corrected by the next predictor, until the training data is accurately predicted or reproduced by the model. A method is called gradient boosting if, instead of assigning different weights to the classifiers after every iteration, it fits the new model to new residuals of the previous prediction and then minimizes the loss when adding the latest prediction. Namely, if a model is updated using gradient decent, then it is called gradient boosting. XGBoost improves upon the base gradient boosting framework through systems optimization and algorithmic enhancements. Some of these enhancements can be listed as parallelized tree building, tree pruning using depth-first approach, cache awareness and out-of-core computing, regularization for avoiding overfitting, efficient handling of missing data, and in-built cross validation capability. For further technical details of the method, we refer the reader to Chen and Guestrin (2016). In our implementation of XGBoost we use the default parameters from XGBoost function in Python sklearn package with the exception that we set the parameter alpha which is L1 regularization term on weights to 15 and we choose lambda which is the L2 regularization term on weights as 10.

4.2. Calculating the prediction success

Assume that the real label of the target variable is denoted by Y and predicted label is denoted by Y' then we employ the following measure to assess the usefulness of our selected forecasting techniques:

Sign Prediction Ratio (SPR): Correctly predicted excess return direction is assigned 1 and 0 otherwise, then sign prediction ratio is calculated by

$$SPR = \frac{\sum_{j=1}^{M} matches(Y_j, Y_j')}{M}, \tag{6}$$

where

$$matches(Y_j, Y_j') = \begin{cases} 1, & if\ Y_j = Y_j', \\ 0, & otherwise. \end{cases} \tag{7}$$

and M denotes the size of the set for which the sign prediction ratio is measured.

In addition, the area under the curve (AUC) and the receiver operating characteristic (ROC) curves are presented for all the classifiers; see Figure 1 and Table 17.

4.3. Testing with a trading strategy

We test our machine learning algorithms with a trading rule such that if the predicted sign is positive then we long the related asset and if the predicted direction is down then we short the asset. We measure the

performance of our trading rule by using the following measures:

The **Maximum Return** is obtained by adding absolute value of all the excess returns (denoted by h)

$$MaxReturn = \sum_{j=1}^{M} abs(h_j) \tag{8}$$

and represents the maximum achievable return assuming the perfect forecast.

The **Total Return** is computed in the following way

$$\text{TotalReturn} = \sum_{j=1}^{M} sign(Y_j') * h_j \tag{9}$$

where $sign$ is the standard sign function and $*$ denotes the usual multiplication. Notice that the better the prediction method, the larger the total return is.

Ideal Profit Ratio is the ratio of the total return in Eq.(9) and the maximum return in Eq.(8).

$$IPR = \frac{TotalReturn}{MaxReturn} \tag{10}$$

4.4. One-sided t-test to measure the model success

In both short and long term analysis parts of empirical investigation, we test whether the average of the accuracy ratios generated by sliding windows is significantly greater than 50%. For this purpose, we conduct one-sided t-test with the following hypothesis:

H_0: The data comes from a population with a mean equal to 50%,

H_1: The data comes from a population with a mean greater than 50%.

4.5. Cross-validation

The simplest way of estimating the prediction error is cross-validation. The popular k-fold cross-validation is particularly appropriate for the case for the situation where data is scarce. The procedure consists by splitting the data into k equal parts and training the classifier on $k-1$ parts and using the k-th part for calculating the prediction error.

The k-fold cross-validation procedure has to be used with much care when working with time-series because of the serial correlation in the data. Although there are suggestions of how to avoid the effects of serial correlation in the literature, our data set is large. This means that any classifier is not going to be trained on the whole data set. In fact, we see that the performance of the classifier does not continue to improve as we increase the training period.

We do cross validation by dividing the data into a sequence of train-test periods. We specify the train-test split using the train-test-shift notation. For example, we use 5 contiguous days for training and immediately following 2 contiguous days for validation; afterwards, train and test windows are shifted by one day. We report the results for multiple train-test-shift combinations.

4.6. Data processing and implementation

The analytics data is only provided at the one-minute level. To train the classifiers, we needed to aggregate the price data to the same temporal resolution. This consists of considering only the close price of the interval. Furthermore, due to opening hours effects of the exchange, the first and last 15 minutes of every working shift were not considered.

Financial data generally shows a low signal-to-noise ratio. For that reason the training set had to be balanced for the training to avoid the classifier just predicting the majority class. The training set was balanced by oversampling the minority class, i.e. random observations from the minority class were doubled for the training sample.

5. Empirical Results

As stated earlier, we sample our data at every one-minute time interval and then apply six different classification algorithms namely kNN, logistic regression, naive Bayes, random forest, SVM, XGBoost which are described in the previous section. The target variable for all of these classification methods is the sign of the excess return of an individual stock against the benchmark index BIST30 at one minute frequency. The down and up ratios computed from the sign of one-minute log-returns are given in Table 2. As it is clear from this table, we have a balanced data set which helps to increase the accuracy of our machine learning algorithms. We conduct both short- and long-term analysis by applying the above-mentioned methodologies with short- and long- period of sliding windows, respectively. In order to catch the short-term dynamics of the assets, we employ three different sliding windows: 2-days train, 1-day test, 1-day shift; 5-days train, 2-days test, 1-day shift; 20-days train, 5-days test, 1-day shift in the short-term case. For each sliding window, we consider a shifted starting point for the training and out-of-sample backtesting. For the period from 1 June 2016 to 14 November 2016, we consider order and trade flow in the time interval from 9:45 am to 12:50 pm in the morning session and from 14:10 pm to 17:20 pm in the afternoon session. These together yield 375 one-minute intervals in a day. The opening and closing times of the stock exchange changed on 14 November 2016. Starting from this date till 31 December 2018, we use order and trade flow in the

time interval from 10:10 am to 12:50 pm in the morning session and from 14:10 pm to 17:50 pm in the afternoon session which in total make 380 one-minute intervals in a day. Hence, for 5-days train and 2-days test window, we have approximately 1900 (5x380) one-minute time intervals in the train set and 760 (2x380) one-minute time intervals in the test set. On the other hand, in order to analyse the long term dynamics of the stocks, we utilize three different sliding windows: 120-day train, 30-day test, 150-day shift; 160-day train, 40-day test, 200-day shift; 200-day train, 50-day test, 250-day shift. The summary of the list of the train and test set divisions are given in Table 3.

As we described in the previous section, we present two key metrics to measure the performance of the different machine learning algorithms. We first use the sign prediction rate or the accuracy rate which is calculated as the proportion of times the related methodology correctly predicts the direction of the next time step excess return. It is known that, if the underlying process were fully random, the correct sign prediction ratio would then be 50%. However, in our case, it is important to note that we are using signs of excess returns against the benchmark index. Hence, any accuracy rate greater than 50% already indicates the success of the algorithm to beat the market. In addition to sign prediction ratio, we also apply ideal profit ratio to measure the performance of the related classification algorithm. As it is formalized in Section 4, the ideal profit ratio is the ratio of the return generated by a given algorithm to the perfect sign forecast. In the following subsections, we present the results for the long- and short-term analysis.

5.1. The long-term analysis

The columns in Table 5-10 show the success ratios for in-sample (train) and out-sample (test) periods as well as ideal-profit ratios (for out-sample period) under three different sliding windows for each stock. Mean value, standard deviation, maximum, and minimum of each column across the stocks are given below the tables. In Table 5, the accuracy results obtained from the naive Bayes classification algorithm are presented. First of all, we observe that both in-sample and out-of-sample average accuracy rates across the stocks are 52% and it remains unchanged along the different specifications of sliding windows. Similarly, average ideal profit ratios are around 8% and they do not change across different sliding windows. The poor performance of the model may result from the main assumption of the method that the features are independent from each other. This assumption may not be easy to satisfy in practice. Although it is the worst performing methodology among all, it still provides a minimum of 50% success rate for all the sliding windows.

Table 5 also provides results for the kNN algorithm which are obtained by optimizing over neighborhood numbers from 1 to 20. kNN methodology yields average out-of-sample (in-sample) success ratio of 55%

(53%) for the first sliding window, 55% (54%) for the second and third sliding windows. Average ideal profit ratios are around 13% for different windows. It is interesting to observe that although in-sample accuracy rates for kNN are higher than all models except XGBoost algorithm, the out-of-sample success rates and ideal profit ratios are only slightly better than the random forest model. The maximum value of out-of-sample success rates (56%) is attained by the same stock (EKGYO) under three different windows.

The in-sample and out-of-sample fits given in Table 7 reveal that although random forest algorithm is computationally more expensive than kNN algorithm, it does not provide any better results compared to kNN method. Similar to kNN, the highest success rate of (56%) is obtained with the same stock (EKGYO) for three different parametrization of windows.

Insert Tables 5 & 6 & 7 about here

The results for the logistic regression classification algorithm are presented in Table 8. We observe that, as a linear classification algorithm, this method provides quite robust average accuracy rates for different sliding windows. All the average in-sample and out-of-sample success rates are statistically and significantly greater than 50% with a persistent value of 54%. The same observation is actually true for the maximum (56%) and minimum (53%) values across different train-test sets, except that maximum value of 57% for the third set.

When we apply a nonlinear classification method such as support vector machine by choosing radial kernel, Table 9 shows that this method does not improve the results compared to logistic regression at all. The best performance is obtained for EKGYO (57%) in the third partition of the data set. However, the support vector machine yields slightly worse ideal profit ratios compared to the logistic regression. It is evident from the results given in Table 10, the in-sample and out-of-sample fits of the XGBoost algorithm are the highest among all the machine learning algorithms considered. The highest average in-sample success rate can reach up to 58% for the first train-test set. However, the out-of-sample average performance is comparatively lower than the in-sample fits. This designates the high variance in the XGBoost classification with high in-sample fit to the noisy data but lower out-of-sample performance. This method also has the highest average ideal profit ratios compared to the other algorithms. The highest value of ideal profit ratio (33%) indicates that XGBoost is not only successful for the prediction of the correct sign of the excess returns but also for the size of excess returns.

Insert Tables 8 & 9 & 10 about here

5.2. Short-term analysis

As stated earlier, we consider three different sliding windows in order to observe short-term dynamic effects of features on the excess return prediction. We have around 17450 sliding windows for each of the 2-day train, 1-day test, 1- day shift (triplet 1); 5-day train, 2-day test, 1- day shift (triplet 2); 20-day train, 5-day test, 1- day shift (triplet 3). For each of these windows, we apply the related machine learning algorithm and then take the average of accuracy rates and ideal profit ratios across the sliding windows. The numbers in the parenthesis next to the out-of-sample ratios in Tables 12 and 13 represent the t-statistics coming from the one-sided t-test which was described in Section 4. All of the average success ratios coming from the sliding windows are statistically and significantly greater than 0.5 for each of the (train, test, shift) triplets across six different algorithms.

More precisely, we observe that, similar to the long-term analysis, naive Bayes results presented in Table 11 yield the lowest success values on average (52%) for the out-of-sample ratios among all the methods for different triplets. As we stated previously, this may be due to the fact that the independence of the features is a difficult assumption to satisfy in practice. Table 12 provides the results for the kNN algorithm. Clearly, the results are very close to naive Bayes except that the average out-of-sample accuracy rates for the triplet 2 (53%) and triplet 3 (53%) differ by 1% from the naive Bayes algorithm. It is important to note that, as we expand the training window, the average success rate increases for the kNN algorithm. Actually, this is also the case for the other methods except the naive Bayes for which the success ratio remains constant across the expanding training window. We also have similar observations for the ideal profit ratios. Table 13 shows that the random forest algorithm produces exactly the same out-of-sample fits with kNN method. However, in-sample fits and ideal profit ratios are slightly better for the random forest.

Insert Tables 11 & 12 & 13 about here

The next best performing method is the XGBoost algorithm for which the results are presented in Table 14. As it is the case for the long term analysis, we observe high in-sample fits for the XGBoost classifier (60% for the triplet 1 and 3, 61% for the triples 2). However, the out-of-sample success rates (53% for triplet 1 and 2, 54% for triplet 3) are significantly lower compared to the in-sample counterparts. This can

be result of the fact that XGBoost is an algorithm with many parameters to calibrate. Hence even small fine tuning in one of the parameters may result in amplified changes in the short term for the other parameters in terms of the output rates. Ideal profit ratios for XGBoost increase from 10% to 15% with the expanding training window. Table 15 shows the results for the support vector machine classification which has exactly the same average accuracy rates as the XGBoost algorithm. However, in-sample fits and ideal profit ratios are lower compared to XGboost. Finally, we observe the highest out-of-sample forecast success rates for the logistic regression for which results are demonstrated in Table 16. Similarly, the average ideal profit ratios can reach up to 16 % for the third triplet. Notably, a linear model like logistic regression with a lower level of computation cost and time performs as good as a nonlinear model such as XGBoost with a higher computational work and time.

Insert Tables 14 & 15 & 16 about here

5.3. Factor importance

To perform a factor importance analysis and have it be representative for different methods, we need to choose a factor importance analysis that is method agnostic. There are a number of techniques that fit this description. The simplest example would be based on subset selection techniques; this, however, requires retraining of the estimators. We apply the permutation importance analysis of our classifiers.

The permutation importance technique compares the performance of the classifier with the performance of the classifier with one of the factor values permuted; the more the performance deteriorates, the more important the factor is. This last point is important because permutation of the factor preserves the distribution of the values of that factor. Moreover, it does not require retraining of the classifier.

Figure 2 shows the results of the factor importance analysis. The plot was produced as follows. We first used the classifiers for the 5/2 sliding window analysis. Therefore, 5 days of data were used for training the classifier. Then, the permutation importance analysis was performed in sample and out of sample. This procedure gives us two numbers (average performance loss over a number of permutations). To obtain information about how the factor importance varies in time, we perform the same analysis on the shifted windows. The box plot of that sample is reported on Figure 2. To test for the case that different factors as time goes on, we analyzed the factor importance data as a function of the starting date of the window in which we analyze it, but the data showed no trends. In particular, the weighing of factors by importance remains stable over the analysis period.

Insert Figure 2 about here

The factor importance analysis shows that for all methods, except for the Random Forest classifier, the factors NT_3 and QT_3 are most important. The factor importance plots also show quite a high variance of the importance weights on those two factors. This is easily explained by the much higher correlation between those two factors as compared to any other factor pair. Furthermore, for all the classifiers, with the exception of the Random Forest classifier, the permutation factor importance in-sample and out-of-sample coincide very well, which indicates that our classification method does not suffer from overfitting. In the case of the Random Forest classifier, higher weights are given to all of the factors in-sample as compared to out-of-sample factor importance analysis. This is consistent with a quite a poor performance of the Random Forest classifier out-of-sample, though it does not strongly point to overfitting.

The reported results in Figure 2 are given for EKGYO as well as the in-sample period of 5 days and out-of-sample period of 2 days. The results of our factor importance analysis do not, however, depend on the train and test periods. That is, the results are representative for all the tested combinations of train/test periods.

6. Conclusion

The vast amount of data generated everyday and improvement trend in technological developments continue to shape the way finance industry operates. These developments provide opportunities for the collection, processing, and eventually analysis of big data for various purposes, including especially visualization, estimation and forecasting. As a result, many market participants in financial markets all around the world could use big data analytics to make better investment decisions and obtain consistent excess returns. For their parts, algorithmic traders in particular can use various market analytics with advanced quantitative methods to maximize their portfolio returns. At the firm level, big data analytics can help businesses improve their decision-making, productivity, business forecasting, accountability and competitive advantages. However, analytical methods as well as data support and guidance lacking are posing serious challenges for informed investors and firms. One of the most appealing steps thus consists of understanding the value of the big data analytics.

In this study, we examined whether the real-time data analytics product, provided to customers by Borsa Istanbul, can be used successfully in predicting intraday excess returns over the benchmark index with the

help of various machine learning (ML) algorithms. Our results reveal that, even using only these analytics as the main features of the ML methods can help investors to forecast one minute ahead excess returns. In our both short- and long-term analyses and for different selection of train and test partitions, we observe prediction accuracy rates that are significantly larger than 50% across all stocks and ML methods, which can reach up to above 57% for specific stocks and methodologies. The results are not different when we consider ideal profit ratios because the ML techniques can capture up to 33% of the all potential excess returns. In the case of forecasting the direction, XGBoost algorithm (logistic regression) provides slightly superior performance compared to competing models in the long-term analysis (short-term analysis) which is also the case when we examine the ideal profit ratios. Finally, a general observation is that as the train period covers a larger sample, ML models become more successful as expected.

Our findings open up several directions for future research. First, without referring to advanced forecasting techniques such as ML and variants, an interesting question arises as to whether the combination of analytics with simple models such as the classical ARIMA could be as successful as the modern approaches used in this study. This comparison is particularly important for those investors without the technical capacity to implement state of the art forecasting techniques. Second, while the analysis in this study is performed using purely the data analytics provided by the exchange, one may question the predictive power of the analytics if they are used in addition to the classical technical trading features such as short- and long-term momentum, moving averages, relative strength indicator; or recently popular indicators such as internet search, media mentions, and even social and responsible investment scores. This would naturally help us evaluate further the value added by the data analytics product. Third, the availability of the data analytics also enables an investigation on the possible improvement of market efficiency in terms of lower return predictability. The answer to this question leads to a better understanding of the role of real-time public data on the pricing efficiency of equity markets.

Table 1: This table reports descriptive statistics for the raw analytics and the features constructed from these analytics. The explanations for the features are provided in Table 4.

Panel A

cross-sectional averages	Number of arrived buy orders	Quantity of arrived buy orders	Number of arrived sell orders	Quantity of arrived sell orders	Number of cancelled buy orders	Quantity of cancelled buy orders	Number of cancelled sell orders	Quantity of cancelled sell orders	Number of buyer-initiated trades	Quantity of buyer-initiated trades	Number of seller-initiated trades	Quantity of seller-initiated trades
mean per day	14.76	69418.77	13.77	71717.45	5.80	27966.07	6.17	29102.01	7.62	21000.74	7.09	18900.26
std per day	13.94	126360.72	13.29	119187.34	6.92	60245.05	7.15	55273.89	10.97	51670.33	11.13	44518.18

Panel B

cross-sectional averages	NT_1	NT_3	QT_1	QT_3	NO_1	NO_3	QO_1	QO_3	NCO_1	NCO_3	QCO_1	QCO_3	Vol_O_1	Vol_O_3
mean per day	0.53	0.19	2100.48	0.16	0.99	0.04	-2298.68	-0.02	-0.37	0.07	-1135.94	0.05	-1135.94	0.05
std per day	14.00	0.71	59781.19	0.81	12.99	0.45	133782.36	0.60	7.91	0.62	75879.62	0.71	75879.62	0.71

Note: Values in the table are the average unitless numbers per minute and per stock. For instance, for a given stock, on a given day, we have the corresponding analytic for each minute. We take the average of these minutely analytic values for each day, then take the average across all the days for the given stock. Finally, we take the mean of these daily averages across all sample stocks.

Table 2: This table reports down and up ratios computed from the sign of one-minute log-returns

	AKBNK	ARCLK	BIMAS	CCOLA	EKGYO	ENKAI	EREGL	FROTO	GARAN	HALKB	ISCTR	KCHOL	KOZAL	KRDMD
down	0.506	0.505	0.507	0.510	0.509	0.506	0.506	0.506	0.507	0.506	0.507	0.505	0.509	0.507
up	0.494	0.495	0.493	0.490	0.491	0.494	0.494	0.494	0.493	0.494	0.493	0.495	0.491	0.493

	OTKAR	PETKM	SAHOL	SISE	TAVHL	TCELL	THYAO	TKFEN	TOASO	TTKOM	TUPRS	ULKER	VAKBN	YKBNK
down	0.510	0.508	0.508	0.510	0.506	0.507	0.508	0.508	0.507	0.508	0.505	0.510	0.509	0.510
up	0.490	0.492	0.492	0.490	0.494	0.493	0.492	0.492	0.493	0.492	0.495	0.490	0.491	0.490

Table 3: This table reports train and test set divisions for the short- and long-term analysis

short-term analysis			long-term analysis		
train	test	shift	train	test	shift
2	1	1	120	30	150
5	2	1	160	40	200
20	5	1	200	50	250

Table 4: This table presents features used in the machine learning algorithms together with their explanations. Definitions for the raw data analytics are given in Table A.2

Feature	**Explanation**
$NO_1 = A5 - A6$	Number of buy orders - sell orders arriving per 60 seconds
$NO_3 = (A5 - A6)/(A5 + A6)$	$\frac{\text{Number of buy orders - sell orders arriving in the last 60 seconds}}{\text{Number of buy orders + sell orders arriving in the last 60 seconds}}$
$QO_1 = A7 - A8$	Quantity of buy orders - sell orders arriving in the last 60 seconds
$QO_3 = (A7 - A8)/(A7 + A8)$	$\frac{\text{Quantity of buy orders - sell orders arriving in the last 60 seconds}}{\text{Quantity of buy orders + sell orders arriving in the last 60 seconds}}$
$NCO_1 = A12 - A13$	Number of cancelled buy orders - cancelled sell orders in the last 60 seconds
$NCO_3 = (A12 - A13)/(A12 + A13)$	$\frac{\text{Number of cancelled buy orders - sell orders in the last 60 seconds}}{\text{Number of cancelled buy orders + sell orders arriving in the last 60 seconds}}$
$QCO_1 = A14 - A15$	Quantity of cancelled buy orders - cancelled sell orders in the last 60 seconds
$QCO_3 = (A14 - A15)/(A14 + A15)$	$\frac{\text{Quantity of cancelled buy orders - sell orders in the last 60 seconds}}{\text{Quantity of cancelled buy orders + sell orders arriving in the last 60 seconds}}$
$Vol_{O_1} = A26 - A27$	Volatility of buy order quantity - Volatility of sell order quantity in the last 5 minutes
$Vol_{O_3} = (A26 - A27)/(A26 + A27)$	$\frac{\text{Volatility of buy order quantity - Volatility of sell order quantity in the last 5 minutes}}{\text{Volatility of buy order quantity + Volatility of sell order quantity in the last 5 minutes}}$
$NT_1 = A32 - A33$	Number of buyer initiated trades - seller initiated trades in the last 60 seconds
$NT_3 = (A32 - A33)/(A32 + A33)$	$\frac{\text{Number of buyer initiated trades - seller initiated trades in the last 60 seconds}}{\text{Number of buyer initiated trades + seller initiated trades in the last 60 seconds}}$
$QT_1 = A34 - A35$	Quantity of buyer initiated trades - seller initiated trades in the last 60 seconds
$QT_3 = (A34 - A35)/(A34 + A35)$	$\frac{\text{Quantity of buyer initiated trades - seller initiated trades in the last 60 seconds}}{\text{Quantity of buyer initiated trades + seller initiated trades in the last 60 seconds}}$

Table 5: This table presents Naive Bayes classification in-sample and out-of-sample accuracy results and ideal profit ratios for different train/test combinations for the static analysis.

	120-day train, 30-day test, 150-day shift			160-day train, 40-day test, 200-day shift			200-day train, 50-day test, 250-day shift		
	success ratio		ideal profit ratio	success ratio		ideal profit ratio	success ratio		ideal profit ratio
	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample
AKBNK	0.53	0.52*** (4.29)	0.09	0.54	0.53*** (3.22)	0.11	0.54	0.54*** (63.96)	0.12
ARCLK	0.52	0.52*** (3.18)	0.07	0.52	0.52*** (2.08)	0.06	0.52	0.52*** (1.26)	0.05
BIMAS	0.51	0.5*** (0.36)	0.02	0.52	0.51*** (1.23)	0.04	0.50	0.5*** (-0.04)	0.01
CCOLA	0.51	0.51*** (1.39)	0.02	0.50	0.5*** (0.34)	0.00	0.50	0.5*** (0.36)	0.01
EKGYO	0.55	0.55*** (11.17)	0.24	0.55	0.55*** (12.6)	0.27	0.55	0.56*** (107.1)	0.27
ENKAI	0.53	0.53*** (2.48)	0.13	0.54	0.53*** (6.83)	0.14	0.54	0.54*** (10.01)	0.19
EREGL	0.53	0.52*** (4.11)	0.07	0.53	0.52*** (4.61)	0.08	0.53	0.53*** (8.93)	0.10
FROTO	0.51	0.51*** (2.72)	0.04	0.52	0.52*** (2.35)	0.07	0.52	0.51*** (1.43)	0.06
GARAN	0.53	0.52*** (4.36)	0.07	0.53	0.53*** (2.76)	0.08	0.52	0.52*** (9.78)	0.06
HALKB	0.52	0.52*** (1.94)	0.06	0.52	0.51*** (2)	0.03	0.52	0.51*** (0.78)	0.02
ISCTR	0.54	0.53*** (11.23)	0.12	0.54	0.53*** (4.66)	0.12	0.54	0.54*** (149.22)	0.14
KCHOL	0.51	0.51*** (0.94)	0.03	0.51	0.51*** (1.59)	0.03	0.51	0.51*** (1.19)	0.04
KOZAL	0.52	0.51*** (1.67)	0.04	0.52	0.52*** (2.17)	0.07	0.52	0.51*** (1.45)	0.04
KRDMD	0.52	0.51*** (2)	0.07	0.53	0.53*** (1.86)	0.13	0.53	0.52*** (0.87)	0.12
OTKAR	0.52	0.52*** (7.95)	0.07	0.52	0.51*** (1.93)	0.04	0.52	0.53*** (3.19)	0.09
PETKM	0.54	0.54*** (4.93)	0.16	0.54	0.53*** (2.77)	0.13	0.55	0.54*** (4.33)	0.14
SAHOL	0.53	0.53*** (6)	0.11	0.53	0.53*** (8.53)	0.12	0.53	0.54*** (32.12)	0.13
SISE	0.54	0.54*** (17.38)	0.18	0.54	0.54*** (8.74)	0.18	0.54	0.54*** (4.67)	0.18
TAVHL	0.50	0.5*** (1.1)	0.02	0.51	0.51*** (0.53)	0.02	0.50	0.5*** (-0.26)	0.00
TCELL	0.50	0.5*** (0.5)	0.02	0.51	0.51*** (1.14)	0.04	0.50	0.5*** (-0.3)	0.00
THYAO	0.52	0.52*** (4.8)	0.06	0.52	0.52*** (7.61)	0.06	0.53	0.53*** (2.77)	0.07
TKFEN	0.52	0.52*** (1.5)	0.06	0.51	0.51*** (0.61)	0.03	0.51	0.5*** (0.11)	0.02
TOASO	0.51	0.51*** (1.94)	0.04	0.52	0.51*** (2.93)	0.05	0.51	0.51*** (58.2)	0.02
TTKOM	0.52	0.52*** (2.67)	0.10	0.51	0.51*** (1.42)	0.07	0.52	0.51*** (0.62)	0.08
TUPRS	0.52	0.51*** (2.3)	0.05	0.51	0.5*** (0.68)	0.00	0.50	0.5*** (-0.62)	0.00
ULKER	0.51	0.51*** (1.61)	0.04	0.51	0.51*** (1.08)	0.03	0.50	0.5*** (-0.3)	0.00
VAKBN	0.52	0.52*** (1.32)	0.08	0.53	0.53*** (1.76)	0.11	0.52	0.52*** (0.89)	0.08
YKBNK	0.54	0.54*** (7.14)	0.16	0.54	0.54*** (7.11)	0.15	0.54	0.54*** (27.77)	0.16
Mean	0.52	0.52	0.08	0.52	0.52	0.08	0.52	0.52	0.08
Std	0.01	0.01	0.05	0.01	0.01	0.06	0.02	0.02	0.07
Max	0.55	0.55	0.24	0.55	0.55	0.27	0.55	0.56	0.27
Min	0.50	0.50	0.02	0.50	0.50	0.00	0.50	0.50	0.00

Note: The values in the parentheses are t-statistics for the one-sided t-test. ***, ** and * denote significant at the 1%, 5% and 10% level respectively.

Table 6: This table presents k-Nearest Neighbors classification in-sample and out-of-sample accuracy results and ideal profit ratios for different train/test combinations for the static analysis.

	120-day train, 30-day test, 150-day shift			160-day train, 40-day test, 200-day shift			200-day train, 50-day test, 250-day shift		
	success ratio		ideal profit ratio	success ratio		ideal profit ratio	success ratio		ideal profit ratio
	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample
AKBNK	0.55	0.53*** (9.95)	0.11	0.55	0.54*** (4.83)	0.12	0.55	0.53*** (20.75)	0.11
ARCLK	0.55	0.53*** (4.16)	0.10	0.55	0.53*** (3.9)	0.11	0.55	0.54*** (33.33)	0.13
BIMAS	0.55	0.53*** (37.13)	0.11	0.55	0.53*** (6.87)	0.12	0.55	0.53*** (34.32)	0.13
CCOLA	0.55	0.53*** (13.29)	0.11	0.54	0.53*** (15.43)	0.11	0.55	0.53*** (94.45)	0.11
EKGYO	0.56	0.56*** (7.37)	0.27	0.56	0.56*** (16.21)	0.30	0.56	0.56*** (30.09)	0.29
ENKAI	0.55	0.54*** (7.84)	0.19	0.55	0.53*** (6.76)	0.17	0.55	0.55*** (31.33)	0.21
EREGL	0.56	0.53*** (3.92)	0.13	0.55	0.53*** (2.96)	0.11	0.56	0.54*** (2.93)	0.15
FROTO	0.55	0.53*** (6.44)	0.12	0.55	0.53*** (3.97)	0.11	0.55	0.53*** (11.58)	0.10
GARAN	0.55	0.53*** (4.47)	0.11	0.55	0.53*** (4.5)	0.10	0.55	0.53*** (7.44)	0.10
HALKB	0.55	0.53*** (5.01)	0.10	0.55	0.53*** (3.92)	0.07	0.55	0.53*** (11.02)	0.07
ISCTR	0.55	0.53*** (8.39)	0.13	0.55	0.54*** (5.93)	0.12	0.55	0.54*** (22.82)	0.15
KCHOL	0.55	0.52*** (10.06)	0.08	0.55	0.53*** (7.86)	0.08	0.54	0.53*** (16.48)	0.09
KOZAL	0.55	0.53*** (11.32)	0.08	0.55	0.53*** (11.54)	0.10	0.55	0.52*** (2.42)	0.09
KRDMD	0.56	0.54*** (3.93)	0.22	0.56	0.54*** (3.17)	0.19	0.56	0.55*** (3.9)	0.22
OTKAR	0.55	0.53*** (11.33)	0.12	0.55	0.54*** (15.78)	0.11	0.55	0.53*** (5.9)	0.12
PETKM	0.56	0.53*** (3.03)	0.15	0.56	0.53*** (2.57)	0.14	0.56	0.54*** (3.03)	0.14
SAHOL	0.55	0.54*** (8.61)	0.13	0.55	0.54*** (15.07)	0.13	0.55	0.54*** (9.01)	0.14
SISE	0.56	0.55*** (15.99)	0.21	0.56	0.55*** (22.97)	0.22	0.56	0.55*** (11.7)	0.22
TAVHL	0.55	0.54*** (19.79)	0.13	0.55	0.54*** (26.88)	0.10	0.55	0.53*** (3.45)	0.11
TCELL	0.55	0.53*** (11.5)	0.10	0.55	0.53*** (9.42)	0.10	0.55	0.53*** (19.95)	0.09
THYAO	0.54	0.52*** (3.79)	0.07	0.54	0.52*** (2.98)	0.07	0.54	0.52*** (2.39)	0.09
TKFEN	0.55	0.53*** (6.67)	0.13	0.55	0.54*** (7.11)	0.13	0.55	0.53*** (4.44)	0.10
TOASO	0.55	0.53*** (7.57)	0.13	0.55	0.53*** (10.1)	0.12	0.55	0.53*** (14.17)	0.11
TTKOM	0.55	0.53*** (13.44)	0.15	0.55	0.54*** (8.42)	0.15	0.55	0.54*** (8.04)	0.16
TUPRS	0.55	0.53*** (12.79)	0.11	0.55	0.53*** (6.94)	0.09	0.55	0.54*** (34.61)	0.12
ULKER	0.55	0.53*** (16.03)	0.12	0.54	0.53*** (81)	0.11	0.55	0.52*** (27)	0.10
VAKBN	0.55	0.53*** (7.28)	0.15	0.55	0.54*** (7.46)	0.15	0.55	0.54*** (17.87)	0.17
YKBNK	0.56	0.54*** (8.56)	0.17	0.56	0.55*** (6.49)	0.18	0.56	0.55*** (177.8)	0.19
Mean	0.55	0.53	0.13	0.55	0.54	0.13	0.55	0.54	0.14
Std	0.00	0.01	0.04	0.01	0.01	0.05	0.01	0.01	0.05
Max	0.56	0.56	0.27	0.56	0.56	0.30	0.56	0.56	0.29
Min	0.54	0.52	0.07	0.54	0.52	0.07	0.54	0.52	0.07

Note: The values in the parentheses are t-statistics for the one-sided t-test. ***, ** and * denote significant at the 1%, 5% and 10% level respectively.

Table 7: This table presents Random forest classification in-sample and out-of-sample accuracy results and ideal profit ratios for different train/test combinations for the static analysis.

	120-day train, 30-day test, 150-day shift			160-day train, 40-day test, 200-day shift			200-day train, 50-day test, 250-day shift		
	success ratio		ideal profit ratio	success ratio		ideal profit ratio	success ratio		ideal profit ratio
	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample
AKBNK	0.54	0.53*** (6.64)	0.12	0.54	0.54*** (3.75)	0.12	0.54	0.54*** (23.64)	0.12
ARCLK	0.53	0.53*** (7.31)	0.09	0.53	0.53*** (2.74)	0.10	0.53	0.53*** (19.13)	0.09
BIMAS	0.53	0.52*** (4.96)	0.09	0.53	0.52*** (3.85)	0.10	0.53	0.54*** (58.44)	0.14
CCOLA	0.53	0.53*** (6.59)	0.13	0.53	0.53*** (6.81)	0.11	0.53	0.52*** (12.72)	0.09
EKGYO	0.56	0.56*** (8.72)	0.27	0.55	0.56*** (11.17)	0.30	0.55	0.56*** (11.8)	0.28
ENKAI	0.54	0.54*** (11.28)	0.20	0.54	0.54*** (11.04)	0.17	0.54	0.55*** (13.69)	0.21
EREGL	0.54	0.53*** (4.34)	0.11	0.54	0.52*** (3.29)	0.08	0.54	0.54*** (14.72)	0.14
FROTO	0.52	0.52*** (9.57)	0.08	0.53	0.53*** (3.58)	0.10	0.54	0.53*** (9.84)	0.11
GARAN	0.54	0.53*** (4.93)	0.11	0.54	0.53*** (5.03)	0.10	0.54	0.53*** (6.73)	0.10
HALKB	0.54	0.54*** (5.56)	0.11	0.54	0.53*** (6.07)	0.07	0.54	0.53*** (10.06)	0.08
ISCTR	0.54	0.54*** (6.05)	0.13	0.54	0.54*** (4.12)	0.12	0.55	0.54*** (22.31)	0.16
KCHOL	0.53	0.52*** (4.73)	0.06	0.53	0.52*** (6.83)	0.08	0.53	0.53*** (52.16)	0.09
KOZAL	0.53	0.52*** (5.84)	0.08	0.52	0.52*** (3.03)	0.08	0.53	0.52*** (3.64)	0.08
KRDMD	0.55	0.55*** (4.91)	0.22	0.55	0.54*** (3.51)	0.19	0.55	0.55*** (4.65)	0.23
OTKAR	0.53	0.53*** (6.43)	0.11	0.53	0.52*** (5.13)	0.06	0.53	0.53*** (3.37)	0.09
PETKM	0.54	0.54*** (3.8)	0.14	0.55	0.54*** (2.91)	0.15	0.55	0.54*** (5.39)	0.14
SAHOL	0.54	0.54*** (15.18)	0.12	0.53	0.53*** (7.2)	0.11	0.54	0.53*** (8.29)	0.12
SISE	0.55	0.54*** (14.38)	0.19	0.55	0.54*** (8.98)	0.20	0.54	0.54*** (3.35)	0.17
TAVHL	0.53	0.53*** (10.29)	0.12	0.53	0.53*** (12.08)	0.09	0.54	0.53*** (2.99)	0.10
TCELL	0.53	0.53*** (8.11)	0.09	0.53	0.53*** (4.4)	0.09	0.53	0.53*** (8.82)	0.09
THYAO	0.53	0.52*** (3.46)	0.07	0.53	0.53*** (2.89)	0.07	0.53	0.53*** (2.88)	0.08
TKFEN	0.54	0.53*** (5.47)	0.10	0.53	0.53*** (11.78)	0.11	0.53	0.53*** (11.09)	0.08
TOASO	0.53	0.53*** (6.64)	0.09	0.53	0.53*** (11.62)	0.09	0.53	0.53*** (4.87)	0.11
TTKOM	0.54	0.54*** (14.06)	0.14	0.54	0.54*** (10.71)	0.15	0.54	0.54*** (8.17)	0.15
TUPRS	0.53	0.53*** (11.07)	0.08	0.53	0.52*** (16.8)	0.04	0.53	0.54*** (4.98)	0.11
ULKER	0.53	0.53*** (9.58)	0.11	0.52	0.53*** (13.87)	0.10	0.53	0.52*** (22.83)	0.08
VAKBN	0.54	0.54*** (6.64)	0.15	0.54	0.54*** (13.74)	0.15	0.54	0.55*** (64.7)	0.17
YKBNK	0.54	0.54*** (15.5)	0.18	0.54	0.55*** (7.52)	0.21	0.55	0.55*** (147.77)	0.21
Mean	0.54	0.53	0.12	0.54	0.53	0.12	0.54	0.54	0.13
Std	0.01	0.01	0.05	0.01	0.01	0.06	0.01	0.01	0.05
Max	0.56	0.56	0.27	0.55	0.56	0.30	0.55	0.56	0.28
Min	0.52	0.52	0.06	0.52	0.52	0.04	0.53	0.52	0.08

Note: The values in the parentheses are t-statistics for the one-sided t-test. ***, ** and * denote significant at the 1%, 5% and 10% level respectively.

Table 8: This table presents Logistic regression classification in-sample and out-of-sample accuracy results and ideal profit ratios for different train/test combinations for the static analysis.

	120-day train, 30-day test, 150-day shift			160-day train, 40-day test, 200-day shift			200-day train, 50-day test, 250-day shift		
	success ratio		ideal profit ratio	success ratio		ideal profit ratio	success ratio		ideal profit ratio
	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample
AKBNK	0.54	0.53*** (15.82)	0.13	0.54	0.54*** (6.1)	0.14	0.54	0.54*** (38.57)	0.14
ARCLK	0.54	0.54*** (7.08)	0.13	0.54	0.54*** (4.35)	0.15	0.54	0.54*** (10.78)	0.14
BIMAS	0.54	0.53*** (20.67)	0.13	0.54	0.54*** (11.15)	0.14	0.54	0.54*** (11.17)	0.15
CCOLA	0.53	0.54*** (9.89)	0.15	0.54	0.54*** (8.2)	0.14	0.54	0.53*** (15.63)	0.12
EKGYO	0.56	0.56*** (9.14)	0.30	0.56	0.56*** (15.29)	0.33	0.56	0.57*** (18.79)	0.31
ENKAI	0.55	0.54*** (10.22)	0.22	0.55	0.54*** (9.95)	0.18	0.54	0.55*** (15.6)	0.23
EREGL	0.55	0.54*** (6.05)	0.14	0.55	0.54*** (3.45)	0.13	0.55	0.55*** (4.27)	0.17
FROTO	0.54	0.54*** (4.89)	0.15	0.54	0.54*** (4.7)	0.16	0.54	0.53*** (5.78)	0.13
GARAN	0.54	0.54*** (5.69)	0.11	0.54	0.54*** (4.5)	0.11	0.54	0.54*** (5.83)	0.11
HALKB	0.54	0.54*** (8.02)	0.14	0.54	0.54*** (7.73)	0.10	0.54	0.54*** (28.96)	0.10
ISCTR	0.55	0.54*** (13.13)	0.15	0.54	0.54*** (7.82)	0.15	0.55	0.55*** (138.38)	0.17
KCHOL	0.53	0.53*** (6.41)	0.09	0.53	0.53*** (13.89)	0.09	0.53	0.53*** (13.22)	0.11
KOZAL	0.54	0.53*** (11.32)	0.10	0.54	0.55*** (32.79)	0.14	0.54	0.53*** (5.27)	0.09
KRDMD	0.55	0.55*** (5.25)	0.24	0.55	0.55*** (3.5)	0.24	0.56	0.55*** (5.29)	0.24
OTKAR	0.54	0.54*** (13.24)	0.15	0.54	0.54*** (9.46)	0.15	0.54	0.54*** (4.38)	0.13
PETKM	0.55	0.54*** (4.32)	0.18	0.55	0.54*** (2.75)	0.17	0.55	0.55*** (5.76)	0.18
SAHOL	0.54	0.54*** (11.83)	0.15	0.54	0.54*** (36.68)	0.15	0.54	0.54*** (38.11)	0.13
SISE	0.55	0.55*** (40.28)	0.24	0.55	0.55*** (26.94)	0.23	0.55	0.55*** (7.43)	0.23
TAVHL	0.54	0.54*** (20.14)	0.15	0.54	0.54*** (12.95)	0.13	0.54	0.54*** (3.23)	0.13
TCELL	0.54	0.54*** (15.01)	0.12	0.54	0.53*** (7.7)	0.12	0.54	0.53*** (49.16)	0.10
THYAO	0.53	0.53*** (4.7)	0.09	0.53	0.53*** (4.69)	0.10	0.53	0.53*** (2.7)	0.11
TKFEN	0.54	0.54*** (15.5)	0.16	0.54	0.54*** (14.37)	0.16	0.54	0.54*** (8.41)	0.14
TOASO	0.54	0.54*** (8.83)	0.14	0.54	0.53*** (9.91)	0.13	0.54	0.54*** (10.39)	0.14
TTKOM	0.54	0.54*** (7.21)	0.18	0.54	0.54*** (4.77)	0.17	0.54	0.54*** (6.67)	0.17
TUPRS	0.54	0.54*** (14.38)	0.12	0.54	0.54*** (5.4)	0.11	0.54	0.54*** (6.72)	0.13
ULKER	0.53	0.53*** (5.67)	0.13	0.53	0.53*** (8.15)	0.14	0.53	0.53*** (27.94)	0.12
VAKBN	0.55	0.54*** (24.75)	0.17	0.55	0.55*** (10.06)	0.17	0.55	0.55*** (28.75)	0.18
YKBNK	0.55	0.54*** (10.07)	0.19	0.55	0.55*** (7.49)	0.21	0.55	0.55*** (30.27)	0.22
Mean	0.54	0.54	0.16	0.54	0.54	0.16	0.54	0.54	0.15
Std	0.01	0.01	0.05	0.01	0.01	0.05	0.01	0.01	0.05
Max	0.56	0.56	0.30	0.56	0.56	0.33	0.56	0.57	0.31
Min	0.53	0.53	0.09	0.53	0.53	0.09	0.53	0.53	0.09

Note: The values in the parentheses are t-statistics for the one-sided t-test. ***, ** and * denote significant at the 1%, 5% and 10% level respectively.

Table 9: This table presents Support Vector Machine (SVM) classification in-sample and out-of-sample accuracy results and ideal profit ratios for different train/test combinations for the static analysis.

	120-day train, 30-day test, 150-day shift			160-day train, 40-day test, 200-day shift			200-day train, 50-day test, 250-day shift		
	success ratio		ideal profit ratio	success ratio		ideal profit ratio	success ratio		ideal profit ratio
	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample
AKBNK	0.54	0.53*** (14.48)	0.12	0.54	0.54*** (4.57)	0.14	0.54	0.54*** (12.67)	0.13
ARCLK	0.54	0.53*** (4.04)	0.12	0.54	0.54*** (2.72)	0.13	0.54	0.54*** (121.17)	0.13
BIMAS	0.54	0.53*** (5.68)	0.11	0.53	0.53*** (10.91)	0.12	0.54	0.54*** (21.19)	0.14
CCOLA	0.53	0.53*** (8.66)	0.13	0.53	0.53*** (6.5)	0.12	0.53	0.53*** (26.27)	0.12
EKGYO	0.56	0.56*** (8.98)	0.30	0.56	0.56*** (16.21)	0.32	0.56	0.57*** (268)	0.32
ENKAI	0.54	0.54*** (9.41)	0.20	0.55	0.54*** (10.01)	0.17	0.54	0.55*** (12.33)	0.21
EREGL	0.55	0.54*** (5.93)	0.14	0.55	0.54*** (3.85)	0.12	0.55	0.55*** (11.61)	0.17
FROTO	0.54	0.54*** (4.74)	0.13	0.54	0.54*** (4.61)	0.14	0.53	0.53*** (3.72)	0.11
GARAN	0.54	0.54*** (6.84)	0.11	0.54	0.54*** (4.21)	0.11	0.54	0.54*** (5.12)	0.11
HALKB	0.54	0.54*** (6.92)	0.13	0.54	0.53*** (7.58)	0.08	0.54	0.53*** (23.67)	0.09
ISCTR	0.55	0.54*** (10.93)	0.15	0.55	0.54*** (5.23)	0.14	0.55	0.54*** (83.8)	0.16
KCHOL	0.53	0.53*** (6.2)	0.09	0.53	0.53*** (10.06)	0.08	0.53	0.53*** (23.91)	0.10
KOZAL	0.54	0.53*** (11.15)	0.10	0.54	0.54*** (9.18)	0.14	0.54	0.53*** (5.95)	0.10
KRDMD	0.55	0.55*** (4.71)	0.25	0.55	0.55*** (3.53)	0.23	0.56	0.55*** (5.04)	0.24
OTKAR	0.54	0.53*** (6.57)	0.13	0.54	0.54*** (12.07)	0.13	0.54	0.54*** (2.83)	0.12
PETKM	0.55	0.54*** (4.58)	0.18	0.55	0.54*** (3.02)	0.16	0.55	0.55*** (5.37)	0.18
SAHOL	0.54	0.54*** (12.82)	0.13	0.54	0.54*** (27.95)	0.13	0.54	0.54*** (36.84)	0.13
SISE	0.55	0.55*** (23.72)	0.23	0.55	0.55*** (26.73)	0.23	0.55	0.55*** (12.64)	0.22
TAVHL	0.54	0.54*** (18.45)	0.14	0.54	0.54*** (16.08)	0.12	0.54	0.53*** (3.76)	0.14
TCELL	0.53	0.53*** (9.06)	0.10	0.53	0.53*** (6.59)	0.10	0.53	0.53*** (12.09)	0.10
THYAO	0.54	0.53*** (5.89)	0.09	0.53	0.53*** (4.48)	0.10	0.54	0.53*** (3.41)	0.10
TKFEN	0.54	0.54*** (12.82)	0.13	0.54	0.54*** (6.82)	0.14	0.54	0.53*** (4.75)	0.11
TOASO	0.54	0.53*** (13.89)	0.12	0.54	0.53*** (7.23)	0.11	0.53	0.53*** (7.94)	0.12
TTKOM	0.54	0.54*** (7.74)	0.17	0.54	0.54*** (8.75)	0.17	0.54	0.54*** (7.24)	0.16
TUPRS	0.54	0.53*** (8.59)	0.11	0.54	0.53*** (5.26)	0.10	0.54	0.54*** (4.84)	0.12
ULKER	0.53	0.53*** (9.2)	0.12	0.53	0.53*** (33.02)	0.12	0.53	0.52*** (7.42)	0.09
VAKBN	0.55	0.54*** (17.8)	0.17	0.55	0.54*** (9.33)	0.16	0.54	0.55*** (43.41)	0.19
YKBNK	0.55	0.54*** (12.68)	0.19	0.55	0.55*** (8.39)	0.21	0.55	0.55*** (32.52)	0.22
Mean	0.54	0.54	0.15	0.54	0.54	0.14	0.54	0.54	0.15
Std	0.01	0.01	0.05	0.01	0.01	0.05	0.01	0.01	0.05
Max	0.56	0.56	0.30	0.56	0.56	0.32	0.56	0.57	0.32
Min	0.53	0.53	0.09	0.53	0.53	0.08	0.53	0.52	0.09

Note: The values in the parentheses are t-statistics for the one-sided t-test. ***, ** and * denote significant at the 1%, 5% and 10% level respectively.

Table 10: This table presents Extreme Gradient Boosting (XGBoost) classification in-sample and out-of-sample accuracy results and ideal profit ratios for different train/test combinations for the static analysis.

	120-day train, 30-day test, 150-day shift			160-day train, 40-day test, 200-day shift			200-day train, 50-day test, 250-day shift		
	success ratio		ideal profit ratio	success ratio		ideal profit ratio	success ratio		ideal profit ratio
	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample
AKBNK	0.56	0.54*** (15.33)	0.13	0.56	0.55*** (6.28)	0.15	0.56	0.54*** (15.29)	0.13
ARCLK	0.57	0.54*** (19.5)	0.15	0.56	0.55*** (10.58)	0.16	0.56	0.55*** (25.85)	0.16
BIMAS	0.56	0.54*** (8.11)	0.15	0.56	0.54*** (14.49)	0.15	0.56	0.54*** (12.81)	0.18
CCOLA	0.56	0.54*** (14.89)	0.16	0.56	0.54*** (11.58)	0.15	0.55	0.54*** (36.53)	0.15
EKGYO	0.58	0.57*** (8.88)	0.30	0.57	0.57*** (11.14)	0.33	0.57	0.57*** (87.41)	0.31
ENKAI	0.57	0.54*** (16.77)	0.23	0.57	0.54*** (12.69)	0.21	0.56	0.55*** (27.19)	0.26
EREGL	0.57	0.54*** (6.74)	0.16	0.57	0.54*** (4.56)	0.14	0.57	0.55*** (4.09)	0.18
FROTO	0.56	0.55*** (5.58)	0.17	0.56	0.55*** (4.71)	0.16	0.55	0.54*** (11.87)	0.15
GARAN	0.56	0.53*** (4.18)	0.11	0.56	0.54*** (3.95)	0.11	0.56	0.54*** (4.86)	0.11
HALKB	0.56	0.54*** (5.49)	0.13	0.56	0.53*** (6.1)	0.09	0.56	0.53*** (19.28)	0.09
ISCTR	0.57	0.54*** (8.08)	0.16	0.56	0.54*** (7.09)	0.14	0.56	0.55*** (21.02)	0.17
KCHOL	0.56	0.53*** (6.7)	0.10	0.56	0.54*** (94.32)	0.13	0.55	0.54*** (64)	0.13
KOZAL	0.56	0.53*** (9.03)	0.10	0.56	0.55*** (13.7)	0.14	0.56	0.53*** (3.37)	0.11
KRDMD	0.57	0.55*** (4.12)	0.25	0.57	0.55*** (3.2)	0.24	0.57	0.55*** (3.66)	0.25
OTKAR	0.56	0.54*** (15.7)	0.16	0.56	0.55*** (11.49)	0.16	0.56	0.55*** (6.87)	0.16
PETKM	0.57	0.54*** (4.42)	0.18	0.57	0.54*** (2.79)	0.17	0.57	0.54*** (4.22)	0.18
SAHOL	0.57	0.54*** (8.01)	0.15	0.56	0.55*** (5.53)	0.16	0.56	0.54*** (16.59)	0.17
SISE	0.58	0.55*** (18.67)	0.25	0.57	0.56*** (18.24)	0.25	0.57	0.55*** (13.96)	0.25
TAVHL	0.57	0.55*** (14.32)	0.17	0.56	0.55*** (25.45)	0.14	0.56	0.54*** (3.39)	0.15
TCELL	0.56	0.54*** (20.23)	0.14	0.56	0.54*** (12.49)	0.13	0.56	0.53*** (659)	0.12
THYAO	0.56	0.53*** (4.91)	0.09	0.55	0.53*** (4.43)	0.09	0.55	0.54*** (3.88)	0.11
TKFEN	0.57	0.55*** (24.34)	0.17	0.56	0.55*** (15.02)	0.17	0.56	0.54*** (3.78)	0.14
TOASO	0.56	0.55*** (7.07)	0.16	0.56	0.54*** (9.63)	0.17	0.56	0.54*** (6.49)	0.16
TTKOM	0.57	0.55*** (11.85)	0.19	0.56	0.54*** (6.03)	0.18	0.56	0.55*** (6.66)	0.20
TUPRS	0.57	0.54*** (18.1)	0.14	0.56	0.55*** (9.88)	0.14	0.56	0.55*** (858)	0.15
ULKER	0.56	0.53*** (8.55)	0.15	0.55	0.54*** (14.38)	0.15	0.55	0.53*** (17)	0.14
VAKBN	0.57	0.55*** (22.6)	0.18	0.56	0.55*** (32.39)	0.18	0.56	0.55*** (112.53)	0.20
YKBNK	0.57	0.54*** (16.24)	0.19	0.57	0.55*** (7.5)	0.22	0.56	0.55*** (93.4)	0.22
Mean	0.57	0.54	0.17	0.56	0.55	0.16	0.56	0.54	0.17
Std	0.01	0.01	0.05	0.01	0.01	0.05	0.01	0.01	0.05
Max	0.58	0.57	0.30	0.57	0.57	0.33	0.57	0.57	0.31
Min	0.56	0.53	0.09	0.55	0.53	0.09	0.55	0.53	0.09

Note: The values in the parentheses are t-statistics for the one-sided t-test. ***, ** and * denote significant at the 1%, 5% and 10% level respectively.

Table 11: This table presents Naive Bayes classification in-sample and out-of-sample accuracy results and ideal profit ratios for different sliding windows for the dynamic analysis.

	2-days train, 1-day test, 1-day shift			5-days train, 2-day test, 1-day shift			20-days train, 5-day test, 1-day shift		
	success ratio		ideal profit ratio	success ratio		ideal profit ratio	success ratio		ideal profit ratio
	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample
AKBNK	0.54	0.52*** (16.86)	0.06	0.53	0.52*** (22.96)	0.07	0.53	0.53*** (32.41)	0.08
ARCLK	0.54	0.51*** (11.02)	0.05	0.53	0.52*** (17.9)	0.05	0.52	0.52*** (20.99)	0.05
BIMAS	0.53	0.51*** (9.13)	0.04	0.52	0.51*** (14.51)	0.05	0.52	0.51*** (22)	0.05
CCOLA	0.53	0.51*** (6.68)	0.03	0.52	0.51*** (10.49)	0.03	0.51	0.51*** (14.84)	0.03
EKGYO	0.55	0.53*** (23)	0.14	0.55	0.54*** (36.24)	0.19	0.55	0.54*** (47.5)	0.21
ENKAI	0.54	0.52*** (15.37)	0.09	0.53	0.52*** (20.47)	0.10	0.53	0.53*** (27.73)	0.11
EREGL	0.54	0.52*** (14.97)	0.07	0.54	0.52*** (24.43)	0.09	0.53	0.53*** (31.18)	0.09
FROTO	0.53	0.51*** (9)	0.04	0.52	0.51*** (13.08)	0.04	0.52	0.51*** (17.27)	0.05
GARAN	0.54	0.52*** (16.04)	0.05	0.53	0.52*** (24.86)	0.06	0.53	0.53*** (41.13)	0.07
HALKB	0.54	0.52*** (15)	0.05	0.53	0.52*** (23.05)	0.07	0.53	0.53*** (34.3)	0.08
ISCTR	0.54	0.52*** (17.43)	0.07	0.53	0.52*** (23.93)	0.09	0.53	0.53*** (38.47)	0.10
KCHOL	0.53	0.51*** (6.77)	0.03	0.52	0.51*** (13.88)	0.03	0.51	0.51*** (20.01)	0.04
KOZAL	0.53	0.51*** (7.94)	0.03	0.53	0.51*** (14.56)	0.04	0.52	0.51*** (20.1)	0.04
KRDMD	0.54	0.52*** (13.2)	0.09	0.53	0.52*** (21.44)	0.11	0.53	0.53*** (30.24)	0.14
OTKAR	0.53	0.51*** (10.16)	0.04	0.52	0.51*** (14.98)	0.04	0.52	0.52*** (22.98)	0.06
PETKM	0.54	0.52*** (17.65)	0.09	0.54	0.53*** (24.58)	0.11	0.54	0.53*** (37.75)	0.13
SAHOL	0.54	0.52*** (15.89)	0.06	0.53	0.52*** (21.06)	0.07	0.53	0.52*** (29.15)	0.08
SISE	0.55	0.53*** (21.4)	0.13	0.54	0.53*** (29.01)	0.14	0.54	0.53*** (35.8)	0.15
TAVHL	0.53	0.51*** (9.13)	0.04	0.52	0.51*** (14.11)	0.04	0.52	0.51*** (18.51)	0.04
TCELL	0.53	0.51*** (11.02)	0.04	0.52	0.51*** (17.08)	0.05	0.52	0.51*** (18.18)	0.05
THYAO	0.54	0.51*** (10.9)	0.04	0.53	0.52*** (18.91)	0.05	0.52	0.52*** (31.66)	0.06
TKFEN	0.54	0.52*** (12.58)	0.05	0.53	0.52*** (18.59)	0.06	0.52	0.52*** (21.91)	0.06
TOASO	0.53	0.51*** (10.81)	0.04	0.52	0.51*** (13.72)	0.04	0.51	0.51*** (15.75)	0.03
TTKOM	0.54	0.52*** (12.77)	0.07	0.53	0.52*** (18.94)	0.08	0.52	0.52*** (21.24)	0.09
TUPRS	0.54	0.51*** (10.44)	0.04	0.53	0.51*** (14.95)	0.04	0.52	0.52*** (23.32)	0.05
ULKER	0.53	0.51*** (7.64)	0.03	0.52	0.51*** (9.7)	0.03	0.51	0.51*** (11.32)	0.03
VAKBN	0.54	0.52*** (17.54)	0.08	0.53	0.52*** (24.86)	0.09	0.53	0.53*** (30.66)	0.10
YKBNK	0.54	0.52*** (20.04)	0.10	0.54	0.53*** (26.64)	0.12	0.54	0.53*** (42.51)	0.15
Mean	0.54	0.52	0.06	0.53	0.52	0.07	0.53	0.52	0.08
Std	0.01	0.01	0.03	0.01	0.01	0.04	0.01	0.01	0.04
Max	0.55	0.53	0.14	0.55	0.54	0.19	0.55	0.54	0.21
Min	0.53	0.51	0.03	0.52	0.51	0.03	0.51	0.51	0.03

Note: The values in the parentheses are t-statistics for the one-sided t-test. ***, ** and * denote significant at the 1%, 5% and 10% level respectively.

Table 12: This table presents k-Nearest Neighbors (kNN) classification in-sample and out-of-sample accuracy results and ideal profit ratios for different sliding windows for the dynamic analysis.

	2-days train, 1-day test, 1-day shift			5-days train, 2-day test, 1-day shift			20-days train, 5-day test, 1-day shift		
	success ratio		ideal profit ratio	success ratio		ideal profit ratio	success ratio		ideal profit ratio
	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample
AKBNK	0.54	0.53*** (25.33)	0.09	0.55	0.53*** (36.78)	0.10	0.55	0.53*** (59.46)	0.11
ARCLK	0.54	0.52*** (19.26)	0.08	0.55	0.53*** (32.7)	0.09	0.55	0.53*** (59.55)	0.11
BIMAS	0.54	0.52*** (16.6)	0.07	0.54	0.52*** (31)	0.09	0.55	0.53*** (50.05)	0.10
CCOLA	0.54	0.51*** (11.15)	0.06	0.54	0.52*** (22.78)	0.07	0.54	0.52*** (41.35)	0.09
EKGYO	0.56	0.55*** (38.31)	0.23	0.56	0.55*** (56.99)	0.25	0.56	0.55*** (87.31)	0.26
ENKAI	0.54	0.53*** (29.48)	0.15	0.55	0.54*** (42.48)	0.17	0.55	0.54*** (62.79)	0.19
EREGL	0.54	0.53*** (23.15)	0.11	0.55	0.53*** (34.38)	0.12	0.55	0.53*** (45.54)	0.13
FROTO	0.54	0.52*** (13.87)	0.06	0.54	0.52*** (24.43)	0.08	0.55	0.53*** (44.23)	0.10
GARAN	0.54	0.52*** (22.03)	0.08	0.54	0.53*** (30.33)	0.08	0.55	0.53*** (49.5)	0.09
HALKB	0.54	0.52*** (18.57)	0.07	0.54	0.53*** (28.4)	0.08	0.55	0.53*** (50.94)	0.09
ISCTR	0.54	0.53*** (28.08)	0.11	0.55	0.53*** (38.56)	0.13	0.55	0.54*** (61.63)	0.14
KCHOL	0.53	0.51*** (12.85)	0.04	0.54	0.52*** (21.78)	0.06	0.54	0.52*** (45.82)	0.07
KOZAL	0.54	0.52*** (13.75)	0.05	0.54	0.52*** (24.56)	0.06	0.55	0.53*** (41.79)	0.08
KRDMD	0.55	0.53*** (26.17)	0.17	0.55	0.54*** (34.47)	0.18	0.55	0.54*** (50.8)	0.20
OTKAR	0.54	0.52*** (16.42)	0.07	0.54	0.52*** (30.86)	0.08	0.55	0.53*** (48.85)	0.11
PETKM	0.55	0.53*** (26.44)	0.14	0.55	0.54*** (36.24)	0.15	0.55	0.54*** (43.2)	0.15
SAHOL	0.54	0.53*** (23.67)	0.09	0.55	0.53*** (38.03)	0.10	0.55	0.53*** (55.47)	0.12
SISE	0.55	0.54*** (33.59)	0.17	0.55	0.54*** (49.6)	0.19	0.56	0.55*** (76.45)	0.21
TAVHL	0.54	0.52*** (16.55)	0.07	0.55	0.52*** (30.98)	0.09	0.55	0.53*** (56.42)	0.10
TCELL	0.54	0.52*** (15.86)	0.06	0.54	0.52*** (28.99)	0.08	0.55	0.53*** (50.63)	0.09
THYAO	0.53	0.52*** (13.83)	0.06	0.54	0.52*** (23.66)	0.06	0.54	0.52*** (37.8)	0.07
TKFEN	0.54	0.52*** (18.75)	0.08	0.55	0.53*** (31.21)	0.10	0.55	0.53*** (61.23)	0.11
TOASO	0.54	0.52*** (16.69)	0.07	0.54	0.52*** (27.42)	0.08	0.55	0.53*** (45.48)	0.10
TTKOM	0.54	0.53*** (25.87)	0.12	0.55	0.53*** (36.01)	0.14	0.55	0.54*** (54.71)	0.15
TUPRS	0.54	0.52*** (18.04)	0.06	0.55	0.53*** (31.2)	0.08	0.55	0.53*** (52.9)	0.10
ULKER	0.54	0.51*** (12.17)	0.06	0.54	0.52*** (24.51)	0.08	0.54	0.53*** (42.83)	0.09
VAKBN	0.55	0.53*** (29.75)	0.12	0.55	0.54*** (45.16)	0.14	0.55	0.54*** (71.57)	0.15
YKBNK	0.55	0.53*** (28.45)	0.15	0.55	0.54*** (41.28)	0.16	0.55	0.54*** (64.38)	0.18
Mean	0.54	0.52	0.10	0.55	0.53	0.11	0.55	0.53	0.12
Std	0.01	0.01	0.05	0.01	0.01	0.05	0.00	0.01	0.05
Max	0.56	0.55	0.23	0.56	0.55	0.25	0.56	0.55	0.26
Min	0.53	0.51	0.04	0.54	0.52	0.06	0.54	0.52	0.07

Note: The values in the parentheses are t-statistics for the one-sided t-test. ***, ** and * denote significant at the 1%, 5% and 10% level respectively.

Table 13: This table presents Random forest classification in-sample and out-of-sample accuracy results and ideal profit ratios for different sliding windows for the dynamic analysis.

	2-days train, 1-day test, 1-day shift			5-days train, 2-day test, 1-day shift			20-days train, 5-day test, 1-day shift		
	success ratio		ideal profit ratio	success ratio		ideal profit ratio	success ratio		ideal profit ratio
	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample
AKBNK	0.56	0.53*** (22.65)	0.09	0.55	0.53*** (35.53)	0.11	0.54	0.54*** (52.53)	0.12
ARCLK	0.56	0.52*** (18.25)	0.07	0.55	0.53*** (32.71)	0.09	0.54	0.53*** (50.93)	0.10
BIMAS	0.56	0.52*** (15.41)	0.07	0.55	0.52*** (29.56)	0.09	0.54	0.53*** (49.82)	0.10
CCOLA	0.56	0.51*** (12.02)	0.06	0.55	0.52*** (21.07)	0.07	0.53	0.52*** (38.2)	0.08
EKGYO	0.58	0.55*** (36.37)	0.24	0.57	0.55*** (57.55)	0.26	0.56	0.56*** (83.55)	0.27
ENKAI	0.56	0.53*** (26.55)	0.16	0.55	0.54*** (42.49)	0.18	0.54	0.54*** (66.16)	0.19
EREGL	0.57	0.53*** (24.42)	0.11	0.55	0.53*** (33.56)	0.12	0.54	0.54*** (47)	0.12
FROTO	0.56	0.52*** (15.4)	0.06	0.55	0.52*** (24.2)	0.08	0.53	0.52*** (39.52)	0.09
GARAN	0.56	0.53*** (23.04)	0.08	0.55	0.53*** (35.54)	0.09	0.54	0.53*** (52.93)	0.10
HALKB	0.56	0.52*** (20.24)	0.07	0.55	0.53*** (32.32)	0.09	0.54	0.53*** (54.54)	0.10
ISCTR	0.57	0.53*** (24.47)	0.11	0.55	0.54*** (40.38)	0.14	0.54	0.54*** (63.08)	0.15
KCHOL	0.56	0.51*** (12.7)	0.05	0.55	0.52*** (21.15)	0.06	0.53	0.52*** (45.51)	0.07
KOZAL	0.56	0.51*** (11.99)	0.05	0.55	0.52*** (22.12)	0.06	0.53	0.52*** (38.37)	0.07
KRDMD	0.57	0.53*** (24.61)	0.17	0.56	0.54*** (37.73)	0.20	0.55	0.54*** (53.22)	0.21
OTKAR	0.56	0.52*** (17.16)	0.07	0.55	0.52*** (28.23)	0.08	0.54	0.53*** (50.99)	0.10
PETKM	0.57	0.53*** (23.46)	0.14	0.56	0.54*** (34.26)	0.15	0.55	0.54*** (47.16)	0.16
SAHOL	0.56	0.53*** (22.46)	0.09	0.55	0.53*** (40.2)	0.11	0.54	0.54*** (59.84)	0.12
SISE	0.57	0.54*** (30.86)	0.18	0.56	0.55*** (50.08)	0.20	0.55	0.55*** (77.39)	0.21
TAVHL	0.56	0.52*** (16.7)	0.07	0.55	0.52*** (28.72)	0.08	0.54	0.53*** (48.63)	0.10
TCELL	0.56	0.52*** (16.41)	0.07	0.55	0.53*** (30.57)	0.08	0.54	0.53*** (54.02)	0.09
THYAO	0.56	0.52*** (15.15)	0.06	0.55	0.52*** (24.19)	0.06	0.53	0.52*** (38.01)	0.07
TKFEN	0.56	0.52*** (18.49)	0.08	0.55	0.53*** (29.62)	0.10	0.54	0.53*** (49.29)	0.10
TOASO	0.56	0.52*** (15.82)	0.07	0.55	0.52*** (26.4)	0.08	0.54	0.53*** (46.18)	0.10
TTKOM	0.56	0.53*** (24.07)	0.13	0.55	0.53*** (38.64)	0.15	0.54	0.54*** (64.16)	0.16
TUPRS	0.56	0.52*** (17.24)	0.07	0.55	0.53*** (31.27)	0.08	0.54	0.53*** (47.56)	0.09
ULKER	0.56	0.52*** (14.8)	0.06	0.55	0.52*** (24.85)	0.08	0.53	0.52*** (40.56)	0.10
VAKBN	0.57	0.53*** (28.93)	0.13	0.55	0.54*** (46.35)	0.15	0.55	0.54*** (76.65)	0.16
YKBNK	0.57	0.53*** (27.55)	0.15	0.56	0.54*** (45.21)	0.18	0.55	0.54*** (69.99)	0.19
Mean	0.56	0.52	0.10	0.55	0.53	0.12	0.54	0.53	0.13
Std	0.01	0.01	0.05	0.00	0.01	0.05	0.01	0.01	0.05
Max	0.58	0.55	0.24	0.57	0.55	0.26	0.56	0.56	0.27
Min	0.56	0.51	0.05	0.55	0.52	0.06	0.53	0.52	0.07

Note: The values in the parentheses are t-statistics for the one-sided t-test. ***, ** and * denote significant at the 1%, 5% and 10% level respectively.

Table 14: This table presents Extreme Gradient Boosting (XGBoost) classification in-sample and out-of-sample accuracy results and ideal profit ratios for different sliding windows for the dynamic analysis.

	2-days train, 1-day test, 1-day shift			5-days train, 2-day test, 1-day shift			20-days train, 5-day test, 1-day shift		
	success ratio		ideal profit ratio	success ratio		ideal profit ratio	success ratio		ideal profit ratio
	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample
AKBNK	0.60	0.53*** (23.63)	0.09	0.61	0.53*** (37.72)	0.11	0.60	0.53*** (60.88)	0.12
ARCLK	0.60	0.52*** (18.47)	0.08	0.61	0.53*** (38.51)	0.12	0.60	0.54*** (65.51)	0.14
BIMAS	0.60	0.52*** (18.32)	0.07	0.60	0.53*** (38.03)	0.11	0.60	0.53*** (72.32)	0.13
CCOLA	0.59	0.52*** (13.24)	0.06	0.60	0.52*** (29.24)	0.10	0.59	0.53*** (59.84)	0.12
EKGYO	0.62	0.55*** (38.41)	0.24	0.61	0.55*** (58.47)	0.26	0.61	0.56*** (88.31)	0.27
ENKAI	0.60	0.53*** (28.21)	0.15	0.60	0.54*** (46.37)	0.19	0.59	0.54*** (74.54)	0.21
EREGL	0.61	0.53*** (22.94)	0.12	0.61	0.54*** (39.82)	0.14	0.61	0.54*** (56.39)	0.15
FROTO	0.59	0.52*** (16.69)	0.06	0.60	0.53*** (32.25)	0.11	0.59	0.54*** (61.38)	0.14
GARAN	0.60	0.52*** (19.88)	0.07	0.61	0.53*** (35.37)	0.09	0.60	0.53*** (52.32)	0.10
HALKB	0.60	0.52*** (20.77)	0.08	0.61	0.53*** (32.98)	0.09	0.60	0.53*** (53.89)	0.10
ISCTR	0.61	0.53*** (26.12)	0.11	0.61	0.54*** (43.87)	0.14	0.60	0.54*** (63.2)	0.15
KCHOL	0.59	0.52*** (13.27)	0.05	0.61	0.52*** (30.45)	0.08	0.60	0.53*** (55.97)	0.10
KOZAL	0.60	0.52*** (15.42)	0.05	0.61	0.53*** (30.61)	0.09	0.60	0.53*** (50.51)	0.10
KRDMD	0.61	0.54*** (25.85)	0.18	0.61	0.54*** (39.07)	0.21	0.61	0.54*** (52.98)	0.22
OTKAR	0.59	0.52*** (19.05)	0.07	0.60	0.53*** (35.9)	0.11	0.59	0.54*** (63.05)	0.14
PETKM	0.61	0.54*** (25.63)	0.14	0.61	0.54*** (35.77)	0.16	0.61	0.54*** (48.89)	0.17
SAHOL	0.60	0.53*** (23.86)	0.10	0.61	0.53*** (42.05)	0.12	0.60	0.54*** (63.04)	0.14
SISE	0.61	0.54*** (34.19)	0.19	0.61	0.55*** (53.05)	0.22	0.60	0.55*** (87.32)	0.24
TAVHL	0.60	0.52*** (18.79)	0.07	0.61	0.53*** (40.17)	0.11	0.60	0.54*** (72.43)	0.14
TCELL	0.60	0.52*** (18.46)	0.08	0.61	0.53*** (36.02)	0.10	0.60	0.53*** (63.26)	0.11
THYAO	0.60	0.52*** (14.15)	0.05	0.61	0.52*** (25.08)	0.08	0.60	0.52*** (34.92)	0.08
TKFEN	0.60	0.52*** (20.55)	0.09	0.61	0.53*** (38.39)	0.13	0.60	0.54*** (66.87)	0.15
TOASO	0.60	0.52*** (18.04)	0.07	0.61	0.53*** (34.91)	0.10	0.60	0.53*** (59.59)	0.13
TTKOM	0.60	0.53*** (26.68)	0.13	0.60	0.54*** (41.64)	0.16	0.60	0.54*** (65.78)	0.18
TUPRS	0.61	0.52*** (19.52)	0.07	0.61	0.53*** (39.11)	0.11	0.60	0.54*** (66.36)	0.13
ULKER	0.59	0.52*** (15.7)	0.06	0.60	0.52*** (30.29)	0.10	0.59	0.53*** (52.81)	0.12
VAKBN	0.61	0.53*** (26.72)	0.13	0.61	0.54*** (48.33)	0.16	0.60	0.54*** (75.45)	0.17
YKBNK	0.61	0.53*** (28.11)	0.15	0.61	0.54*** (47.18)	0.18	0.60	0.54*** (75.2)	0.20
Mean	0.60	0.53	0.10	0.61	0.53	0.13	0.60	0.54	0.15
Std	0.01	0.01	0.05	0.00	0.01	0.05	0.01	0.01	0.05
Max	0.62	0.55	0.24	0.61	0.55	0.26	0.61	0.56	0.27
Min	0.59	0.52	0.05	0.60	0.52	0.08	0.59	0.52	0.08

Note: The values in the parentheses are t-statistics for the one-sided t-test. ***, ** and * denote significant at the 1%, 5% and 10% level respectively.

Table 15: This table presents Support Vector Machine (SVM) classification in-sample and out-of-sample accuracy results and ideal profit ratios for different sliding windows for the dynamic analysis.

	2-days train, 1-day test, 1-day shift			5-days train, 2-day test, 1-day shift			20-days train, 5-day test, 1-day shift		
	success ratio		ideal profit ratio	success ratio		ideal profit ratio	success ratio		ideal profit ratio
	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample
AKBNK	0.56	0.53*** (28.87)	0.11	0.55	0.54*** (45.58)	0.12	0.55	0.54*** (65.63)	0.13
ARCLK	0.56	0.53*** (21.86)	0.08	0.55	0.53*** (36.38)	0.10	0.54	0.54*** (59.88)	0.11
BIMAS	0.56	0.52*** (18.13)	0.08	0.54	0.53*** (34.43)	0.10	0.54	0.53*** (57.18)	0.11
CCOLA	0.55	0.52*** (13.9)	0.07	0.54	0.52*** (24.84)	0.08	0.54	0.53*** (45.02)	0.10
EKGYO	0.57	0.55*** (40.53)	0.25	0.57	0.55*** (60.45)	0.26	0.56	0.56*** (86.17)	0.27
ENKAI	0.56	0.54*** (30.36)	0.17	0.55	0.54*** (46.71)	0.19	0.55	0.54*** (64.46)	0.20
EREGL	0.56	0.53*** (25.63)	0.12	0.55	0.54*** (39.24)	0.13	0.55	0.54*** (56.87)	0.14
FROTO	0.55	0.52*** (16.39)	0.07	0.54	0.52*** (26.65)	0.09	0.54	0.53*** (47.21)	0.11
GARAN	0.56	0.53*** (26.41)	0.09	0.55	0.53*** (39.54)	0.10	0.54	0.54*** (61.07)	0.11
HALKB	0.56	0.53*** (22.97)	0.09	0.55	0.53*** (38.26)	0.10	0.54	0.54*** (61.74)	0.11
ISCTR	0.56	0.53*** (28.14)	0.13	0.55	0.54*** (45.23)	0.14	0.55	0.54*** (68.61)	0.15
KCHOL	0.56	0.52*** (16.38)	0.06	0.54	0.52*** (27.49)	0.07	0.54	0.53*** (52.51)	0.09
KOZAL	0.56	0.52*** (14.63)	0.06	0.55	0.52*** (28.03)	0.08	0.54	0.53*** (53.08)	0.09
KRDMD	0.57	0.54*** (27.07)	0.18	0.56	0.54*** (39.11)	0.21	0.55	0.55*** (59.35)	0.22
OTKAR	0.56	0.52*** (19.74)	0.08	0.55	0.53*** (32.76)	0.10	0.54	0.53*** (53.72)	0.11
PETKM	0.57	0.54*** (29.69)	0.15	0.56	0.54*** (42.33)	0.16	0.55	0.54*** (54.68)	0.17
SAHOL	0.56	0.53*** (27.26)	0.11	0.55	0.53*** (41.72)	0.12	0.54	0.54*** (62.54)	0.13
SISE	0.57	0.54*** (37.84)	0.20	0.56	0.55*** (56.53)	0.21	0.55	0.55*** (83.19)	0.22
TAVHL	0.56	0.52*** (19.24)	0.08	0.55	0.53*** (33.85)	0.10	0.54	0.53*** (60.2)	0.12
TCELL	0.56	0.52*** (18.8)	0.08	0.55	0.53*** (34.42)	0.09	0.54	0.53*** (56.27)	0.10
THYAO	0.56	0.52*** (17.28)	0.07	0.55	0.53*** (30.13)	0.08	0.54	0.53*** (43.78)	0.09
TKFEN	0.56	0.53*** (22.25)	0.09	0.55	0.53*** (35.99)	0.11	0.55	0.54*** (62.69)	0.12
TOASO	0.56	0.52*** (19.73)	0.08	0.55	0.53*** (30.55)	0.09	0.54	0.53*** (50.78)	0.11
TTKOM	0.56	0.53*** (27.92)	0.14	0.55	0.54*** (42.96)	0.16	0.55	0.54*** (65.98)	0.17
TUPRS	0.56	0.52*** (19.82)	0.07	0.55	0.53*** (34.78)	0.10	0.54	0.54*** (58.5)	0.11
ULKER	0.55	0.52*** (13.72)	0.07	0.54	0.52*** (26.77)	0.09	0.54	0.53*** (43.83)	0.11
VAKBN	0.56	0.54*** (32.22)	0.14	0.55	0.54*** (50.86)	0.16	0.55	0.54*** (84.44)	0.17
YKBNK	0.56	0.54*** (31.56)	0.17	0.55	0.54*** (45.98)	0.18	0.55	0.55*** (71.64)	0.20
Mean	0.56	0.53	0.11	0.55	0.53	0.13	0.54	0.54	0.14
Std	0.01	0.01	0.05	0.01	0.01	0.05	0.01	0.01	0.05
Max	0.57	0.55	0.25	0.57	0.55	0.26	0.56	0.56	0.27
Min	0.55	0.52	0.06	0.54	0.52	0.07	0.54	0.53	0.09

Note: The values in the parentheses are t-statistics for the one-sided t-test. ***, ** and * denote significant at the 1%, 5% and 10% level respectively.

Table 16: This table presents the logistic regression classification in-sample and out-of-sample accuracy results and ideal profit ratios for different sliding windows for the dynamic analysis.

	2-days train, 1-day test, 1-day shift			5-days train, 2-day test, 1-day shift			20-days train, 5-day test, 1-day shift		
	success ratio		ideal profit ratio	success ratio		ideal profit ratio	success ratio		ideal profit ratio
	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample	in-sample	out-sample (t-stat)	out-sample
AKBNK	0.57	0.53*** (26.59)	0.10	0.55	0.54*** (43.6)	0.12	0.54	0.54*** (67.29)	0.13
ARCLK	0.57	0.53*** (26.99)	0.10	0.56	0.54*** (42.77)	0.12	0.55	0.54*** (67.49)	0.14
BIMAS	0.56	0.52*** (20.67)	0.10	0.55	0.53*** (39.4)	0.12	0.54	0.53*** (67.41)	0.13
CCOLA	0.56	0.52*** (17.69)	0.08	0.55	0.53*** (31.23)	0.11	0.54	0.53*** (54.5)	0.13
EKGYO	0.58	0.55*** (41.2)	0.24	0.57	0.56*** (60.36)	0.27	0.56	0.56*** (85.62)	0.29
ENKAI	0.57	0.53*** (27.63)	0.15	0.56	0.54*** (45.56)	0.19	0.55	0.54*** (73.88)	0.21
EREGL	0.57	0.53*** (27.97)	0.13	0.56	0.54*** (41.92)	0.15	0.55	0.54*** (61.31)	0.16
FROTO	0.57	0.53*** (21.6)	0.09	0.55	0.53*** (33.4)	0.12	0.54	0.53*** (54.42)	0.14
GARAN	0.56	0.53*** (22.97)	0.09	0.55	0.53*** (37.13)	0.10	0.54	0.54*** (62.72)	0.12
HALKB	0.56	0.53*** (22.37)	0.09	0.55	0.53*** (35.76)	0.11	0.54	0.54*** (61.25)	0.12
ISCTR	0.57	0.53*** (27.69)	0.13	0.56	0.54*** (43.76)	0.15	0.55	0.54*** (68.42)	0.16
KCHOL	0.56	0.52*** (19.31)	0.07	0.55	0.53*** (34.17)	0.09	0.54	0.53*** (57.02)	0.10
KOZAL	0.57	0.53*** (23.34)	0.08	0.55	0.53*** (36.84)	0.10	0.54	0.54*** (55.38)	0.11
KRDMD	0.57	0.54*** (30.03)	0.19	0.56	0.54*** (41.42)	0.22	0.55	0.55*** (58.14)	0.24
OTKAR	0.56	0.53*** (21.92)	0.09	0.55	0.53*** (39.76)	0.12	0.54	0.54*** (61.31)	0.13
PETKM	0.57	0.54*** (28.3)	0.16	0.56	0.54*** (38.62)	0.18	0.55	0.55*** (55.05)	0.19
SAHOL	0.57	0.53*** (27.4)	0.11	0.55	0.54*** (44.27)	0.13	0.55	0.54*** (63.39)	0.14
SISE	0.58	0.54*** (37.34)	0.20	0.56	0.55*** (54.93)	0.22	0.56	0.55*** (91.26)	0.24
TAVHL	0.57	0.53*** (24.8)	0.10	0.55	0.53*** (41.13)	0.12	0.55	0.54*** (67.62)	0.14
TCELL	0.56	0.52*** (20.81)	0.09	0.55	0.53*** (38.17)	0.11	0.54	0.53*** (64.08)	0.12
THYAO	0.56	0.52*** (18.14)	0.07	0.55	0.53*** (29.88)	0.08	0.54	0.53*** (44.6)	0.09
TKFEN	0.57	0.53*** (28.27)	0.12	0.55	0.54*** (44.11)	0.14	0.55	0.54*** (69.25)	0.15
TOASO	0.57	0.53*** (22.93)	0.10	0.55	0.53*** (34.64)	0.11	0.54	0.53*** (59.67)	0.13
TTKOM	0.57	0.53*** (27.03)	0.14	0.56	0.54*** (43.61)	0.17	0.55	0.54*** (63.89)	0.18
TUPRS	0.57	0.53*** (25.88)	0.10	0.56	0.54*** (43.84)	0.12	0.55	0.54*** (72.35)	0.13
ULKER	0.56	0.52*** (17.37)	0.08	0.55	0.53*** (33.02)	0.11	0.54	0.53*** (54.94)	0.13
VAKBN	0.57	0.54*** (31.35)	0.14	0.56	0.54*** (50.44)	0.16	0.55	0.54*** (77.97)	0.18
YKBNK	0.57	0.54*** (30.05)	0.16	0.56	0.54*** (48.27)	0.19	0.55	0.55*** (78.31)	0.21
Mean	0.57	0.53	0.12	0.55	0.54	0.14	0.55	0.54	0.16
Std	0.01	0.01	0.04	0.01	0.01	0.04	0.01	0.01	0.05
Max	0.58	0.55	0.24	0.57	0.56	0.27	0.56	0.56	0.29
Min	0.56	0.52	0.07	0.55	0.53	0.08	0.54	0.53	0.09

Note: The values in the parentheses are t-statistics for the one-sided t-test. ***, ** and * denote significant at the 1%, 5% and 10% level respectively.

Table 17: This table presents the (average) area under the curve measure of classifier performance. The results are given for the sliding window setup with the training period of 20 days and test period of 5 days.

	k-NN	Logistic	Naïve Bayes	Random Forest	SVM	XGBOOST
AKBNK	0.54	0.55	0.54	0.56	0.56	0.55
ARCLK	0.55	0.56	0.54	0.56	0.55	0.56
BIMAS	0.54	0.55	0.53	0.55	0.54	0.55
CCOLA	0.53	0.55	0.53	0.55	0.54	0.54
EKGYO	0.57	0.58	0.57	0.59	0.58	0.58
ENKAI	0.55	0.55	0.54	0.57	0.55	0.56
EREGL	0.55	0.56	0.55	0.57	0.56	0.56
FROTO	0.54	0.55	0.52	0.55	0.54	0.55
GARAN	0.54	0.55	0.54	0.56	0.55	0.55
HALKB	0.55	0.55	0.54	0.56	0.55	0.55
ISCTR	0.55	0.56	0.55	0.57	0.56	0.56
KCHOL	0.53	0.54	0.53	0.55	0.54	0.54
KOZAL	0.54	0.55	0.53	0.55	0.55	0.54
KRDMD	0.55	0.57	0.55	0.57	0.57	0.57
OTKAR	0.54	0.55	0.54	0.56	0.55	0.55
PETKM	0.55	0.56	0.55	0.57	0.57	0.57
SAHOL	0.55	0.55	0.54	0.56	0.56	0.56
SISE	0.56	0.56	0.55	0.58	0.56	0.57
TAVHL	0.55	0.55	0.54	0.56	0.55	0.56
TCELL	0.54	0.55	0.53	0.55	0.55	0.55
THYAO	0.53	0.54	0.53	0.54	0.54	0.53
TKFEN	0.55	0.56	0.54	0.56	0.55	0.56
TOASO	0.54	0.55	0.53	0.55	0.55	0.55
TTKOM	0.55	0.56	0.54	0.57	0.56	0.56
TUPRS	0.54	0.56	0.54	0.56	0.55	0.56
ULKER	0.54	0.54	0.53	0.54	0.54	0.54
VAKBN	0.55	0.56	0.54	0.57	0.56	0.56
YKBNK	0.55	0.56	0.55	0.57	0.56	0.57

Figure 1: This figure presents the ROC curves for different classifiers with the asset in question is EKGYO. The curves are given for the sliding window setup with the training period of 20 days and test period of 5 days. The thick black line corresponds to the average ROC curve over all the shifts.

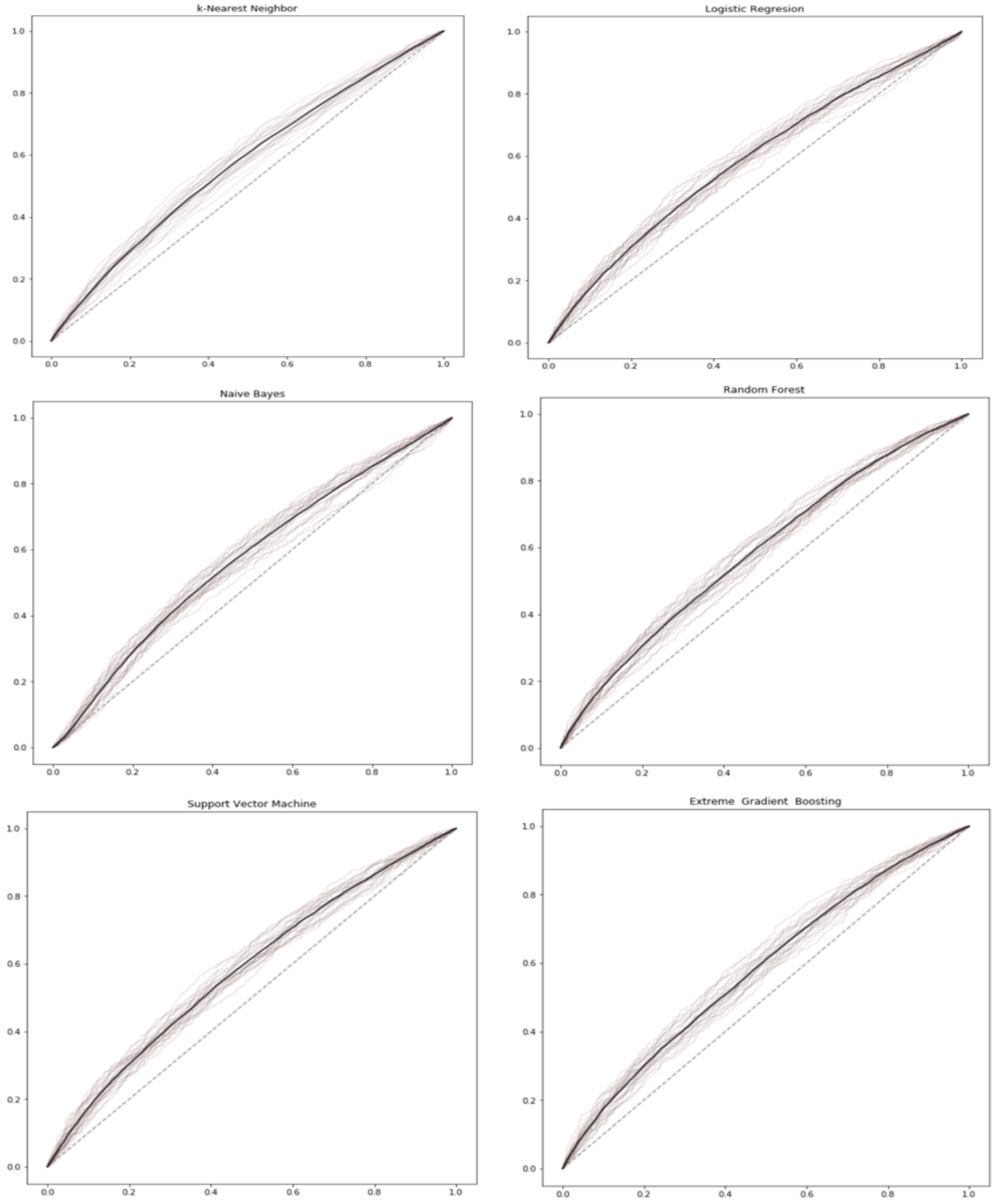
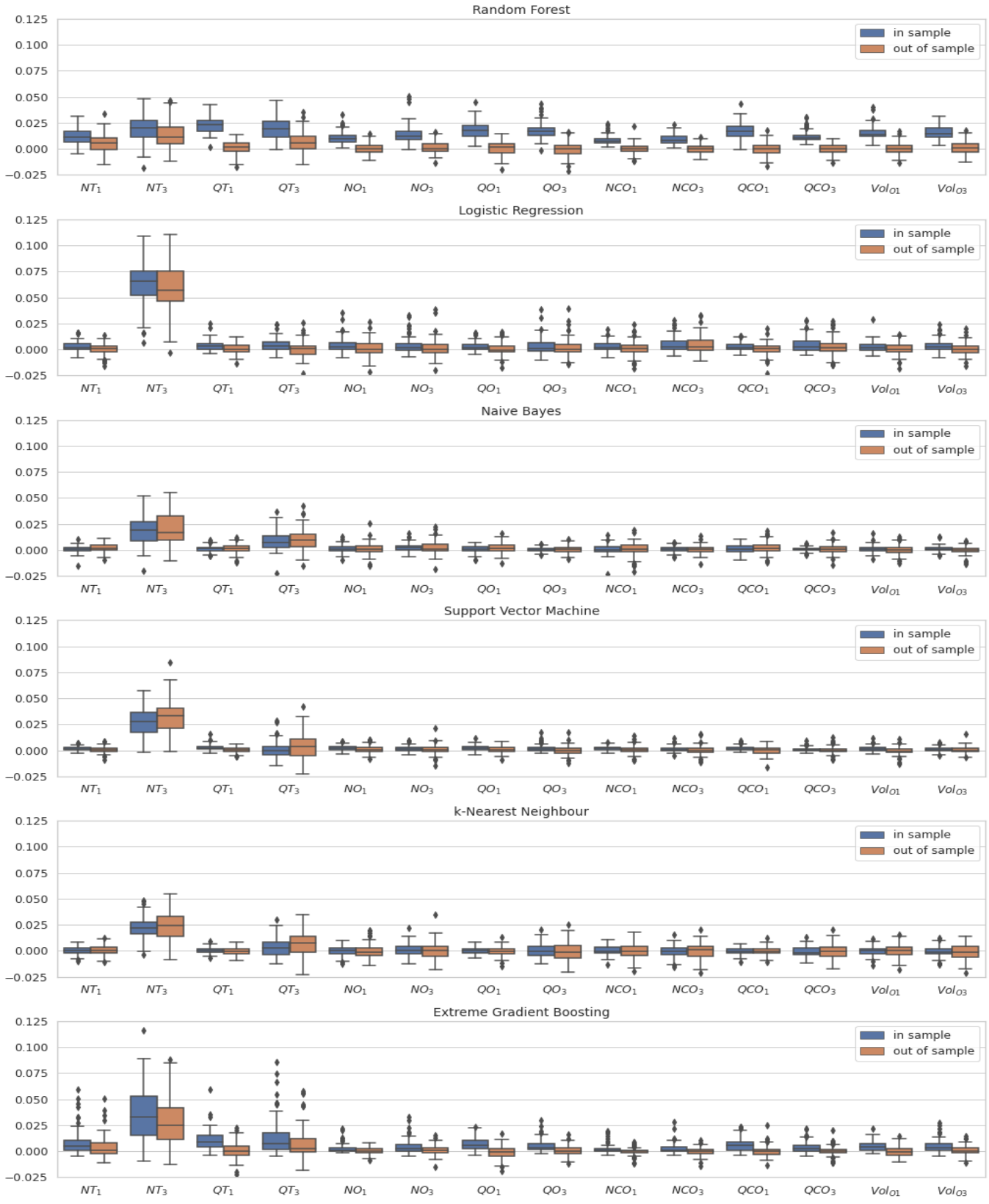


Figure 2: This figure shows factor importance for different classifiers with the asset in question is EKGYO. The box plots are generated for the sliding window setup with the training period of 20 days and test period of 5 days.



Data Accessibility Statement

The data that support the findings of this study are available from Borsa Istanbul. Restrictions apply to the availability of these data, which were used under license for this study. Data are available from Borsa Istanbul Marketing Department with the permission of Borsa Istanbul General Management.

References

Albergaria, M., Jabbour, C.J.C., 2020. The role of big data analytics capabilities (BDAC) in understanding the challenges of service information and operations management in the sharing economy: Evidence of peer effects in libraries. International Journal of Information Management 51, 102023.

Amat, C., Michalski, T., Stoltz, G., 2018. Fundamentals and exchange rate forecastability with simple machine learning methods. Journal of International Money and Finance 88, 1–24.

Amaya, D., Filbien, J.Y., Okou, C., Roch, A.F., 2018. Distilling liquidity costs from limit order books. Journal of Banking and Finance 94, 16–34.

Antweiler, W.and Frank, M.Z., 2004. Is all that talk just noise? The information content of internet stock message boards. Journal of Finance 59, 1259–1294.

Atsalakis, G.S., Atsalaki, I.G., Pasiouras, F., Zopounidis, C., 2019. Bitcoin price forecasting with neuro-fuzzy techniques. European Journal of Operational Research 276, 770–780.

Aydiner, A.S., Tatoglu, E., Bayraktar, E., Zaim, S., Delen, D., 2019. Business analytics and firm performance: The mediating role of business process performance. Journal of Business Research 96, 228–237.

Aziz, S., Dowling, M., Hammami, H., Piepenbrink, A., 2021. Machine learning in finance: A topic modeling approach. European Financial Management (forthcoming).

Bailey, W., Cai, J., Cheung, Y.L., Wang, F., 2009. Stock returns, order imbalances, and commonality: Evidence on individual, institutional, and proprietary investors in China. Journal of Banking and Finance 33, 9–19.

Baker, M., Wurgler, J., 2006. Investor sentiment and the cross-section of stock returns. Journal of Finance 61, 1645–1680.

Begenau, J., Farboodi, M., Veldkamp, L., 2018. Big data in finance and the growth of large firms. Journal of Monetary Economics 97, 71–87.

Bernard, V.L., Thomas, J.K., 1990. Evidence that stock prices do not fully reflect the implications of current earnings for future earnings. Journal of Accounting and Economics 13, 305–340.

Bernile, G., Hu, J., Tang, Y., 2016. Can information be locked up? Informed trading ahead of macro-news announcements. Journal of Financial Economics 121, 496–520.

Beutel, J., List, S., von Schweinitz, G., 2019. Does machine learning help us predict banking crises? Journal of Financial Stability 45, 100693.

Boehmer, E., Jones, C.M., Zhang, X., 2008. Which shorts are informed? Journal of Finance 63, 491–527.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Cai, C.X., Zhang, Q., 2016. High-frequency exchange rate forecasting. European Financial Management 22, 120–141.

Cepni, O., Guney, I.E., Gupta, R., Wohar, M.E., 2020. The role of an aligned investor sentiment index in predicting bond risk premia of the US. Journal of Financial Markets (forthcoming).

Chan, K., Fong, W.M., 2000. Trade size, order imbalance, and the volatility-volume relation. Journal of Financial Economics 57, 247–273.

Chelley-Steeley, P., Lambertides, N., Savva, C.S., 2019. Sentiment, order imbalance, and co-movement: An examination of shocks to retail and institutional trading activity. European Financial Management 25, 116–159.

Chen, H., Chiang, R.H.L., Storey, V.C., 2012. Business intelligence and analytics: From big data to big impact. Management Information Systems Quarterly 36, 1165–1188.

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.

Chordia, T., Roll, R., Subrahmanyam, A., 2002. Order imbalance, liquidity, and market returns. Journal of Financial Economics 65, 111–130.

Chordia, T., Roll, R., Subrahmanyam, A., 2005. Evidence on the speed of convergence to market efficiency. Journal of Financial Economics 76, 271–292.

Chordia, T., Subrahmanyam, A., 2004. Order imbalance and individual stock returns: Theory and evidence. Journal of Financial Economics 72, 485–518.

Chung, D.Y., Hrazdil, K., 2012. Speed of convergence to market efficiency: The role of ECNs. Journal of Empirical Finance 19, 702–720.

Colombo, E., Pelagatti, M., 2020. Statistical learning and exchange rate forecasting. International Journal of Forecasting (forthcoming).

Cont, R., Kukanov, A., Stoikov, S., 2014. The price impact of order book events. Journal of Financial Econometrics 12, 47–88.

Da, Z., Engelberg, J., Gao, P., 2011. In search of attention. Journal of Finance 66, 1461–1499.

Da, Z., Engelberg, J., Gao, P., 2014. The sum of all FEARS: Investor sentiment and asset prices. Review of Financial Studies 28, 1–32.

Diether, K.B., Lee, K.H., Werner, I.M., 2009. Short sale strategies and return predictability. Review of Financial Studies 22, 575–607.

Ding, R., Zhou, H., Li, Y., 2020. Social media, financial reporting opacity, and return comovement: Evidence from Seeking Alpha. Journal of Financial Markets (forthcoming).

Fama, E.F., 1970. Efficient capital markets: A review of theory and empirical work. Journal of Finance 25, 383–417.

Ghasemaghaei, M., 2020. The role of positive and negative valence factors on the impact of bigness of data on big data analytics usage. International Journal of Information Management 50, 395–404.

Gogas, P., Papadimitriou, T., Agrapetidou, A., 2018. Forecasting bank failures and stress testing: A machine learning approach. International Journal of Forecasting 34, 440–455.

Gonzalez, M.R., Basse, T., Saft, T., Kunze, F., 2021. Leading indicators for US house prices: New evidence and implications for EU financial risk managers. European Financial Management (forthcoming).

Gu, S., Kelly, B., Xiu, D., 2020. Empirical asset pricing via machine learning. Review of Financial Studies 33, 2223–2273.

Henrique, B.M., Sobreiro, V.A., Kimura, H., 2019. Literature review: Machine learning techniques applied to financial market prediction. Expert Systems with Applications 124, 226–251.

Hvidkjaer, S., 2008. Small trades and the cross-section of stock returns. Review of Financial Studies 21, 1123–1151.

Jiang, F., Lee, J., Martin, X., Zhou, G., 2019. Manager sentiment and stock returns. Journal of Financial Economics 132, 126–149.

Kearney, F., Shang, H.L., 2020. Uncovering predictability in the evolution of the WTI oil futures curve. European Financial Management 26, 238–257.

Khandani, A.E., Kim, A.J., Lo, A.W., 2010. Consumer credit-risk models via machine-learning algorithms. Journal of Banking and Finance 34, 2767–2787.

Kim, H.Y., Won, C.H., 2018. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. Expert Systems with Applications 103, 25–37.

Klein, O., 2020. Trading aggressiveness and market efficiency. Journal of Financial Markets 47, 100515.

Lakonishok, J., Shleifer, A., Vishny, R.W., 1994. Contrarian investments, extrapolation and risk. Journal of Finance 49, 1541–1578.

Lee, C.M.C., Ready, M.J., 1991. Inferring trade direction from intraday data. Journal of Finance 46, 733–747.

Li, F., 2010. The information content of forward-looking statements in corporate filings - a naïve Bayesian machine learning approach. Journal of Accounting Research 48, 1049–1102.

Li, X., Shang, W., Wang, S., 2019. Text-based crude oil price forecasting: A deep learning approach. International Journal of Forecasting 35, 1548–1560.

Lin, C.S., Khan, H.A., Chang, R.Y., Wang, Y.C., 2008. A new approach to modeling early warning systems for currency crises: Can a machine-learning fuzzy expert system predict the currency crises effectively? Journal of International Money and Finance 27, 1098–1121.

Manyika, J., Bughin, J., Chui, M., Dobbs, R., Brown, B., Roxburgh, C., 2011. Big data: The next frontier for innovation, competition, and productivity. McKinsey & Company.

Ngo, J., Hwang, B.G., Zhang, C., 2020. Factor-based big data and predictive analytics capability assessment tool for the construction industry. Automation in Construction 110, 103042.

Patel, J., Shah, S., Thakkar, P., Kotecha, K., 2015. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Systems with Applications 42, 259–269.

Reinsel, D., Gantz, J., Rydning, J., 2018. Data age 2025 - the digitization of the world from edge to core. International Data Corporation.

Risse, M., 2019. Combining wavelet decomposition with machine learning to forecast gold returns. International Journal of Forecasting 35, 601–615.

Sheng, J., Amankwah-Amoah, J., Wang, X., 2017. A multidisciplinary perspective of big data in management research. International Journal of Production Economics 191, 97–112.

Sivarajah, U., Kamal, M.M., Irani, Z., Weerakkody, V., 2017. Critical analysis of big data challenges and analytical methods. Journal of Business Research 70, 263–286.

Tetlock, P.C., 2007. Giving content to investor sentiment: The role of media in the stock market. Journal of Finance 62, 1139–1168.

Tetlock, P.C., Saar-Tsechansky, M., Macskassy, S., 2008. More than words: Quantifying language to measure firms' fundamentals. Journal of Finance 63, 1437–1467.

Wang, B., Wang, C., 2019. Energy futures prices forecasting by novel DPFWR neural network and DS-CID evaluation. Neurocomputing 338, 1–15.

Wang, G., Gunasekaran, A., Ngai, E.W., Papadopoulos, T., 2016. Big data analytics in logistics and supply chain management: Certain investigations for research and applications. International Journal of Production Economics 176, 98–110.

Wang, J., Athanasopoulos, G., Hyndman, R.J., Wang, S., 2018. Crude oil price forecasting based on internet concern using an extreme learning machine. International Journal of Forecasting 34, 665–677.

Yamamoto, R., 2012. Intraday technical analysis of individual stocks on the Tokyo stock exchange. Journal of Banking and Finance 36, 3033–3047.

Appendix

Table A.1: This table presents the name and stock tickers of the companies under consideration

Ticker	Company Name
AKBNK	AKBANK
ARCLK	ARCELIK
BIMAS	BIM MAGAZALARI
CCOLA	COCA COLA ICECEK
EKGYO	EMLAK KONUT GMYO
ENKAI	ENKA INSAAT
EREGL	EREGLI DEMIR CELIK
FROTO	FORD OTOSAN
GARAN	GARANTI BANKASI
HALKB	TURKIYE HALK BANKASI
ISCTR	IS BANKASI
KCHOL	KOC HOLDING
KOZAL	KOZA ALTIN
KRDMD	KARDEMIR
OTKAR	OTOKAR
PETKM	PETKIM
SAHOL	SABANCI HOLDING
SISE	SISE CAM
TAVHL	TAV HAVALIMANLARI
TCELL	TURKCELL
THYAO	TURK HAVA YOLLARI
TKFEN	TEKFEN HOLDING
TOASO	TOFAS OTO FABRIKALARI
TTKOM	TÜRK TELEKOM
TUPRS	TÜPRAS
ULKER	ULKER BISKUVI
VAKBN	VAKIFLAR BANKASI
YKBNK	YAPI VE KREDI BANKASI

Table A.2: This table presents the list of equity market data analytics

No.	Analytics
A1	Number of orders arriving in the last 60 seconds
A2	Total number of arrived orders up to that time
A3	Quantity of arrived orders in the last 60 seconds
A4	Total quantity of arrived orders up to that time
A5	Number of buy orders arriving in the last 60 seconds
A6	Number of sell orders arriving in the last 60 seconds
A7	Quantity of buy orders arriving in the last 60 seconds
A8	Quantity of sell orders arriving in the last 60 seconds
A9	Number of fill and kill orders arriving in the last 60 seconds
A10	Number of cancelled orders in the last 60 seconds
A11	Quantity of cancelled orders in the last 60 seconds
A12	Number of cancelled buy orders in the last 60 seconds
A13	Number of cancelled sell orders in the last 60 seconds
A14	Quantity of cancelled buy orders in the last 60 seconds
A15	Quantity of cancelled sell orders in the last 60 seconds
A16	Total number of cancelled orders up to that time
A17	Volume weighted average price of cancelled orders up to that time
A18	Volume weighted average price of cancelled buy orders up to that time
A19	Volume weighted average price of cancelled sell orders up to that time
A20	Ratio of the number of cancelled orders to the number of arrived orders in the last 60 seconds
A21	Ratio of the quantity of cancelled orders to the quantity of arrived orders in the last 60 seconds
A22	Ratio of the total number of cancelled orders to the total number of arrived orders up to that time
A23	Ratio of the total quantity of cancelled orders to the total quantity of arrived orders up to that time
A24	Average quantity of buy orders in the last 5 minutes
A25	Average quantity of sell orders in the last 5 minutes
A26	Volatility of buy order quantity in the last 5 minutes
A27	Volatility of sell order quantity in the last 5 minutes
A28	Volume weighted average price (VWAP) of trades in the last 5 minutes
A29	Volume weighted average price (VWAP) of trades up to that time
A30	Volume weighted average price (VWAP) of buyer-initiated trades in the last 5 minutes
A31	Volume weighted average price (VWAP) of seller-initiated trades in the last 5 minutes
A32	Number of buyer-initiated trades in the last 60 seconds
A33	Number of seller-initiated trades in the last 60 seconds
A34	Quantity of buyer-initiated trades in the last 60 seconds
A35	Quantity of seller-initiated trades in the last 60 seconds
A36	Ratio of the number of buyer-initiated trades to the number of seller-initiated trades in the last 60 seconds
A37	Ratio of the quantity of buyer-initiated trades to the quantity of seller-initiated trades in the last 60 seconds
A38	Cumulative ratio of the total number of buyer-initiated trades to the total number of seller-initiated trades
A39	Cumulative ratio of the total quantity of buyer-initiated trades to the total quantity of seller-initiated trades