ISSN: 2658-8455
Volume 2, Issue 3 (May, 2021), pp.390-405.
www.ijafame.org

IJAFAME
International Journal of Accounting, Finance,
Auditing, Management & Economics

Predicting Stock Market Movements Using Machine Learning Techniques

Bilal Elmsili, ***(PhD, Student)***

Laboratory of Economic Analysis and Modeling (LEAM)

Faculty of Law, Economics and Social Sciences Souissi

Mohammed V University of Rabat, Morocco

Benaceur Outtaj, ***(PhD, Professor)***

Laboratory of Economic Analysis and Modeling (LEAM)

Faculty of Law, Economics and Social Sciences Souissi

Mohammed V University of Rabat, Morocco

Correspondence address:	Faculty of Law, Economics and Social Sciences Souissi Al Irfane. BP 6430 Rabat Mohammed V University in Rabat Morocco (Rabat) +212 5 37 67 17 19 bilal_elmsili@um5.ac.ma
Disclosure statement:	The authors are not aware of any funding, that might be perceived as affecting the objectivity of this study.
Conflicts of interest:	The authors reports no conflicts of interest.
Cite this article	Elmsili, B., & Outtaj, B. (2021). Predicting Stock Market Movements Using Machine Learning Techniques. International Journal of Accounting, Finance, Auditing, Management and Economics, 2(3), 390-405. https://doi.org/10.5281/zenodo.4869914
License	**This is an open access article under the CC BY-NC-ND license**

DOI: 10.5281/zenodo.4869914
Received: May 09, 2021 **Published online:** May 30, 2021

ISSN: 2658-8455
Volume 2, Issue 3 (May, 2021), pp.390-405.
www.ijafame.org

IJAFAME
International Journal of Accounting, Finance, Auditing, Management & Economics

Predicting Stock Market Movements Using Machine Learning Techniques

Abstract

The purpose of this paper is to compare the performance of various state-of-the-art machine learning techniques in predicting the behavior of stock-market returns. To do so, we gathered ten years of daily historical data (2488 observations per stock) for the top ten most liquid stocks in Casablanca Stock Exchange (Morocco) and trained six machines learning classifiers (ridge regression, LASSO regression, support-vector machine, k-nearest neighbors, random forest, and adaptive boosting) and an ensemble of them (i.e. ensemble learning) in order to predict one-day-ahead, one-week-ahead, and one-month-ahead prices direction (i.e. positive or negative returns). The performance of each algorithm is then evaluated using accuracy, precision, recall, and F1 scores. Applying the Diebold-Mariano test at a significance level of 5%, we have found that support-vector machine, random forest, and adaptive boosting perform equally well and outperform all other single classifiers for short-term predictions (one-day-ahead and one-week-ahead). However, for monthly predictions, all methods display similar predictive accuracy. In addition, our study suggests that ensemble learning significantly improves all performance metrics for the three prediction horizons. We have also found that for all models the performance significantly decreases as the prediction horizon increases.

Keywords: Machine Learning, Stock Returns Prediction, Efficient Market Hypothesis, Technical Analysis, Casablanca Stock Exchange

JEL Classification: G17

Paper type: Empirical research

1. Introduction

The predictability of stock market returns is one of the most controversial questions in financial economics. In fact, while economists and statisticians have always been concerned with building and improving models of stock market returns, there is still no consensus whether they are predictable or not.

The idea of the unpredictability of stock market returns is supported by the efficient market hypothesis (EMH). According to this hypothesis, major capital markets are efficient in aggregating all available information instantaneously, i.e. when new information is released, it spreads very quickly and gets incorporated into the prices without delay. As a consequence, it is impossible to predict the price change at time $T+1$ based on the information set available at time T because it was already reflected in the prices at T. The $T+1$ price change will reflect only news released at $T+1$ which are by definition unpredictable, thus $T+1$ price change is unpredictable as well.

The concept of the efficient market was first pointed out by Fama (1965). He observed that stock prices follow a random walk, i.e. successive price changes are independent, identically distributed random variables. According to him, this property of stock prices is consistent with the existence of an efficient market, a market where competition between the many rational participants leads to a situation in which actual prices always fully reflect all available information. Using a taxonomy suggested by Roberts (1967), Fama (1970) distinguished between three forms of market efficiency:

- The weak form efficiency: the information set includes only previous historical prices. Future prices cannot be predicted by analyzing prices from the past;
- The semi-strong form: when prices efficiently adjust to all publically available information (e.g. announcements of annual earnings, stock splits, etc.);
- The strong form: current security prices reflect all information, public and private (e.g. information available only to corporate executives).

The information set of weak form efficiency was later extended by Fama (1991) to include other explanatory variables like dividend yields and interest rates. As a consequence, weak form efficiency tests are expanded to cover the more general area of tests for return predictability.

However, the efficient market hypothesis is not universally accepted. Many researchers have shown that many market anomalies (i.e. inefficiencies) may appear from time to time, which indicates that future stock market returns are at least partially predictable (Green et al., 2013; Jacobs, 2015). Two techniques for predicting stock returns are commonly discussed in the literature: fundamental analysis and chartist or technical theories.

Fundamental analysis attempts to measure the intrinsic value of a stock by examining fundamental factors that affect the earning potential of the firm (e.g. management quality, economy and industry conditions, etc.). The aim is to determine whether the actual price of a stock is bellow or above its intrinsic value. Assuming that the actual price tends to move toward the intrinsic value, then attempting to determine the intrinsic value of a stock is equivalent to predicting its future price (Curtis, 2012).

A radically different approach to predicting stock returns is technical analysis. This approach assumes that recurring patterns can be identified from historical market data (especially prices and volume data). More formally, the technical analysis starts from the premise that successive price changes are dependent (Lo & MacKinlay, 1987). As a consequence, at any point in time, the sequence of historical price changes is important in predicting future price change. Lo, Mamaysky, and Wang (2000) provide more direct support in favor of the usefulness of the technical analysis.

However, in the past, most prediction models that use past price data and other technical and fundamental factors as inputs were based on conventional statistical techniques like linear regression, autoregressive integrated moving average, etc. But, stock prices are noisy, non-stationary, and exhibit non-linear dynamics that cannot be captured by simple linear models (Abu-Mostafa & Atiya, 1996). To address this, numerous machine learning methods have been proposed in order to improve prediction results.

Our study embraces the technical analysis approach. We assume that past stock data contain, to some extent, useful information that could be used to predict future stock returns. Learning from past data is performed using seven state-of-the-art machine learning algorithms: ridge regression, LASSO (least absolute shrinkage and selection operator) regression, support-vector machine (SVM), k-nearest neighbors (KNN), random forest, adaptive boosting (AdaBoost), and ensemble learning. The remainder of this paper is organized as follows. In the next section, we will review some previous works that are related to our study. Data and research methodology are described in section three. Results will be discussed in section four and section five concludes.

2. Related works

Machine learning (ML) is the field of study that gives computers the ability to learn from experience without being explicitly programmed (Samuel, 1959). In recent years, ML-based methods have attracted ever-increasing research interests due to their ability to deliver the state of the art results in a variety of domains like computer vision, natural language understanding, and speech recognition. In financial literature, despite the widespread belief in the efficient market hypothesis, several studies have examined the predictability of stock market returns using some cutting edge supervised ML techniques.

For instance, Krauss, Do, and Huck (2017) compared the performance of three ML algorithms in predicting whether the one-day-ahead return of all S&P 500 index constituents will outperform the market or not. Using about 31 lagged simple returns as independent variables (i.e. input variables) and relying on the profits generated by a trading strategy as an evaluation metric, they found that random forest outperforms both gradient-boosted trees and deep neural networks. In addition, they showed that an equally weighted ensemble of those three algorithms produces better results than any single classifier.

Hsu, Lessmann, Sung, Ma, and Johnson (2016) attempted to contrast the performance of econometric models (AR, ARIMA, and GARCH) with ML methods (SVM and artificial neural networks) in predicting positive/negative returns of 34 financial indices covering both emerging and developed markets. Experimenting with simple price-based covariates (open, high, low, and close prices) and also some technical indicators (simple moving averages, moving average convergence divergence, relative strength index, Williams %R, and accumulation distribution oscillator), they showed that the best ML algorithm (SVM) performs better than the best econometric method (AR) for both one-hour-ahead and one-day-ahead forecasting windows. However, they noted that technical indicators do not offer much advantage over simple price-based covariates.

Qian and Rasheed (2007) investigated the predictability of the Dow Jones Industrial Average index (DJIA index) using three machine learning classifiers. Their study reveals that artificial neural networks slightly outperform k-nearest neighbors and decision trees in terms of accuracy. In addition, they showed that due to the high correlation between the predictions of each pair of classifiers, simple voting and stacking ensemble methods did not improve significantly the results. However, they proposed a consistent voting ensemble that only counts predictions agreed upon by all classifiers. They reported that using this ensemble boosted the accuracy rate by approximately 5 points.

While the majority of the studies relate only to short term forecasting windows (especially daily and weekly predictions), Ballings, Van den Poel, Hespeels, and Gryp (2015) predicted whether the one-year-ahead stock price of 5767 European companies will go up by a predetermined threshold (15%, 25%, and 35%). Using about 80 financial and macroeconomic indicators as input variables and based on the area under the receiver operating characteristic curve (AUC), they reported random forest as the best performing algorithm followed by SVM, kernel factory, adaptive boosting, neural networks, k-nearest neighbors, and finally LASSO regression.

Machine learning models are also widely applied in forecasting stock returns in emerging markets. For example, Patel, Shah, Thakkar, and Kotecha (2015) used SVM, artificial neural networks, random forest, and naïve Bayes classifier in predicting the one-day-ahead price movement of two Indian stocks. Using ten technical indicators as input variables, they found that random forest performs better than the other three methods in terms of accuracy and F1 score.

Similarly, Huang, Yang, and Chuang (2008) explored the predictability of both Korea and Taiwan stock market indices. Using a wrapper approach to select the best subset of features among 23 technical indicators and then training five machine learning algorithms, they found that SVM predictions are more accurate than artificial neural networks, k-nearest neighbors, decision trees, and logistic regression. They also showed that combining the forecasts of single classifiers using a voting ensemble yields a better accuracy rate.

In summary, the available literature shows that the application of machine learning methods in forecasting stock returns provides interesting and plausible results. However, whereas the majority of previous works set out to benchmark the performance of only a few machine learning algorithms, in our study we examine a wider set of models including ridge regression, LASSO regression, SVM, random forest, adaptive boosting, and ensemble learning. Moreover, in almost all previous studies, the authors did not conduct appropriate statistical testing in order to confirm if the reported results are statistically significant or not. On the contrary, we perform two sorts of hypothesis testing in order to assess the statistical significance of our findings. First, we test whether the estimated accuracy of each method is statistically better than random guessing. Second, we also examine if the difference in accuracy observed between two different methods is statistically significant or not. Finally, in contrast to the existing literature where each study generally covers only one prediction window, we investigate how the performance of our models varies as a function of three forecasting horizons: on-day-ahead, one-week-ahead, and one-month-ahead.

3. Methodology

3.1. Data

In this study, we experiment with stocks listed in the Casablanca Stock Exchange (CSE). We chose CSE as a case study because it is considered among the most promising financial markets in Africa. In fact, established in 1929, CSE currently includes 74 listed companies and it is the 2nd largest African stock market in terms of capitalization and 3rd in terms of trading volumes (Casablanca Stock Exchange, 2017).

The data used in our research is daily and spans 10 years, from January 01, 2008 to December 31, 2017, a total of 2488 daily observations per stock. However, among the 74 listed stocks in CSE, we only considered the top 10 most liquid ones (Table 1). The share turnover is used as a measure of stock liquidity. It is calculated by dividing the number of shares traded over a period by the average number of shares outstanding for the same period. The ranking is processed as follow:

- For each stock "*s*" (74 stocks), we calculate the daily share turnover "*ST*";

- Then, for each day "t" in our study period, we calculate the median share turnover (across stocks) "MST";
- Finally, for each stock, we calculate the number of days the share turnover of that stock is greater than the daily median share turnover, then we divide the result by the number of days in our study period (i.e. $n = 2488$ trading days):

$$Liquidity\ Ratio_s = \frac{1}{n} * \sum_{t=1}^{n} \mathbb{1}(ST_{s,t} > MST_t)$$

Also, in order to be able to generate the features space (i.e. lagged values up to 240 trading days earlier. See section 3.2 for more details), in our selection process we only considered stocks with IPO date (initial public offering) at least two years before the beginning of the study period.

TABLE 1. *TOP 10 STOCKS RANKED BY LIQUIDITY*

Stock	IPO Date	Liquidity Ratio
ITISSALAT AL MAGHRIB	2004-12-13	0.9855
ATTIJARIWAFA BANK	1943-08-13	0.9775
BCP	2004-07-06	0.9618
BMCE BANK	1975-06-16	0.9216
MANAGEM	2000-07-11	0.9096
CIH	1967-06-23	0.8810
SONASID	1996-07-02	0.7990
LAFARGEHOLCIM.MAR	1997-02-19	0.7721
AUTO HALL	1941-09-04	0.6957
LESIEUR CRISTAL	1972-12-07	0.6527

Source: *Author's calculation*

3.2. Features and Targets

In this study, we experiment with three different targets (i.e. dependent variables): the sign of future daily, weekly, and monthly returns. The aim is to test whether the performance of each machine learning algorithm varies as a function of the prediction horizon. Formally, for each stock in our basket, we generate the output variable as follow:

$$Target_t = \begin{cases} 1 \ ; & if \quad P_{(t+n)} > P_t \\ 0 \ ; & \quad otherwise \end{cases}$$

Where P_t is the adjusted closing price at time t and $n \in [1, 5, 20]$ for respectively daily, weekly, and monthly prediction horizon. The output variable is either equal to one (i.e. class 1) if the return at $t + n$ is positive or zero otherwise (i.e. class 0).

As a preprocessing step, and before training each model, we down-sampled the training set in order to avoid issues related to unbalanced classes (i.e. ensure that 50% of the training instances belong to class 1 and the remaining 50% belong to class 0).

Several studies have shown that past returns contain information about expected returns (Bondt & Thaler, 1985; Grinblatt & Moskowitz, 2004; Jegadeesh, 1990; Jegadeesh & Titman, 1993). For this reason, and following a similar approach to Krauss et al. (2017), the feature space (i.e. independent variables) of each stock is constructed by calculating various lagged logarithmic returns. Also called rates of change (ROC), they are simple technical indicators that compare today's closing price with the close n days ago:

$$ROC_n = log\left(\frac{P_t}{P_{(t-n)}}\right)$$

$$n \in \{\{1, 2, 3, \dots 18, 19, 20\} \cup \{40, 60, 80, \dots, 200, 220, 240\}\}$$

First, we focus on the previous 20 days (one trading month). Then, we shift to a lower resolution and consider the subsequent 11 trading months. In total, we end up with 31 features covering one trading years.

Features standardization is a common requirement for many ML algorithms. This is generally done by removing the mean and dividing by the standard deviation. However, these two sample statistics are very sensitive to outliers. In such a case, the sample median and interquartile range (the range between the 1st quartile and the 3rd quartile) often lead to better results since they are very robust to outliers. For this reason, and as a preprocessing step, every input variable in our feature space is centered by removing its corresponding median and then scaled according to its interquartile range.

3.3. Models Training

For a more detailed description of each algorithm, we recommend James, Witten, Hastie, and Tibshirani (2013). In this study, all models are implemented using the Python library scikit-learn (Pedregosa et al., 2011). Data cleaning and preprocessing are conducted in R (R Core Team, 2013). The R package reticulate (Allaire et al., 2018) is used as an interface between Python and R computing environments.

3.3.1. Regularized Logistic Regression: Ridge and LASSO

In order to achieve better generalization and thus avoid overfitting, we used two regularized, also called penalized, approaches to logistic regression: ridge and LASSO (least absolute shrinkage and selection operator). The goal of these techniques is to reduce the variance of the model and hence improve the overall prediction performance. Both methods work by adding a penalty term to the standard negative log-likelihood loss in the objective function:

$$Ridge\ cost = NLL + \alpha * \frac{1}{2} * w^T w$$

$$LASSO\ cost = NLL + \alpha * \sum_{j=1}^{p} |w_j|$$

Where w is the vector of model coefficients (without the constant, i.e. we do not penalize the intercept term), p is the number of predictors (i.e. explanatory variables), and NLL is the negative log-likelihood loss. The hyper-parameter α controls the amount of parameters shrinkage. For ridge regression, a higher value of α will result in small coefficients while in LASSO, a large α will lead to a model with sparse coefficients, thus, LASSO is generally used as a features selection algorithm. The best value of α is selected using the cross-validation methodology described in section 3.4.

3.3.2. Support Vector Machine

The fundamental idea behind the support vector machine (SVM) is to map the input data into a high-dimensional feature space using a kernel function and then find the optimal hyperplane that maximizes the margin between classes. The vectors (cases) that define the hyperplane are called the support vectors. SVM is based on Vapnik's structural risk minimization principle (Vapnik, 2000) which reduces empirical risk based on bounds of generalization error instead of the empirical error as in other classifiers. In this study, the SVM algorithm is trained with a radial basis function (RBF) kernel. In addition, we follow a cross-validation methodology in order to choose the optimal values for the kernel parameter $gamma$ and the error penalty parameter C.

3.3.3. K-nearest Neighbors

The k-nearest neighbors (KNN) is a non-parametric, instance-based learning algorithm. Given a new observation x_0, it uses a distance calculation function (e.g. Euclidean distance, Manhattan distance, etc.) in order to identify the K points (denoted N_0) in the training dataset that are closest (i.e. neighbors) to x_0. It then predicts the class label of this new instance as the most common class among N_0 (i.e. majority vote). The majority is either calculated using uniform weights (all points in N_0 are weighted equally) or weights which are inversely proportional to points distance (closest neighbors will have a greater influence than further neighbors). In this study, we used the Euclidean distance metric. Also, for efficiency reason, we used the k-d tree (Bentley, 1975) implementation instead of the default brute force search. The number of neighbors K and the voting strategy (uniform or distance-based) are left as hyper-parameters to be tuned by cross-validation.

3.3.4. Random Forest

Introduced by Tin Kam Ho (1995), the Random Forest algorithm operates by constructing a multitude of de-correlated decision trees. Each decision tree uses a different bootstrapped training sample and only a subset of the features space. Decisions of individual trees are then aggregated using a majority vote rule in order to generate the final classification output. Random Forest is very robust to overfitting due to the use of an ensemble of de-correlated trees trained on different samples of training data and different subsets of the predictors. There are three fundamental tunable hyper-parameters in the Random Forest algorithm: the number of trees in the forest (i.e. number of single estimators), the maximum depth of each tree, and the maximum number of features to consider when looking for the best split. In this study, the quality of a split is measured using the GINI criterion.

3.3.5. Adaptive Boosting

The aim of boosting is to convert a weak learning algorithm into one that achieves arbitrarily high accuracy. Adaptive Boosting (Freund & Schapire, 1997) implements this idea by sequentially applying a learning algorithm to reweighted versions of the training data. In each boosting round, the instances that were misclassified during the previous iteration are assigned more weights; as a result, the classifier will focus on examples that have been hard to classify previously. Predictions from the series of all weak learners are then combined through a weighted majority vote to produce the final prediction. In our study, we used a simple decision tree as a base weak learner. To prevent overfitting, the maximum depth of our tree is limited to 3 and it is only allowed to consider 50% of the predictors in each split. The optimal number of boosting iterations is left as a hyper-parameter to be tuned using cross-validation.

3.3.6. Ensemble Learning

In addition to ridge regression, LASSO regression, support-vector machine, k-nearest neighbors, random forest, and adaptive boosting, we used a simple ensemble of them. By an ensemble, we mean a set of classifiers whose individual decisions are typically combined using weighted or un-weighted voting. In order for an ensemble to outperform individual classifiers, these classifiers should be accurate (i.e. perform better than random guessing) and diverse (i.e. make un-correlated errors). However, in financial forecasting, it is very common to observe highly correlated predictions between classifiers. For this reason, we use a consistent voting ensemble that only counts predictions agreed upon by all classifiers (i.e. the ensemble makes a prediction if and only if all classifiers in that ensemble output the same decision). This approach is shown to perform better than other traditional ensemble learning techniques especially in financial applications (Qian & Rasheed, 2007), but it presents the disadvantage of ignoring instances where individual classifiers output inconsistent results.

3.4. Models Validation and Testing

We used the first nine years (2008-2016) of data as a training/validation set. The last year (2017) is kept as a holdout set for out-of-sample models testing and comparison. In this study, we follow a multi-task learning approach similar to Krauss et al. (2017) and Ballings et al. (2015). So, instead of training a separate model for each stock (e.g. ten SVM models, one model for each stock in our basket), we combined the data of the ten stocks into one single set and then trained each model to predict future returns direction for all the ten stocks. In total, we have 17568 observations (corresponding to nine years of daily observations for our ten stocks) in our training/validation set and 2500 instances in the test set.

The performance of each algorithm is evaluated using the precision, recall, F1 score, and accuracy. Precision measures the proportion of positive predictions that were actually correct. However, recall reports the proportion of actual positive predictions that were identified correctly. F1 score is simply the harmonic mean of precision and recall. It reaches its best value at 1 ($precision = recall = 1$) and worst value at 0. Finally, the accuracy score measures the percentage of correct predictions:

$$Precision = \frac{True\ Positives}{Number\ of\ predicted\ positive}$$

$$Recall = \frac{True\ Positives}{Number\ of\ actual\ positive}$$

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$Accuracy(y, \hat{y}) = \frac{1}{n_{samples}} * \sum_{i=0}^{n_{samples}-1} \mathbb{1}(y_i = \hat{y}_i)$$

For hyper-parameters tuning, we performed 6-fold cross-validation using a forward chaining approach. Also called rolling-origin evaluation, this technique is specially designed to avoid many biases related to time series modeling (e.g. the look-ahead bias). Starting from 2011 until 2016, we successively consider each year as the validation set and assign all previous data into the training set (Table 2). The median F1 score (across the six folds) is then used to select the best combination of hyper-parameters (i.e. grid search) for the model in question. As an example, the k-nearest neighbors' algorithm requires two hyper-parameters to be tuned: the number of neighbors (n) and the voting strategy (uniform or distance-based). If we consider 10 possible values for n, the grid will contain 20 possible combinations. For each possible combination, we run our 6-fold rolling-origin cross-validation, report the median F1 score, and then choose the combination that maximizes the median F1 score. With the best combination of hyper-parameters in hand, we re-train our model in the whole training set (2008-2016) and report the results in the test set (2017). Cross-validation results are shown in Table 3.

TABLE 2. *OUR 6-FOLD ROLLING ORIGIN CROSS-VALIDATION SCHEMA*

Folds	Training	Validation
Fold 1	2008-01-01 / 2010-12-31	2011-01-01 / 2011-12-31
Fold 2	2008-01-01 / 2011-12-31	2012-01-01 / 2012-12-31
Fold 3	2008-01-01 / 2012-12-31	2013-01-01 / 2013-12-31
Fold 4	2008-01-01 / 2013-12-31	2014-01-01 / 2014-12-31
Fold 5	2008-01-01 / 2014-12-31	2015-01-01 / 2015-12-31
Fold 6	2008-01-01 / 2015-12-31	2016-01-01 / 2016-12-31

Source: *Author's calculation*

TABLE 3. *CROSS-VALIDATION RESULTS*

	Hyper-parameters	Daily	Weekly	Monthly
Ridge Regression	Regularization C [a]	0.005	0.2	1
LASSO Regression	Regularization C [a]	0.04	0.005	5
SVM	Penalty C	0.10	0.04	50
	Kernel Coefficient Gamma	1/31	1/31	1/31
KNN	Number of Neighbors	25	75	5
	Weighting	distance	uniform	distance
Random Forest	Number of Trees	85	55	10
	Max. Depth	10	10	10
	Max. Features	25%	25%	15%
AdaBoost	Boosting rounds	200	250	300

[a.] The inverse of the regularization strength α. Smaller values specify stronger regularization.

Source: *Author's construction*

4. Results & Discussion

A total of 2500 out-of-sample forecasts (for the ten stocks in our portfolio) are made for the test period from January 01, 2017 to December 31, 2017. Results for one-day-ahead, one-week-ahead, and one-month-ahead forecasting are reported respectively in panel A, B, and C of Table 4.

For the three experiments, and in accordance with the existing literature, all methods achieved an accuracy rate greater than 50% suggesting that they performed at least better than random guessing. In order to check if this conclusion is statistically significant or not, we evaluate the null hypothesis that the forecasts of method i have inferior or equal accuracy than random guessing. The alternative hypothesis indicates that the accuracy of method i is strictly greater than 50%:

$$\begin{cases} H_0: Accuracy\ of\ method\ i \leq 50\% \\ H_1: Accuracy\ of\ method\ i > 50\% \end{cases}$$

Assuming that the accuracy of method i is indeed 50%, we could model the number of correct forecasts (i.e. successes) as a binomial distribution with parameters $n = 2500$ (i.e. the number of out-of-sample forecasts) and $p = 0.5$. Under this assumption, we could now compute the probability of achieving more than the observed accuracy of each method (i.e. one-tailed binomial test). These probabilities are reported in the p-value column of table 4. For the three forecasting horizons, and for all methods, we obtained very small p-values. At a significance level of 1%, we can actually reject the null hypothesis and accept that the accuracy of each method is strictly greater than random guessing.

Table 4 also suggests that SVM outperforms all other single classifiers (i.e. ridge regression, LASSO regression, k-nearest neighbors, random forest, and adaptive boosting) in terms of accuracy for both 1-day-ahead and 5-day-ahead forecasting (59% and 57% respectively). However, for long-term forecasting (i.e. one-month-ahead horizon) we observe that tree-based methods (i.e. random forest and adaptive boosting) are generally more accurate (approximately 54%). In terms of F1 score (i.e. the harmonic average of precision and recall), the best value is achieved by adaptive boosting for 1-day-ahead forecasting (0.537), SVM for 5-day-ahead forecasting (0.569), and LASSO regression for 20-day-ahead forecasting (0.566).

These results are in accordance with Huang et al. (2008) and Hsu et al. (2016) who have also found that SVM forecasts are more accurate than the other methods for one-day-ahead predictions. In contrast, Patel et al. (2015) reported that random forest outperforms SVM in predicting the next day movement of two financial indices. For long-term predictions, similar to our findings, Ballings et al. (2015) have also reported that random forest performs better than SVM, adaptive boosting, k-nearest neighbors, and LASSO regression.

However, in order to investigate if the difference in accuracy observed between two methods is statistically significant or not, we performed the Diebold-Mariano test (Diebold & Mariano, 1995) to evaluate the null hypothesis that the forecasting accuracy of method i is less than or equal the accuracy of method j $(i \neq j)$. The alternative hypothesis suggests that the accuracy of method i is strictly greater than the accuracy of method j:

$$\begin{cases} H_0: Accuracy\ of\ method\ i \leq Accuracy\ of\ method\ j \\ H_1: Accuracy\ of\ method\ i > Accuracy\ of\ method\ j \end{cases}$$

Results of the Diebold-Mariano test are reported in Table 5. In panel A (1-day-ahead horizon), for the null hypothesis that SVM accuracy is less than or equal the accuracy of ridge regression, LASSO regression, k-nearest neighbors, random forest, and adaptive boosting, we obtained p-values of 0.000001, 0.000002, 0.000284, 0.220557, and 0.094609 respectively. As a result, at a significance level of 0.05, we reject the null hypothesis that SVM is less accurate than ridge regression, LASSO regression, and k-nearest neighbors. In contrast, we fail to reject this hypothesis when the accuracy of SVM is compared with the accuracy of random forest and adaptive boosting or vice versa (i.e. these three methods seem to exhibit similar accuracy). Similarly, we reject the null hypothesis that the accuracy of both random forest and adaptive boosting is less than or equal the accuracy of ridge regression, LASSO regression, and k-nearest neighbors. Thus, for 1-day-ahead forecasting, we can conclude that SVM, random forest, and adaptive boosting perform equally well (i.e. no statistically significant difference in accuracy), but they all outperform ridge regression, LASSO regression, and k-nearest neighbors. Table 4 also suggests that there is no statistically significant difference in accuracy between ridge regression, LASSO regression, and k-nearest neighbors.

For 5-day-ahead forecasting (panel B of Table 4), at a significance level of 5%, we reject the null hypothesis that SVM is less accurate than LASSO regression and k-nearest neighbors (p-values of 0.037174 and 0.003994 respectively). We also reject the null and conclude (i.e. accept the alternative hypothesis) that random forest and adaptive boosting are more accurate than k-nearest neighbors in forecasting the direction of next-week prices (p-values of 0.020816 and 0.012355 respectively). However, for 20-day-ahead forecasting (panel C of Table 4), we fail to reject the null hypothesis for every possible combination of methods i and j $(i \neq j)$. This suggests that for the next-month forecasting experiment all used methods display similar predictive accuracy.

Similarly to Krauss et al. (2017), Qian and Rasheed (2007), and Huang et al. (2008), Table 4 also reveals that the ensemble learner widely outperforms all single classifiers in terms of precision, recall, F1 score, and accuracy. In fact, for the one-day-ahead forecasting horizon, the accuracy of the ensemble is greater than the accuracy of the best performing single classifier (i.e. SVM) by approximately 3 points (62% vs. 59%). A Similar difference in accuracy is observed for one-week-ahead forecasting. However, for the one-month-ahead horizon, the difference is more important and reaches about 5 points (the ensemble accuracy is about 58% vs. only 53% for random forest, adaptive boosting, and SVM). Concerning F1 score, the difference is about 5 points for one-day-ahead forecasting (59% vs. 54% for adaptive boosting), 7 points for 5-day-ahead forecasting (64% vs. 57% for SVM), and 8 points for 20-day-ahead forecasting (64% vs. 56% for LASSO regression). However, since we have used a consistent ensemble that makes a forecast if and only if all other single classifiers output the same prediction for a given day, the number of forecasts generated by this ensemble represents only about 42% (about 1050 forecasts) of the number of instances in our out-of-sample dataset for the one-day-ahead experiment, 44% (about 1100 forecasts) for one-week-ahead forecasting, and only 24% (about 600 forecasts) for the one-month-ahead horizon.

Table 4 indicates also a clear connection between the accuracy of each model and the prediction horizon. The accuracy of the seven ML-based algorithms decreases significantly as the prediction horizon increases. For example, the predictive accuracy of the ensemble learner decreases from 62% for one-day-ahead forecasting to 60% for one-week-ahead forecasting and then to only 58% for one-month-ahead forecasting. A similar pattern was identified by (Hsu et al., 2016). They found evidence that the forecast horizon affects the predictive accuracy as well as the profitability of a model-based trading system.

TABLE 4. *FORECASTING RESULTS*

	Precision	**Recall**	**F1 Score**	**Accuracy**	**P-value**
Panel A. **1-day-ahead horizon**					
Ridge Regression	0.45209	0.56312	0.50154	0.5460	0.0000023
LASSO Regression	0.45489	0.58185	0.51060	0.5476	0.0000011
SVM	0.50044	0.56608	0.53124	0.5948	0.0000000
K-nearest neighbors	0.46289	0.59665	0.52133	0.5556	0.0000000
Random Forest	0.49405	0.57298	0.53059	0.5888	0.0000000
Adaptive Boosting	0.48832	0.59763	0.53747	0.5828	0.0000000
Consistent Ensemble	**0.52381**	**0.68750**	**0.59459**	**0.62644**	0.0000000
Panel B. **5-day-ahead horizon**					
Ridge Regression	0.50469	0.62023	0.55653	0.55440	0.0000000
LASSO Regression	0.50306	0.58385	0.54045	0.55240	0.0000001
SVM	0.51901	0.62999	0.56914	0.57000	0.0000000
K-nearest neighbors	0.49342	0.63177	0.55409	0.54160	0.0000172
Random Forest	0.51341	0.57764	0.54363	0.56280	0.0000000
Adaptive Boosting	0.51613	0.58208	0.54712	0.56560	0.0000000
Consistent Ensemble	**0.57973**	**0.72171**	**0.64298**	**0.60620**	0.0000000
Panel C. **20-day-ahead horizon**					
Ridge Regression	0.50593	0.63947	0.56491	0.52680	0.0039008
LASSO Regression	0.50659	0.64030	0.56565	0.52760	0.0030664
SVM	0.51243	0.58368	0.54574	0.53320	0.0004816

K-nearest neighbors	0.50260	0.56370	0.53140	0.52240	0.0131995
Random Forest	0.51783	0.58035	0.54731	0.53880	0.0000563
Adaptive Boosting	0.51533	0.57369	0.54295	0.53600	0.0001709
Consistent Ensemble	**0.57474**	**0.73355**	**0.64451**	**0.58376**	0.0000267

For each method, we report the precision, recall, F1 score, and accuracy. We also evaluate the null hypothesis that the accuracy of method i is less than or equal the accuracy of random guessing (i.e. the hypothesis that method i has randomly achieved these results). The alternative hypothesis is that the accuracy of method i is strictly greater than random guessing (i.e. 50% accuracy).

Source: *Author's calculation*

TABLE 5. DIEBOLD-MARIANO TEST RESULTS

For each pair of methods, we report the p-value corresponding to the null hypothesis that the forecasting accuracy of method i (horizontal axis) is less than or equal the accuracy of method j (vertical axis). The alternative hypothesis is that the accuracy of method i is strictly greater than the accuracy of method j. P-values that yield to reject the null hypothesis at 5% significance level are underlined and in bold.

	Ridge Regression	LASSO Regression	SVM	K-nearest neighbors	Random Forest	Adaptive Boosting
Panel A. **1-day-ahead horizon**						
Ridge Regression	−	0.605798	0.999999	0.776686	0.999985	0.999754
LASSO Regression	0.394202	−	0.999998	0.736195	0.999970	0.999549
SVM	**0.000001**	**0.000002**	−	**0.000284**	0.220557	0.094609
K-nearest neighbors	0.223314	0.263805	0.999716	−	0.998103	0.989738
Random Forest	**0.000015**	**0.000030**	0.779443	**0.001897**	−	0.213686
Adaptive Boosting	**0.000246**	**0.000451**	0.905391	**0.010262**	0.786314	−
Panel B. **5-day-ahead horizon**						
Ridge Regression	−	0.414766	0.945065	0.131935	0.789115	0.868793
LASSO Regression	0.585234	−	0.962826	0.183445	0.846415	0.893005
SVM	0.054935	**0.037174**	−	**0.003994**	0.202241	0.309129
K-nearest neighbors	0.868065	0.816555	0.996006	−	0.979184	0.987645
Random Forest	0.210885	0.153585	0.797759	**0.020816**	−	0.637288
Adaptive Boosting	0.131207	0.106995	0.690871	**0.012355**	0.362712	−
Panel C. **20-day-ahead horizon**						
Ridge Regression	−	0.921330	0.696788	0.375199	0.833238	0.783111
LASSO Regression	0.078670	−	0.674096	0.353358	0.816371	0.762338
SVM	0.303212	0.325904	−	0.199574	0.684008	0.597290

K-nearest neighbors	0.624801	0.646642	0.800426	–	0.889801	0.840670
Random Forest	0.166762	0.183629	0.315992	0.110199	–	0.388191
Adaptive Boosting	0.216889	0.237662	0.402710	0.159330	0.611809	–

Source: *Author's calculation*

5. Conclusion

This study set out to compare the performance of seven machine learning algorithms (ridge regression, LASSO regression, support-vector machine, k-nearest neighbors, random forest, adaptive boosting, and ensemble learning) in predicting stock prices direction (i.e. positive/negative stock returns). To the best of our knowledge, this is the first study to apply this broad set of ML-based models on stocks listed in Casablanca Stock Exchange (Morocco).

The contribution of our study to the available literature is threefold. First, using Diebold-Mariano test at a significance level of 5%, we have found that for short-term predictions (i.e. one-day-ahead and five-day-ahead) support-vector machine, random forest, and adaptive boosting perform equally well (i.e. no statistically significant difference in accuracy) and in general, they outperform ridge regression, LASSO regression, and k-nearest neighbors. However, for one-month-ahead forecasting, all the six single classifiers display similar predictive accuracy. Second, our results show that the consistent ensemble improves significantly all performance metrics (precision, recall, F1 score, and accuracy) for the three forecasting horizons. Finally, the obtained results indicate a strong relationship between the prediction horizon and the accuracy of our models: the accuracy decreases considerably when the prediction horizon increases.

However, our study could be extended in many ways. First, our research is based only on data from Casablanca Stock Exchange; it might be valuable to benchmark with other markets and test, for example, if the market maturity affects the predictive performance of each algorithm. Second, it would be extremely useful to also include deep learning-based approaches. For example, Fischer and Krauss (2018) found that a recurrent neural network with a long-short-term memory (LSTM) performs better than random forest and logistic regression. Finally, our focus was on predicting the sign of stock returns (i.e. classification problem). Another aim might be to predict the exact stock returns (i.e. regression problem).

References:

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

(2) Allaire, J. J., Ushey, K., & Tang, Y. (2018). reticulate: Interface to "Python." https://CRAN.R-project.org/package=reticulate

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

(4) Bentley, J. L. (1975). Multidimensional Binary Search Trees Used for Associative Searching. Commun. ACM, 18(9), 509–517. https://doi.org/10.1145/361002.361007

(5) Bondt, W. F. M. D., & Thaler, R. (1985). Does the Stock Market Overreact? The Journal of Finance, 40(3), 793–805. https://doi.org/10.2307/2327804

(6) Casablanca Stock Exchange. (2017). Casablanca Stock Exchange Annual Report. http://www.casablanca-bourse.com/bourseweb/en/content.aspx?IdLink=206

(7) Curtis, A. (2012). A Fundamental-Analysis-Based Test for Speculative Prices. The Accounting Review, 87(1), 121–148. JSTOR.

(8) Diebold, F. X., & Mariano, R. S. (1995). Comparing Predictive Accuracy. Journal of Business & Economic Statistics, 13(3), 253–263. https://doi.org/10.1080/07350015.1995.10524599

(9) Fama, E. F. (1965). The Behavior of Stock-Market Prices. The Journal of Business, 38(1), 34–105.

(10) Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. The Journal of Finance, 25(2), 383–417. https://doi.org/10.2307/2325486

(11) Fama, E. F. (1991). Efficient Capital Markets: II. The Journal of Finance, 46(5), 1575–1617. https://doi.org/10.2307/2328565

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

(13) Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences, 55(1), 119–139. https://doi.org/10.1006/jcss.1997.1504

(14) Green, J., Hand, J. R. M., & Zhang, X. F. (2013). The supraview of return predictive signals. Review of Accounting Studies, 18(3), 692–730. https://doi.org/10.1007/s11142-013-9231-1

(15) Grinblatt, M., & Moskowitz, T. J. (2004). Predicting stock price movements from past returns: The role of consistency and tax-loss selling. Journal of Financial Economics, 71(3), 541–579. https://doi.org/10.1016/S0304-405X(03)00176-4

(16) Ho, T. K. (1995). Random Decision Forests. Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1, 278-. http://dl.acm.org/citation.cfm?id=844379.844681

(17) Hsu, M.-W., Lessmann, S., Sung, M.-C., Ma, T., & Johnson, J. E. V. (2016). Bridging the divide in financial market forecasting: Machine learners vs. financial economists. Expert Systems with Applications, 61, 215–234. https://doi.org/10.1016/j.eswa.2016.05.033

(18) Huang, C.-J., Yang, D.-X., & Chuang, Y.-T. (2008). Application of wrapper approach and composite classifier to the stock trend prediction. Expert Systems with Applications, 34(4), 2870–2878. https://doi.org/10.1016/j.eswa.2007.05.035

(19) Jacobs, H. (2015). What explains the dynamics of 100 anomalies? Journal of Banking & Finance, 57, 65–85. https://doi.org/10.1016/j.jbankfin.2015.03.006

(20) James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: With Applications in R. Springer-Verlag. //www.springer.com/gp/book/9781461471370

(21) Jegadeesh, N. (1990). Evidence of Predictable Behavior of Security Returns. The Journal of Finance, 45(3), 881–898. JSTOR. https://doi.org/10.2307/2328797

(22) Jegadeesh, N., & Titman, S. (1993). Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency. The Journal of Finance, 48(1), 65–91. https://doi.org/10.2307/2328882

(23) Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. European Journal of Operational Research, 259(2), 689–702. https://doi.org/10.1016/j.ejor.2016.10.031

(24) Lo, A. W., & MacKinlay, A. C. (1987). Stock Market Prices Do Not Follow Random Walks: Evidence From a Simple Specification Test (Working Paper No. 2168). National Bureau of Economic Research. https://doi.org/10.3386/w2168

(25) Lo, A. W., Mamaysky, H., & Wang, J. (2000). Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation. The Journal of Finance, 55(4), 1705–1765. https://doi.org/10.1111/0022-1082.00265

(26) 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(1), 259–268. https://doi.org/10.1016/j.eswa.2014.07.040

(27) Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825−2830.

(28) Qian, B., & Rasheed, K. (2007). Stock market prediction with multiple classifiers. Applied Intelligence, 26(1), 25–33. https://doi.org/10.1007/s10489-006-0001-7

(29) R Core Team. (2013). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. http://www.R-project.org/

(30) Roberts, H. (1967). Statistical versus Clinical Prediction of the Stock Market.

(31) Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development, 3(3), 210–229. https://doi.org/10.1147/rd.33.0210

(32) Vapnik, V. (2000). The Nature of Statistical Learning Theory (2nd ed.). Springer-Verlag. //www.springer.com/gp/book/9780387987804