A General Framework for Risk Controlled Trading Based on Machine Learning and Statistical Arbitrage

Salvatore Carta[0000−0001−9481−511X], Diego Reforgiato Recupero[0000−0001−8646−6183], Maria Madalina Stanciu[0000−0002−6522−908X], and Roberto Saia[0000−0002−1734−0437]

Department of Mathematics and Computer Science, University of Cagliari, Italy
{salvatore,diego.reforgiato,roberto.saia,madalina.stanciu}@unica.it

Abstract. Nowadays, machine learning usage has gained significant interest in financial time series prediction, hence being a promise land for financial applications such as algorithmic trading. In this setting, this paper proposes a general framework based on an ensemble of regression algorithms and dynamic asset selection applied to the well known statistical arbitrage trading strategy. Several extremely heterogeneous state-of-the-art machine learning algorithms, exploiting different feature selection processes in input, are used as base components of the ensemble, which is in charge to forecast the return of each of the considered stocks. Before being used as an input to the arbitrage mechanism, the final ranking of the assets takes also into account a quality assurance mechanism that prunes the stocks with poor forecasting accuracy in the previous periods. The framework has a general application for any risk balanced trading strategy aiming to exploit different financial assets. It was evaluated implementing an intra-day trading statistical arbitrage on the stocks of the S&P500 index. Our approach outperforms each single base regressor we adopted, which we considered as baselines. More important, it also outperforms Buy-and-hold of S&P500 Index, both during financial turmoil such as the global financial crisis, and also during the massive market growth in the recent years.

Keywords: Stock Market Forecast · Machine Learning · Statistical Arbitrage · Ensemble learning.

1 Introduction

In financial investing, the general goal is to dynamically allocate a set of assets to maximize the returns over time and minimize risk simultaneously. A very well-known financial trading strategies is statistical arbitrage, or StatArb for short, which evolved out of pairs trading strategy [12], where stocks are paired based on fundamental or market similarities [18]. In pairs intra-day trading, when one stock of the pair under-performs the other, the stock is sold short with the expectation that its price will *drop* when the positions are closed. Similarly, the out-performer is bought with the expectation that its price will *climb* when positions are closed. The same concept applies to the StatArb strategy, except that it extends at portfolio level with more stocks [33]. Furthermore, the portfolio construction is automated and comprises two phases: (i) the *scoring* phase, where each stock is assigned to a relevance score, with high scores indicating stocks that should be held long and low scores indicating stocks that are candidates for short operations; and (ii) the *risk reduction* phase, where the stocks are combined to eliminate, or at least significantly reduce the risk factor [4, 27].

The most important challenges the financial investors using StatArb strategy are exposed to consist of determining pairs of stocks that exhibit a relationship, a balance point between them, and determining the point in time in which prices move sufficiently away from that balance. As such, researchers have expended unremitting efforts on investigating novel approaches to tackle the asset choice problem and developed a wide range of *statistical tools* for the matter: distance based [18], co-integration approach [39], and models based on stochastic spread [25]. As previously noted in the literature [21], these tools exhibit a drawback as they rely solely on statistical relationship of a pair at the price level, and lack forecasting component. Moreover, if a divergence between stocks in a pair is observed, then it is *assumed* that the prices must converge in the future and positions are closed only when the equilibrium is reached, an event that is not accurately determined in time.

At the same time, the rapid growth of market integration yielded massive amounts of data in the finance industry, which promotes the study of advanced data analysis tools. By the same token, considering that StatArb is performed at portfolio level (hence a large number of assets is involved), the strategy needs to be implemented in an automated fashion. As such, cutting-edge analytical techniques and machine learning algorithms use has grown [20]. However, incorporating machine learning algorithms comes with its own set of drawbacks as the financial data contains a large amount of noise, jump and movement, leading to highly non-stationary time series that are thought to be highly unpredictable [34], thus deteriorating the forecasting performances. One successful alternative to mitigate the noise present in the data has already been proven to be ensemble methods. In literature, they demonstrated superior predictive performance compared to individual forecasting algorithms and hence their notorious success in different domains such as credit scoring [9] or sentiment analysis [3]. Furthermore, in literature, it has been proved that the employment of heterogeneous ensembles for forecasting outperforms homogeneous ones [7, 30]. When mentioning the forecasting, there are two different tasks that can be targeted: classification and regression. In literature, we can find several implementations of StatArb that use classification [37, 29] and this has always been proved easier to solve than the regression [36]. Although regression in the context of financial predictions poses more challenges [16, 32], it allows for a more *granular ranking*, without reference to any balance point. As such, in this paper we propose a general framework for risk-controlled trading based on machine learning and StatArb. The framework employs an ensemble of regressors and provides three levels of heterogeneous features:

1. Its components consist of any number of state-of-the-art machine learning and statistical models.
2. We train our models with information pertaining to constituents of financial time series with a diversified feature set, considering not only lagged daily prices return, but also a series of technical indicators.
3. We consider diversified models such as the ones that use as training either data from individual companies or companies in the same industry.

Finally, in our framework, after the assets have been ranked in descending order, we propose the use of a *dynamic asset selection*, which looks at the past and influences the ranking by removing stocks with bad past behavior. Then, the strategy buys (performing long operations) the flop k stocks and sells (performing short operations) the top k stocks.

In this paper, we also propose one possible instance of our framework that has been configured for intra-day operations and on the well-known S&P500 Index. The regressors we have employed for such an instance are the following state-of-the-art machine learning algorithms, Random Forests (RF), Light Gradient Boosted trees (LGB), Support Vector Regressors (SVR), and the widely known statistical model, ARIMA. ARIMA models are known to be robust and efficient for short-term prediction when employed to model economical and financial time series [14, 1] even more than the most popular ANNs techniques [31, 35].

To validate the configuration we have chosen for our instance, we evaluate its performance from both return and risk performance perspectives. The comparisons against Buy-and-Hold strategy of S&P500 Index and individual regressors that we adopted in our instance, lucidly illustrate its superiority in performing the forecast.

In summary, the contributions of this paper are the following:

1. We propose a general framework for risk-controlled trading based on machine learning and StatArb.
2. We defined the problem as a regression of price returns, instead of a classification one.
3. Our framework can be easily implemented using different types of assets.
4. We propose an ensemble methodology for StatArb, tackling the ensemble construction from three different perspectives:
   - *model diversity*, by using machine learning algorithms and even statistical algorithms;
   - *data diversity*, by considering lagged price returns and technical indicators so to enrich the data used by models;
   - *method diversity*, by simultaneously training single models across several assets (*i.e.*, models per industries) and, conversely, models for each stock.
5. We develop a dynamic asset selection based on models' most recent prediction performance that keeps the ranking of an asset if the past predictions of its return trend exceed a pre-determined behavior.
6. We provide a possible instance of our framework for intra-day trading with four kinds of regressors (machine learning algorithms and statistical models) for StatArb within the S&P500 Index.
7. We carried out a performance evaluation of our instance and its results outperform baseline methods on the S&P500 Index for intra-day trading.

The remaining of this paper is organized as it follows. Section 2 briefly describes relevant related work in the literature. Section 3 introduces the problem we are facing whereas Section 4 includes the architecture of the proposed general framework and the instance we have generated. The next sections include details of the adopted instance. In particular, all the features that we have used are described within Section 5. Section 6 details the regressors that we have been considered in the ensemble of our instance. Section 7 describes the proposed ensemble methodology and how we have aggregated the results of the single components. The dynamic asset selection approach is illustrated in Section 8. Section 9 discusses the experiments we have carried out. Finally, Section 10 ends the paper with conclusions and directions where we are headed.

2 Related Work

The literature dealing with applications on machine learning and neural networks in finance is presented and analyzed in several works [20, 10, 2, 8]. There can be distinguished various streams of research and their applications, but this section highlights only a limited number of articles, highly correlated to this paper and to StatArb, as will be discussed in the following. The work in [21] proposes a StatArb system that entails three phases: forecasting, ranking and trading. For the forecasting phase, the authors propose the use of an Elman recurrent neural network to perform weekly predictions and anticipate return spreads between any two securities in the portfolio. Next, a multi-criteria decision-making method is considered to outrank stocks based on their weekly predictions. Lastly, trading signals are generated for top k and bottom k stocks. This work is later extended in [22] by introducing a multi-step-ahead forecast. This approach considers constituents of S&P100 Index on a period spanning from 1992 to 2006. Although these two approaches also consider regression, they are not scalable as their applicability is limited to a few number of stocks, and in case of broader indexes such as S&P500 or Russell 1000, would become computationally intractable. In [37], deep neural networks were used and standardized cumulative returns were considered as features. The approach computes the probability that one stock outperforms the cross-sectional median return of all stocks in the holding month. Next, all stocks are ranked according to the forecasted probability and, then, the trading signals are constructed based on the top decile of predictions, which are bought, and flop decile which are sold short. The stock universe used is the U.S. CRSP and the study period spans from 1965 until 2009. Instead of the classification, one of the challenges of our framework is to tackle the regression. The work in [15] adopts a similar strategy, but in a high-frequency setting with five-minutes binned return data. Following the approach proposed by [37], in [29] the authors construct a similar classification problem using cumulative returns as input features and employ models like deep neural networks, random forests, gradient boosted trees and three of their ensembles. The authors validate their study using $S\&P500$ Index constituents on a period ranging from 1992 to 2015, with trading frequency of one day. Later, the authors extend their work in [17] by using a Long Short-Term Memory network for the same prediction task. This enhanced approach outperforms memory-free classification methods. However, as the authors note, the out-performance is registered from 1992 to 2009, whereas from 2010 the excess return fluctuates around zero. The ensemble proposed in this work is used to tackle a classification problem whereas ours aims at solving a more difficult regression problem. A different approach in terms of features is presented in [23], where the author evaluates whether an increased number of predictors translates to an increased excess returns. The author explores around 600 features on a period from January of 1993 to June 2015, with two forecasting/holding periods (*e.g.* one and five days). The machine learning algorithms used are Random Forests, Elastic Net and Deep belief networks. As in previous works, positive excess returns are reported before 2009 only, and returns turned into negative in the following years. Also, increasing the number of features does not represent a guarantee of increased performance. One difference with respect to this work is that we employ an ensemble strategy to mitigate the results of all the used models. In [28], the authors take a different approach for predicting returns of S&P500, where the used features are stock tweets information. The aim is to unveil how the textual data reflects in stocks' future returns. For this goal, they use factorization matrix and support vector machines. The proposed system performs prediction in a 20 minutes frequency over a two years period: from January 2014 to December 2015. The selection of flop and top stocks is made at the formation period based on the algorithms performance evaluation (*i.e.* lowest root relative squared error) and trading signals are generated based on Bollinger bands. The authors state that their factorization machines approach yields positive results even after transaction costs. In contrast to previously presented studies, in this work we consider the trading performance of an ensemble of diversified regression techniques that considers diverse models and data. Additionally, our approach includes in the pipeline a dynamic asset selection within the *risk reduction* phase, in order to avoid bad past stocks performances that jeopardize future trading. Such a heterogeneous setup is important to deal with the uncertain behavior of the market, as richer models and complementary information are used in the process. Moreover, the proposed approach is a general framework that can be instantiated with a huge number of configurations: number and types of regressors, market type (e.g. intra-day), selected features (e.g. lagged returns, technical indicators), number of assets to buy or sell (choice for k).

3 Problem Formulation

The problem tackled by our general framework consists of an algorithmic trading task in the context of StatArb that leverages machine learning to identify possible sources of profit and balance risk at the same time. The StatArb technique consists of three steps: forecasting, ranking, and trading.

- *Forecasting* - We tackle StatArb as a regression problem, investigating the potential of forecasting price returns for each of the assets in a pre-selected asset collection S, on a target trading day d.
- *Ranking* - Based on the anticipated price returns for the assets, we rank them in descending order. We *balance the risk* incurred by inaccurate predictions by pruning the "bad" assets based on their past behavior. This dynamical asset pruning yields a reorganized ranking of the assets.
- *Trading* - Having the trading desirability given by ranking in the previous stage, we issue trading signals for the top k and flop k stocks.

4 The Proposed Approach

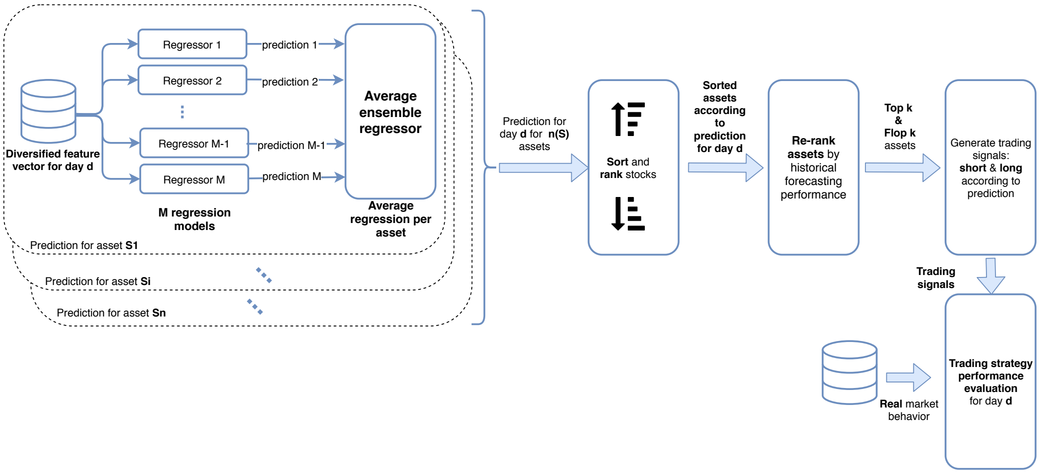
Regressor 1
Regressor 2
Regressor M-1
Regressor M
prediction 1
prediction 2
prediction M-1
prediction M
Average ensemble regressor
Diversified feature vector for day d
M regression models
Average regression per asset
Prediction for asset S1
Prediction for asset Si
Prediction for asset Sn
Prediction for day d for n(S) assets
Sort and rank stocks
Sorted assets according to prediction for day d
Re-rank assets by historical forecasting performance
Top k & Flop k assets
Generate trading signals: short & long according to prediction
Trading signals
Trading strategy performance evaluation for day d
Real market behavior

Fig. 1: Architecture of the proposed general framework for risk controlled trading

Figure 1 depicts the architecture for the general framework for risk controlled trading we propose in this paper. Once the set of assets to work with has been selected, first we collect raw financial information for each asset s_i in the pre-selected asset collection S. We split our raw data in study periods, composed of training (in-sample data, used for training models) and trading (test) sets, which are non-overlapping. This procedure is a well-known validation procedure for time-series data-sets [13], known as walk-forward strategy. Figure 2 illustrates such a procedure. For each study period and each asset s_i, we generate the diversified feature set denoted by $\mathcal{F}_{d-1}^{s_i}$[1]. For in sample period we also generate the label $y_d^{s_i}$. The feature set it used as input to each regressor m in our regressors pool $\mathcal{M}$. The *forecast* is then performed using test data, where each trained model makes its prediction, $o_d^{s_i,m}$ for day d and stock s_i. Then, their results are averaged by a given ensemble method, to obtain a final output $o_d^{s_i,ENS} = \frac{\sum_{m\in\mathcal{M}} o_d^{s,m}}{n(\mathcal{M})}$. Next, we *sort assets* in descending order. That means that we will find at the top assets whose prices are expected to increase, and at the bottom assets whose prices will drop. Assets at the top and at the bottom of our sorting represent the most suitable candidates for trading. After the ranking is performed, we introduce the *dynamic asset selection* step: from this pool of assets, we discard those that do not satisfy a prediction accuracy higher than a given threshold ε in a past trading period, rearranging the ranking accordingly. The next step consists of selecting the top k (winners) and flop k (losers) assets and issue the corresponding trading signals: k long signals for the top k stocks and k short signals for the bottom k stocks. These selections are repeated for every day d in the trading period. Finally, we evaluate the performance of our architecture by means of back-testing strategy [4]. As mentioned in the introduction we have instantiated one example out of our general framework by using out pool of assets, S as being stocks within the S&P500 Index [29, 17], the trading session to be intra-day, and number of pairs to be traded $k = 5$. The set of features $\mathcal{F}$ and the regressors will be described, respectively, in the next two sections.

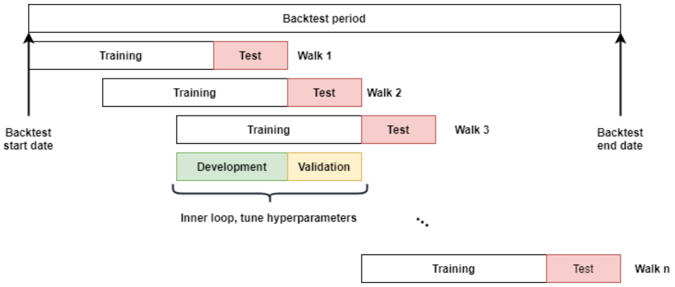

Fig. 2: Illustration of walk-forward procedure

5 Feature Engineering

As already mentioned, our dataset of reference for the instance we propose is the S&P500 Index. Therefore we have collected the information for all the stocks that have been listed, at least once, as constituents of it in a period from January 2003 to January 2016.

For each stock, we have available daily raw financial information such as *Open Price*, *High Price* in the day, *Close Price*, *Low Price* in the day, and *Volume* of stocks traded during the day. Based on this information, we have created two different kinds of features:

[1] We are using information available prior to the target date d

Table 1: Selected Technical Indicators and their acronyms throughout this paper.

Name of technical indicator	Formula
Exponential Moving Average ($EMA(10)$)	$(C_t \times a) + (EMA_{t-1} \times (1-a))$ where $a = 2/(n+1)$
Stochastic %K (%K)	$\%K = \frac{(C_t - LLt-n)}{(HH_{t-n} - LL_{t-n})} \times 100$
Price rate of change (ROC)	$\frac{C_t - C_{t-n}}{C_{t-n}} \times 100$
Relative Strength Index (RSI)	$100 - \frac{100}{1+(U/T_n)}$
Accumulation Distribution Oscillator (AccDO)	$\frac{(C_t - LL_{t-n}) - (HH_{t-n} - C_t)}{HH_{t-n} - LLt-n} \times V$
Moving Average Convergence - Divergence (MACD)	$EMA_{12}(t) - EMA_{26}(t)$
Williams %R	$\frac{HH_{t-n} - C_t}{HH_{t-n} - LL_{t-n}} \times 100$
Disparity 5 (Disp (5))	$\frac{C_t}{MA_5} \times 100$
Disparity 10 (Disp (10))	$\frac{C_t}{MA_{10}} \times 100$

C_t is the closing price at time t, L_t the low price at time t, H_t high price at time t, LL_{t-n} lowest low in the last $t-n$ days, HH_{t-n} highest high in the last $t-n$ days, MA_t the simple moving average of t days, U represents the total gain in the last n days and T_n represents the total loss in last n days

i. **Lagged daily price returns (LR)**: historical price returns are the set of features most used in financial studies. For a given trading day d, in the lag $[d - \varDelta d, d-1]$, we compute the $LR_{d,\varDelta d}$ as follows:

$$LR_{d,\varDelta d} = \frac{closePrice_{d-\varDelta d} - openPrice_{d-\varDelta d}}{openPrice_{d-\varDelta d}}, \tag{1}$$

We have set $\varDelta d \in \{1, \ldots, 10\}$, thus having for each day d 10 different lagged price returns shown as it follows:

$$[LR^{s_i}_{d-10}, LR^{s_i}_{d-9}, LR^{s_i}_{d-8}, LR^{s_i}_{d-7}, LR^{s_i}_{d-6}, LR^{s_i}_{d-5}, LR^{s_i}_{d-4}, LR^{s_i}_{d-3}, LR^{s_i}_{d-2}, LR^{s_i}_{d-1}]$$

The target value associated to this feature vector is the intra-day price return for d.

ii. **Technical Indicators (TI)**: following [24], we use a set of technical indicators summarized in Table 1. We opted for this set of features as we are interested in predicting the price movement range and also its direction. Each of the technical indicators has different insights of the stock price movement.

For this second type of feature we built the following vector:

$$[EMA(10), \%K, ROC, RSI, AccDO, MACD, \%R, Disp(5), Disp(10)]$$

Similarly as for the **LR** feature vector, the associated target value (label) is the intra-day price return for the current day.

6 Baselines

In the proposed instance of our general framework we considered the following three different state-of-the-art machine learning models and the widely known statistical model, ARIMA.

Light Gradient Boosting (LGB) is a relatively new Gradient Boosting Decision Tree algorithm, proposed in [26], which has been successfully employed in multiple tasks not only for classification and regression but also for ranking. LGB applies iteratively weak learners (decision trees) to re-weighted versions of the training data [19]. After each boosting iteration, the results of the prediction are evaluated according to a decision function and data samples are re-weighted in order to focus on examples with higher loss in previous steps. This method grows the trees by applying the leaf-wise (or best-first) strategy. The tree is grown until the maximum depth is reached, thus making this algorithm more prone to over-fitting. To control this behavior we defined the maximum depth levels of the tree, *max_depth*, to 8. We chose to vary the *num_leaves* parameter in the set $[70, 80, 100]$, achieving a balance between a conservative model and a good generalization. The feature selection is restricted by a parameter *colsample_by_tree* set at 0.8 of the total number of features, which can be thought as a regularization parameter. The work in [19] suggests a learning rate lower than 0.1, so we set it to 0.01 to account for a better generalization over the data set.

Random forests (RF) belong to a category of ensemble learning algorithms introduced in [6]. This learning method is the extension of traditional decision trees techniques where random forests are composed of many deep decorrelated decision trees. Such a decorrelation is achieved by bagging and by random feature selection. These two techniques make this algorithm robust to noise and outliers. In the case of RF, the larger the size of the forest (the number of trees), the better the convergence of the generalization error. But a higher number of trees or a higher depth of each tree induces computations costs, therefore a trade-off must be made between the number of trees in the forest and the improvement in learning after each tree is added to the forest. We opt to vary the number of trees by ranging *n_estimators* from 50 to 500 with a 25 increment. We based our choice on the work of [23]. Random feature selection operations substantially reduce trees bias, thus we set *min_samples_leaf* to 3 of the total number of features in a leaf. The learning rate is set to 0.01.

Support Vector Regressors (SVR) were proposed initially as supervised learning model in classification, and later revised for regression in [38]. Given the set of training data the goal is to find a function that deviates from actual data by a value no greater than ε for each training point, and at the same time is as flat as possible. It extends least-square regression by considering an ε-insensitive loss function. Further, to avoid overfitting of the training data, the concept of regularization is usually applied. An SVR thus solves an optimization problem that involves two parameters: the regularization parameter (referred to as C) and the error sensitivity parameter (referred to as ε). C, the regularization cost, controls the trade off between model complexity and the number of non-separable samples. A lower C will encourage a larger margin, whereas higher C values lead to hard margin [38]. Thus, we set our search space in $\{8, 10, 12\}$. Parameter ε controls the width of the ε-insensitive zone, and is used to fit the training data. A too high value leads to flat estimates, whereas a too small value is not appropriate for large or noisy data-sets. Therefore, we set it to 0.1. In this study, we selected the radial basis function (RBF) as kernel. The work in [11] suggests that the γ value of the kernel function should vary together with C, and higher values of C require higher values for gamma too. Therefore, we set a smaller search space in $\{0.01, 0.5\}$.

ARIMA model was first introduced by [5], and has been ever-since one of the most popular statistical methods used for time-series forecasting. The algorithm captures a suite of different time-dependent structures in time series. As its acronym indicates $ARIMA(p, d, q)$ comprises three parts: *autoregression model* that uses the dependencies between an observation and a number of lagged observations (p); *integration differencing* of observations with different degree, to make the time series stationary; and *Moving Average model* that accounts the dependency between observations and the residual error terms when a moving average model is used to the lagged observations (q). We chose the lag order $p \in \{1, 5\}$, the degree of differencing $d \in \{1, 5\}$, the size of the moving average window $q \in \{0, 5\}$.

7 Ensemble

In the last section we have described the regressors that are included in the ensemble of the instance we proposed in this paper. There has been a parameters-tuning step for each of the regressors in order to find the best features and parameters combinations that gave the highest performance for each model. This has been done for each walk and for each company. Besides the features mentioned within Section 5, our models might also be stock-based or industry-based. In particular, we considered:

- a model for each stock $s_i \in S$ in the training period,
- a model for each industry by grouping stocks by their industry sector as given by the Global Industry Classification Standard (GICS).

This was encouraged by previous work [18], where some portfolios were restricted to only include stocks from the same industry. Moreover, usually companies in the same industry tend to have similar behavior and exhibit some sort of correlation in their stock prices movement. Therefore, for the training of each regressor, we used a cross-validation procedure, as illustrated in Figure 2, composed of three steps where, for each walk and company:

- we split the training portion of the dataset into development and validation sets;
- each model has been trained on the development subset, and the parameters and features (LR, TI, stock-based, industry based) combination that minimized the forecasting error on the validation set were chosen;
- finally the best model found at the previous step is trained on the full training set and tested on the test set.

During each of the walks, for each company, all the combinations of parameters, industry and stock-based training, and features described in Section 5 are selected for each of the four regressor types in order to find the model that gives the best validation prediction rate, *i.e.*, the lowest Mean Squared Error (MSE). During each walk and for each company, the four predictions are hence averaged to obtain the ensemble score.

8 Dynamic asset selection

We propose a stock pruning mechanism by performing a *dynamic asset selection* strategy. For a stock $s_i \in S$, given its past forecastings $o_t^{s_i,ENS}$, and also its past real values $y_d^{s_i}$ in a predefined look-back period T, we compute a modified version of the mean directional accuracy as follows:

$$MDA_{s_i,T,d} = \frac{1}{T} \sum_{t=d-1}^{d-T-1} \mathbf{1}_{sgn(o_t^{s_i,ENS})==sgn(y_t^s)}, \tag{2}$$

where d is the current trading day, T is the look-back length and $\mathbf{1}_P$ is the indicator function that converts any logical proposition P into a number that is 1 if the proposition is satisfied, and 0 otherwise, $sgn(\cdot)$ is the sign function. The $MDA_{s,T,d}$ metric compares the forecasted direction (upward or downward) with the realized direction, providing the probability that the forecasting model can detect the correct direction of returns for a stock s_i on a given timespan T prior to day d. Such a component introduces a new step in the StatArb pipeline: after the forecast is done, we rank the companies by their forecasted daily price returns. From this pool of companies, we discard those that do not satisfy a prediction accuracy higher than a given threshold ε in a past trading period, rearranging the ranking accordingly. The proposed dynamic asset selection strategy requires a series of parameters: the accuracy threshold ε, and rolling window length related to the past trading period, T. The threshold value has been set to $\varepsilon = 0.5$ as advised in [21] for a similar scenario.

9 Experimental Framework

In this section, we describe the experimental procedure we have carried out for the instance we have tested of the proposed general framework for risk controlled trading. We conducted the experiments on the S&P500 Index dataset focusing on data from January 2003 to January 2016. We considered four years for training (that is why our tests begin from March 2007)[2] and approximately one year for trading (or testing). We compared our approach (ensemble with the dynamic asset selection , ENS-DS), against the ensemble without the dynamic asset selection (ENS) and against each single regressor and the well known Buy&Hold passive investment strategy. The metrics we have used for comparison are: (i) return (cumulative, annual and mean daily); (ii) Sharpe ratios; and (iii) Maximum drawdown. Return defines the amount that the returns on assets have gained or lost over the indicated period of time. The Sharpe ratio (SR) measures the reward-to-risk ratio of a portfolio strategy, and is defined as excess return per unit of risk measured in standard deviations. The Maximum drawdown (MaxDD) is the maximum amount of wealth reduction that a cumulative return has produced from its maximum value over time. The results are summarized in Table 2. According to the cumulative return development over time in

Table 2: Results of the StatArb strategy over a period between March 2007 to January 2016

Metod	Cumulative Return (%)	Annual Return (%)	Daily Return (%)	MaxDD(%)	SR
LGB	157.52	20.13	0.071	38.52	1.08
RF	78.476	7.89	0.035	24.15	0.5
SVM	160.56	20.13	0.072	32.35	0.1
ARIMA	108.62	12.82	0.049	42.05	0.64
S&P500 Buy-and-Hold	37.36	3.02	0.013	45.42	0.15
ENS	250.95	31.30	0.113	14.42	1.76
ENS-DS, $T = 40$ days	**263.99**	**36.6**	**0.119**	**11.5**	**2.01**

[2] There are 21 trading days in one month.

Table 2, the ENS strategy outperforms all the other non-ensemble models. Its daily returns is almost ten times the level of the Buy&Hold and up to three times the return of some individual regressors, (*e.g.*, RF). Moreover, compared to the simple average ensemble, the ENS-DS approach (with $T = 40$) has a performance increase of 5 percentage points.

Besides the return, in terms of risk exposure, the MaxDD offers an outlook on how sustainable an investment loss can be (lower is better). Also for this metric we notice the better performance of ENS-DS compared to the Buy&Hold strategy and each other baseline. The ENS-DS strategy produces a MaxDD of 11.5% that is less than one fourth of the Buy-and-Hold strategy(45%). Finally, it can be noticed that SR started from 1.76 for the simple ensemble and turned into 2.01 for the proposed ENS-DS, beating all the other baselines.

10 Conclusions and Future Work

In order to provide insights about efficient stock trading, in this paper we proposed a general framework for risk controlled trading based on machine learning and statistical arbitrage. The forecast is performed by an ensemble of regression algorithms and a dynamic asset selection strategy that prunes assets if they had a decreasing performance in the past period. As the proposed framework is general as all of its components, we fixed them and thus created an instance out of it. In our instance we focused on the S&P500 Index, using the statistical arbitrage as a trading strategy. Moreover, we propose to forecast intraday returns using an ensemble of Light Gradient Boosting, Random Forests, Support Vector Machines and ARIMA. We also proposed a set of heterogeneous features that can be used to train the models. By performing a walk-forward procedure, for each company and walk we tested all the combinations of features and internal parameters of each regressor to select the best model for each of them. The ensemble decision has been performed for each walk and company by averaging the forecast of each regressor. Our experiments showed that our ensemble strategy with the dynamic asset selection reaches significant returns of 0.119% per day, or 36.6% per year.

References

[1] Ariyo, A.A., Adewumi, A.O., Ayo, C.K.: Stock price prediction using the arima model. In: 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation. pp. 106–112 (March 2014). https://doi.org/10.1109/UKSim.2014.67

[2] Atsalakis, G.S., Valavanis, K.P.: Surveying stock market forecasting techniques – Part II: Soft computing methods. ESWA **36**(3), 5932–5941 (apr 2009). https://doi.org/10.1016/J.ESWA.2008.07.006

[3] Atzeni, M., Recupero, D.R.: Multi-domain sentiment analysis with mimicked and polarized word embeddings for human–robot interaction. FGCS (2019). https://doi.org/https://doi.org/10.1016/j.future.2019.10.012, http://www.sciencedirect.com/science/article/pii/S0167739X19309719

[4] Avellaneda, M., Lee, J.H.: Statistical arbitrage in the us equities market. Quantitative Finance **10**(7), 761–782 (2010). https://doi.org/10.1080/14697680903124632

[5] Box, G.E.P., Jenkins, G.: Time Series Analysis, Forecasting and Control. Holden-Day, Inc., San Francisco, CA, USA (1990)

[6] Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (Oct 2001). https://doi.org/10.1023/A:1010933404324

[7] Brown, G., Wyatt, J.L., Tiňo, P.: Managing diversity in regression ensembles. J. Mach. Learn. Res. **6**, 1621–1650 (Dec 2005)

[8] Carta, S., Corriga, A., Ferreira, A., Recupero, D.R., Saia, R.: A holistic auto-configurable ensemble machine learning strategy for financial trading. Computation **7**(4), 67 (2019)

[9] Carta, S., Ferreira, A., Recupero, D.R., Saia, M., Saia, R.: A combined entropy-based approach for a proactive credit scoring. Eng. Appl. Artif. Intell. **87** (2020). https://doi.org/10.1016/j.engappai.2019.103292, https://doi.org/10.1016/j.engappai.2019.103292

[10] Cavalcante, R.C., Brasileiro, R.C., Souza, V.L., Nobrega, J.P., Oliveira, A.L.: Computational Intelligence and Financial Markets: A Survey and Future Directions. Expert Systems with Applications **55**, 194–211 (aug 2016). https://doi.org/10.1016/J.ESWA.2016.02.006

[11] Chalimourda, A., Schölkopf, B., Smola, A.J.: Experimentally optimal ν in support vector regression for different noise models and parameter settings. Neural Networks **17**(1), 127 – 141 (2004). https://doi.org/https://doi.org/10.1016/S0893-6080(03)00209-0

[12] Damghani, B.M.: The non-misleading value of inferred correlation: An introduction to the cointelation model. Wilmott **2013**(67), 50–61 (2013). https://doi.org/10.1002/wilm.10252

[13] Dawid, A.P.: Present position and potential developments: Some personal views statistical theory the prequential approach. Journal of the Royal Statistical Society: Series A (General) **147**(2), 278–290 (1984)

[14] Devezas, T.: Principles of forecasting. a handbook for researchers and practitioners: J. scott armstrong. kluwer academic publishers, norwell, ma, usa, 2001, xii and 849 pages. isbn 0-7923-7930-6 (hardbound); us$190. Technological Forecasting and Social Change **69**(3), 313 – 316 (2002). https://doi.org/https://doi.org/10.1016/S0040-1625(02)00180-4

[15] Dixon, M., Klabjan, D., Bang, J.H.: Classification-based financial markets prediction using deep neural networks. Algorithmic Finance **6**(3-4), 67–77 (2017). https://doi.org/10.3233/AF-170176

[16] Enke, D., Thawornwong, S.: The use of data mining and neural networks for forecasting stock market returns. Expert Systems with Applications **29**(4), 927–940 (nov 2005). https://doi.org/10.1016/J.ESWA.2005.06.024, https://www.sciencedirect.com/science/article/pii/S0957417405001156?via%3Dihub

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

[18] Gatev, E., Goetzmann, W.N., Rouwenhorst, K.G.: Pairs Trading: Performance of a Relative-Value Arbitrage Rule. The Review of Financial Studies **19**(3), 797–827 (02 2006). https://doi.org/10.1093/rfs/hhj020

[19] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition Springer Series in Statistics (02 2009)

[20] Henrique, B.M., Sobreiro, V.A., Kimura, H.: Literature review: Machine learning techniques applied to financial market prediction. Expert Systems with Applications **124**, 226–251 (jun 2019). https://doi.org/10.1016/J.ESWA.2019.01.012

[21] Huck, N.: Pairs selection and outranking: An application to the S&P 100 index. European Journal of Operational Research **196**(2), 819–825 (2009). https://doi.org/10.1016/j.ejor.2008.03.025

[22] Huck, N.: Pairs trading and outranking: The multi-step-ahead forecasting case. European Journal of Operational Research **207**(3), 1702 – 1716 (2010). https://doi.org/https://doi.org/10.1016/j.ejor.2010.06.043

[23] Huck, N.: Large data sets and machine learning: Applications to statistical arbitrage. European Journal of Operational Research **278**(1), 330–342 (oct 2019). https://doi.org/10.1016/J.EJOR.2019.04.013

[24] Kara, Y., Acar Boyacioglu, M., Baykan, Ö.K.: Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. Expert Systems with Applications **38**(5), 5311–5319 (may 2011). https://doi.org/10.1016/J.ESWA.2010.10.027

[25] Kaufman, C., Lang, D.T.: Pairs trading. Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving (April 2015), 241–308 (2015). https://doi.org/10.1201/b18325

[26] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30. pp. 3146–3154. Curran Associates, Inc. (2017)

[27] Khandani, A.E., Lo, A.W.: What happened to the quants in august 2007? evidence from factors and transactions data. Journal of Financial Markets **14**(1), 1 – 46 (2011). https://doi.org/https://doi.org/10.1016/j.finmar.2010.07.005

[28] Knoll, J., Stübinger, J., Grottke, M.: Exploiting social media with higher-order factorization machines: statistical arbitrage on high-frequency data of the s&p 500. Quantitative Finance **19**(4), 571–585 (2019), www.scopus.com

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

[30] Large, J., Lines, J., Bagnall, A.: The heterogeneous ensembles of standard classification algorithms (hesca): the whole is greater than the sum of its parts (2017)

[31] Lee, K.J., Yoo, S., Jin, J.J.: Neural network model vs. sarima model in forecasting korean stock price index (kospi) (2007)

[32] Leung, M.T., Daouk, H., Chen, A.S.: Forecasting stock indices: a comparison of classification and level estimation models. International Journal of Forecasting **16**(2), 173 – 190 (2000). https://doi.org/https://doi.org/10.1016/S0169-2070(99)00048-5, http://www.sciencedirect.com/science/article/pii/S0169207099000485

[33] Lo, A.W.: Hedge Funds: An Analytic Perspective (Revised and Expanded Edition). Princeton University Press, stu - student edition edn. (2010)

[34] Lo, A., Hasanhodzic, J.: The Evolution of Technical Analysis: Financial Prediction from Babylonian Tablets to Bloomberg Terminals. Bloomberg, Wiley (2011)

[35] Merh, N., Saxena, V.P., Pardasani, K.R.: A comparison between hybrid approaches of ann and arima for indian stock trend forecasting (2010)

[36] Sutherland, I., Jung, Y., Lee, G.: Statistical arbitrage on the kospi 200: An exploratory analysis of classification and prediction machine learning algorithms for day trading. Journal of Economics and International Business Management **6**(1), 10–19 (2018)

[37] Takeuchi, L.: Applying Deep Learning to Enhance Momentum Trading Strategies in Stocks (2013)

[38] Vapnik, V.N.: An overview of statistical learning theory. IEEE Transactions on Neural Networks **10**(5), 988–999 (Sep 1999). https://doi.org/10.1109/72.788640

[39] Vidyamurthy, G.: Pairs trading : Quantitative methods and analysis / g. vidyamurthy. John Wiley & Sons (01 2004)