

Model Calibration and Automated Trading Agent for Euro Futures

Germán Creamer
Stevens Institute of Technology
Castle Point on Hudson
Hoboken, NJ 07302
email: gcream@stevens.edu

ABSTRACT

We explored the application of a machine learning method, Logitboost, to automatically calibrate a trading model using different versions of the same technical analysis indicators. This approach takes advantage of boosting's feature selection capability to select an optimal combination of technical indicators and design a new set of trading rules. We tested this approach with high frequency data of the Dow Jones EURO STOXX 50 Index Futures (FESX) and the DAX Futures (FDAX) for March 2009. Our method was implemented with different learning algorithms and outperformed a combination of the same group of technical analysis indicators using the parameters typically recommended by practitioners.

We incorporated this method of model calibration in a trading agent that relies on a layered structure consisting of the machine learning algorithm described above, an online learning utility, a trading strategy, and a risk management overlay. The online learning layer combines the output of several experts and suggests a short or long position. If the expected position is positive (negative), the trading agent sends a buy (sell) limit order at prices slightly lower (higher) than the bid price at the top of the buy (sell) order book less (plus) transaction costs. If the order is not 100% filled within a fixed period (i.e. 1 minute) of being issued, the existent limit orders are cancelled, and limit orders are reissued according to the new experts' forecast. As part of its risk management capability, the trading agent eliminates any weak trading signal.

The trading agent algorithm generated positive returns for the two major European index futures (FESX and FDAX) and outperformed a buy and hold strategy.

KEY WORDS

Automated trading, machine learning, algorithmic trading, agent based economics, trading agents, boosting.

1. Introduction

Most of the research applying agent based modeling and machine learning methods to finance have been in the equity [1, 46, 47, 57] and foreign exchange markets [21,

6, 20, 40, 2] using daily prices. There is very limited research in the area of high frequency futures trading, but equity, foreign exchange and index futures trading have a similar forecasting problem: anticipating price trend. Additionally, there is a potential increase of time series variance when the analysis is done using high frequency data instead of using daily data. Hence, there are new challenges in the application of learning algorithms to forecast futures using high frequency data.

Among the most important learning algorithms used to forecast futures are the following:

- Neural network (connectionist approach): This has been the most commonly studied approach. Most systems generate trading rules using neural networks where their main inputs are technical analysis indicators.¹

This approach has been applied to forecast and trade S&P 500 index futures [58, 15], the Warsaw stock price index 20 (WIG20) futures [60], and Korea stock index 200 (KOSPI 200) futures [38]. Hamid and Iqbal [33] forecast the volatility of S&P 500 index futures using 16 futures indexes and 3 spot prices. Tsaih et al. [59] integrate a rule based system and neural network to forecast the daily trend of S&P 500 index futures. Duke and Long [22] use price and economic indicators as input to forecast daily German government bond futures prices. Kim [37] uses a method of feature transformation based on domain knowledge to discretize the data and forecast the KOSPI 200. Even though these studies indicate that they outperform their benchmarks, the main problem with the neural network approach is that it is very difficult to interpret the trading rules generated, especially in the case of complex networks with many nodes and hidden layers.

- Genetic algorithm (emergent approach): the genetic algorithm [35] or genetic programming [39] approach is used to generate evolving trading rules. These rules are represented as binary trees where the leaves are technical indicators and the non-leaves are boolean functions. Together they represent simple decision functions. Following this perspective, Kyung-shik Shin et al. [54] predict the KOSPI using nine technical indicators. The advantage of this approach is that the rules are interpretable.
- Support vector machine: Tay and Cao [56] show a support vector machine application that gives more weight to more recent values, and improves forecasts of the S&P 500 index, and US and German government bond futures using moving averages and lagged prices.
- Reinforcement learning: Moriyama et al. [48] successfully test the application of reinforcement learning to trade on a futures market simulator (U-Mart) of the large Japanese industrial companies (J30) index.

In this research we followed the tradition of the papers in this section that use machine learning algorithms to find profitable trading strategies, and build automated

¹Technical analysis or technical trading strategies try to exploit statistically measurable short-term market opportunities, such as trend spotting and momentum, in individual industrial sectors (e.g. financial, pharmaceutical etc.).

trading agents. We also examined a method to calibrate the trading rules or technical indicators using a learning algorithm.

The rest of the paper is organized as follows: section 2 announces the methods used; section 3 introduces the model calibration process; section 4 presents the trading agent and its algorithm; section 5 describes the data; section 6 explains in detail the experiments; section 7 presents the results of our trading agent algorithm; section 8 concludes, and the appendix introduces the main investment indicators used in this research.

2. Methods

2.1. Boosting

Adaboost is a general discriminative learning algorithm invented by Freund and Schapire [28].

The basic idea of Adaboost is to repeatedly apply a simple learning algorithm, called the *weak* or *base learner*², to different weightings of the same training set. In its simplest form, Adaboost is intended for binary prediction problems where the training set consists of pairs $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$, x_i corresponds to the features of an example, and $y_i \in \{-1, +1\}$ is the binary label to be predicted. A *weighting* of the training examples is an assignment of a non-negative real value w_i to each example (x_i, y_i) .

On iteration t of the boosting process, the weak learner is applied to the training set with a set of weights w_1^t, \dots, w_m^t and produces a prediction rule h_t that maps x to $\{0, 1\}$.³ The requirement on the weak learner is for $h_t(x)$ to have a small but significant correlation with the example labels y when measured using the *current weighting of the examples*. After the rule h_t is generated, the example weights are changed so that the weak predictions $h_t(x)$ and the labels y are decorrelated. The weak learner is then called with the new weights over the training examples, and the process repeats. Finally, all of the weak prediction rules are combined into a single *strong* rule using a weighted majority vote. One can prove that if the rules generated in the iterations are all slightly correlated with the label, then the strong rule will have a very high correlation with the label – in other words, it will predict the label very accurately.

The whole process can be seen as a variational method in which an approximation $F(x)$ is repeatedly changed by adding to it small corrections given by the weak prediction functions. In Figure 1, we describe Adaboost in these terms. We shall refer to $F(x)$ as the *prediction score* in the rest of the document. The strong prediction rule learned by Adaboost is $\text{sign}(F(x))$.

A surprising phenomenon associated with Adaboost is that the test error of the strong rule (percentage of mistakes made on new examples) often continues to decrease even after the training error (fraction of mistakes made on the training set) reaches zero.

²Intuitively, a weak learner is an algorithm with a performance at least slightly better than random guessing

³Mapping x to $\{0, 1\}$ instead of $\{-1, +1\}$ increases the flexibility of the weak learner. Zero can be interpreted as “no prediction”.

This behavior has been related to the concept of a “margin”, which is simply the value $yF(x)$ [52]. While $yF(x) > 0$ corresponds to a correct prediction, $yF(x) > a > 0$ corresponds to a *confident* correct prediction, and the confidence increases monotonically with a . Friedman et al. [29], followed by Collins et al. [17] suggested a mod-

$$\begin{aligned}
&F_0(x) \equiv 0 \\
&\text{for } t = 1 \dots T \\
&\quad w_i^t = e^{-y_i F_{t-1}(x_i)} \\
&\quad \text{Get } h_t \text{ from } \textit{weak learner} \\
&\quad \alpha_t = \frac{1}{2} \ln \left(\frac{\sum_{i: h_t(x_i)=1, y_i=1} w_i^t}{\sum_{i: h_t(x_i)=1, y_i=-1} w_i^t} \right) \\
&\quad F_{t+1} = F_t + \alpha_t h_t
\end{aligned}$$

Figure 1: The Adaboost algorithm [28]. y_i is the binary label to be predicted, x_i corresponds to the features of an instance i , w_i^t is the weight of instance i at time t , h_t and $F_t(x)$ are the prediction rule and the prediction score at time t respectively

ification of Adaboost, called Logitboost. Logitboost can be interpreted as an algorithm for step-wise logistic regression. This modified version of Adaboost—known as Logitboost—assumes that the labels y_i 's are stochastically generated as a function of the x_i 's. Then it includes $F_{t-1}(x_i)$ in the logistic function to calculate the probability of y_i , and the exponent of the logistic function becomes the weight of the training examples. Figure 2 describes Logitboost using notation similar to the one used in 1. Some successful and popular way of using boosting is to combine it with a decision tree learning algorithms as the base learning algorithm [29]. We use boosting both to learn the decision rules constituting the tree and to combine these rules through a weighted majority vote. The form of the generated decision rules is called an *alternating decision tree* (ADT) [27]. In ADTs each node can be understood in isolation.

In terms of boosting, each prediction node represents a weak prediction rule, and

$$\begin{aligned}
&F_0(x) \equiv 0 \\
&\text{for } t = 1 \dots T \\
&\quad w_i^t = \frac{1}{1 + e^{y_i F_{t-1}(x_i)}} \\
&\quad \text{Get } h_t \text{ from } \textit{weak learner} \\
&\quad \alpha_t = \frac{1}{2} \ln \left(\frac{\sum_{i: h_t(x_i)=1, y_i=1} w_i^t}{\sum_{i: h_t(x_i)=1, y_i=-1} w_i^t} \right) \\
&\quad F_{t+1} = F_t + \alpha_t h_t
\end{aligned}$$

Figure 2: The Logitboost algorithm [29]. y_i is the binary label to be predicted, x_i corresponds to the features of an instance i , w_i^t is the weight of instance i at time t , h_t and $F_t(x)$ are the prediction rule and the prediction score at time t respectively

at every boosting iteration, a new splitter node together with its two prediction nodes is added to the model. The splitter node can be attached to any previous prediction node, not only leaf nodes. Each prediction node is associated with a weight α that contributes to the prediction score of every example reaching it. The weak hypothesis $h(x)$ is 1 for every example reaching the prediction node and 0 for all others.

We decided to use boosting, specifically Logitboost, as our learning algorithm because of its feature selection capability, its error bound proofs [28], its interpretability, and its capacity to combine quantitative, and qualitative variables. Additionally, we used ADT implemented with Logitboost because of its capacity to generate classification rules that are smaller and easier to interpret than the rules generated by other boosting decision trees learning algorithms such as CART or C4.5. Additionally, ADT gives a measure of confidence or classification margin which is very useful for experts selection.

2.2. Bagging

Bagging was proposed by Breiman [10] as a method that reduces the variance of a prediction function. If the training set Υ consists of pairs $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$, x_i corresponds to the features of an example, and y_i is either a class label or a numerical response to be predicted. The predictor of y is $\psi(x, \Upsilon)$. Bagging or bootstrap aggregation generates uniform bootstrap samples with replacement of Υ . These samples and their predictors are $\Upsilon^{(B)}$ and $\psi(x, \Upsilon^{(B)})$ respectively.

When y_i is a numerical response, the final predictor is obtained by the average of the predictors of the bootstrap samples as

$$\psi_B(x) = av_B \psi(x, \Upsilon^{(B)}).$$

If y_i is a class label, $\psi_B(x)$ is obtained by the majority vote of $\psi(x, \Upsilon^{(B)})$.

Bagging has been shown to be particularly effective for reducing the variance of decision trees.

2.3. Logistic regression

The logistic regression models [34] the posterior probabilities $Pr(Y = l|X_i)$ of L classes Y using linear regression in the observed values X_{ij} of the input variable $X_i = (X_{i1} \dots X_{in})$ of the feature j :

$$Pr(Y = l|X_i) = \frac{e^{\sum_{j=1}^n \beta_{lj} X_{ij}}}{1 + e^{\sum_{j=1}^n \beta_{lj} X_{ij}}}$$

The summation of these probabilities equals one. Logistic regression results are better interpreted using the odds ratio:

$$\frac{Pr(Y=l|X_i)}{Pr(Y=L|X_i)}$$

In the current paper, L is two because we are trying to evaluate if the price trend is positive or negative.

3. Model calibration using boosting

In contrast to econometric models where coefficients are defined by formal statistical models, trading rules and technical analysis require many parameters that in many cases are established by practitioners' experience. Hence, model calibration is one of

the major practical problems that algorithmic trading may have, especially if it is based on technical analysis.

Genetic algorithm and simulated annealing are methods used by practitioners to calibrate technical trading models [36]. nez [49] applies genetic algorithm to optimize the parameters used to calculate moving averages, although a trading strategy derived from this indicator underperforms a simple buy and hold strategy for the two main indexes of the Madrid stock market (IBEX-35 and the General Index, IGE). Bodas-Sagi et al. [7] use a genetic algorithm multi-objective optimization model to calibrate the moving average convergence divergence and the relative strength index indicators for the Dow Jones index.

Another approach is the “brute force” method where a large number of alternatives are tested and the best option is selected. The problem with this perspective is that it is computationally intensive, and high frequency data may change significantly in different periods, possibly requiring continuous calibration.

The method that we evaluate in this paper, initially explored by Creamer and Freund [19], recalculates every technical indicator n number of times using different values of its parameters. All these different versions of the technical indicators become the input of the boosting algorithm. Boosting, based on its feature selection capability, will select a group of parameters and technical indicators that optimizes its evaluation function. The initial parameters used in this research are those recommended in the literature of technical analysis.

4. Trading agent algorithm

Agent based economics is becoming a well recognized methodology for market simulation. Main economics conferences such as the Eastern Economics Association meeting include several sessions on agent based economics [4, 5]. From a trading perspective, the agent-based approach could be useful for backtesting new ideas without risking any money. The Santa Fe stock market model⁴ has inspired many other agent-based financial market models such as Ehrentreich [24] that is based on the Grossman and Stiglitz model [32]. In the Santa Fe stock market, agents can classify and explore several forecasting rules that are built using genetic algorithms. Many of the models built according to this perspective test the performance of agents or algorithms that have unique characteristics. For example, Lettau [44] builds an agent-based financial market using simple agent benchmarks based on genetic algorithms; Gode and Sunder [30] develop a double action market using zero intelligence traders; Arifovic [2] builds a model of the foreign exchange market using genetic algorithms; Routledge [51] extends the basic framework of Grossman and Stiglitz [32] with agents that can learn by using genetic algorithms; Chan et al. [12] and Chan [13] use the artificial market framework to explore the behavior of different trading approaches and their microstructure impact.

⁴For a presentation of the Santa Fe stock market model see [3, 43], and a later version at [42]. LeBaron [41] also has a general review of papers in the area of agent-based finance.

In contrast to the previous references, this paper proposes a high frequency trading agent for equity index futures. This agent uses the expert weighting algorithm introduced by Creamer and Freund [19] to forecast a price trend as an input for a trading strategy based on a variation of a market maker strategy proposed by Creamer and Freund [18]. This approach forecasts the price trend of the future using a machine learning algorithm, and combines several investment signals as well as the output of several experts using an online learning utility. Finally, it controls the level of risk using a risk management overlay.

One of the strengths of our approach is that the algorithm optimizes the parameters of the technical analysis indicators as explained in section 3; defines new trading rules, and generates experts at different moments of the trading cycle. The main objective of this process is to assure that the investment decision takes into account old and new patterns of the time series under study, and that there is an online learning process with the capacity to integrate the feedback of previous outcomes. The online learning algorithm comes from Freund et al. [26], and from the weighted majority algorithm proposed by Littlestone and Warmuth [45] and further studied by Cesa-Bianchi et al. [11].

The objective of the algorithm is to predict the return of the future contract in the next period. $y_t \in [-1, +1]$ is the binary label to be predicted where 1 represents the expectation of a positive return, and -1 otherwise.

The market maker trading strategy initially proposed by Creamer and Freund [18] starts with a long or short portfolio position based on a daily forecast of the market trend. During the day, the trading agent also sends simultaneous buy and sell limit orders at prices slightly below and above the prices at the top of the buy and sell order books. This strategy takes advantage of the former electronic market Island's pricing policy: when a trade is executed, the trader that submitted the limit order shall receive a rebate of \$0.002, and the party that submitted the incoming order shall pay a transaction fee of \$0.003. During the week of January 5-9, 2004, an annualized Sharpe ratio of 0.21 was established using the market maker strategy which is superior to the Sharpe ratio of -0.28 that was obtained using a constant rebalanced portfolio (CRP) strategy during the same period of time. This market maker strategy is profitable because it captures the bid-ask spread and also tries to get the maximum amount in rebates and the minimum amount in fees. This strategy cannot be directly applied to products traded at Eurex because this exchange does not offer rebates, instead the trading costs are slightly lower for the party that submitted the incoming order.

Our limit order trading strategy sends only one buy or sell limit order based on the futures return forecast which is updated every g periods of time (e.g. 30 minutes) and the following rules:

- If the expected price trend is positive, it sends a buy limit order at a price slightly lower than the bid price at the top of the buy order book less transaction costs. If there is a short position, the size of the order is the short position. Otherwise, it is δ futures contract
- If the expected price trend is negative, it sends a sell limit order at a price slightly higher than the ask price at the top of the sell order book plus transaction costs.

If there is a long position, the size of the order is the long position. Otherwise, it is δ futures contract

- If the order is not 100% filled within a fixed period (i.e. 1 minute) of being issued, existent limit orders are cancelled, and limit orders are reissued according to current experts' forecast

The position is liquidated at the end of every trading day.

As part of its risk management capability, the trading agent eliminates any weak trading signal when it is below a certain threshold which is established during the training and optimization stage. If the maximum drawdown (D_t) [21, 47], the largest potential loss in a certain period of time, passes a certain threshold, then the system holds its current position.

The maximum drawdown is calculated as:

$$D_t \doteq \max(R_{t_x} - R_{t_y} | t_0 \leq t_x \leq t_y \leq t)$$

where R_{t_x} and R_{t_y} are the accumulated return from time t_0 until time t_x and t_y respectively.

The main reason to propose this risk management rule is that the trading agent should not invest further in an unprofitable strategy. However, it can reverse course, if the market conditions improve.

Additionally, the algorithm calculates the Sharpe ratio as a risk adjusted return indicator. The annualized Sharpe ratio (SR_t) is also used as a weight for each observation.

$$SR_t \doteq \frac{\mu(R_t)}{\sigma(R_t)}$$

The complete algorithm is presented in Figure 3.

5. Data

The data used in this paper is the Dow Jones EURO STOXX 50 Index Futures (FESX) and the DAX Futures (FDAX). The Dow Jones EURO STOXX 50 and the DAX indexes represent the 50 largest European companies and the 30 major German companies traded on the Frankfurt Stock Exchange, respectively. These two future indexes were selected in consideration that they both reflect the market expectation of the underlying indexes. These indexes are highly liquid, especially FESX. This data was obtained directly from Eurex.

Before expiration, both index futures are traded at their market values. If they are not used for hedging, the main decision variable for trade is the price trend. In this respect, forecasting the futures price trend is a simple classification problem.

For every index future (FESX and FDAX) we have trade and quote level 2 data. We recreated the order book at ten seconds intervals, integrating the quotes (bid price, bid volume, ask price, and ask volume) with the trades (trade volume and trade price) for March 2009 (22 trading days). The previous prices were aggregated using volume weighted average prices (VWAP), and the volumes were aggregated by a simple sum.

Input:

Set of price series: P_{best}^b and P_{best}^a are prices at the top of the buy and sell order book respectively

tc represents transaction costs (\$0.3 for FESX and \$0.5 for FDAX)

$zeta$ is a markup that at least covers the transaction costs tc

q_t is portfolio size at time t

r_t is return at period t

δ , n , d , f , and g are the number of futures contract to be traded, number of different values of parameters to calculate investment signals, number of periods between experts' training, number of periods between orders, and number of periods to evaluate market trend

γ_0 and γ_1 are thresholds to filter experts' weight.

C is an exogenous parameter for expert weighting.

Recalibrate and train with machine learning algorithm:

1. Select a representative asset (i.e. index future) from targeted market.

2. Calculate investment signals, and labels with basic parameters for the selected asset (in this research we used all the investment signals described in the appendix).

3. Recalculate investment signals with n variations of basic parameters, and include all of the investment signals as features in the training and test sets where the binary label is $y_t = \text{sign}(r_t)$ and r_t is return at time t .

4. Every d periods train a new expert i ψ_i using a learning algorithm. Call the sequence of experts as $\psi = \psi_1, \psi_2, \dots, \psi_E$ where E is the maximum number of the most recent experts.

Every d periods recalculate test set and weight experts as in next steps.

Expert weighting algorithm:

5. Calculate the weight of the first expert at time t as $w_t^1 \doteq \exp\left(\frac{C \cdot \text{car}_t^1}{\sqrt{t}}\right)$ where:

$$\text{car}_t^i \doteq \sum_{s=t_i+1}^t \text{sign}(S_s^i) \cdot r_s^i$$

$t_1 = 0$, t_i is the time step at which ψ_i is calculated when $i > 1$

r_s^i is the return for expert i at time s .

6. Calculate the weight of expert ψ_i at time $t > t_i$ as

$$w_t^i \doteq I_i \cdot \text{ramp}(t - t_i) \cdot \exp\left(\frac{C \cdot \text{car}_t^i}{\sqrt{t - t_i}}\right)$$

where:

$$I_i \doteq \frac{\sum_{j=1}^{i-1} w_{t_i}^j}{i-1}$$
 is the initial weight assigned to i (ψ_i)

$$\text{ramp}(t - t_i) \doteq \min\left(\frac{t - t_i}{t_{i+1} - t_i}, 1\right)$$

t_{i+1} is the time that the next expert is added.

7. Calculate the experts' weight as $W_t = L_t - S_t$ where $L_t = \frac{\sum_{i: S_t^i > 0} w_t^i}{\sum_i w_t^i}$ and $S_t = 1 - L_t^i$.

Risk management:

8. If $|W_t| < \gamma_0$, then $W_t = 0$

If $D_{t-1} < \gamma_1$, then $W_t = W_{t-1}$ where:

$D_t \doteq \max(R_{t_x} - R_{t_y} | t_0 \leq t_x \leq t_y \leq t)$ is the maximum drawdown

R_{t_x} , R_{t_y} , and W_t are the accumulated return from time t_0 until time t_x and t_y , and experts' weights respectively.

Trading:

9. Based on the experts' weight, every g periods take one of the following positions:

If $W_t > 0$, buy limit order for δ and $P^b = P_{best}^b - \zeta$. If $q_t < 0$ then $\delta = q_t$.

If $W_t < 0$, sell limit order for δ and $P^a = P_{best}^a + \zeta$. If $q_t > 0$ then $\delta = q_t$.

Hold, otherwise

If any of the above orders are not executed after f periods, cancel and resubmit limit orders.

10. Liquidate position q_t before market closes at the end of each trading day.

Output:

Expert weighted cumulative return is:

$$CR \doteq \sum_t (W_t \cdot r_t - (W_t - W_{t-1}) \cdot tc)$$

Figure 3: Trading agent algorithm. This is a modified version of the trading algorithm introduced by Creamer and Freund [19]

The trades and quotes data were linked according to their time stamp. Mid prices were calculated as the average between the bid and ask prices.

6. Experiments

6.1. Model calibration

We evaluated the performance of the calibration method proposed in section 3 with the FESX and FDAX index futures, splitting these datasets into training (70%) and test (30%) datasets. The number of observations for training and test datasets are 68,995 and 29,536 for FESX and 68,913 and 29,501 for FDAX, respectively. We tested our calibration method with two technical indicators: momentum and Bollinger bands,⁵ and with the combination of all the indicators listed in the appendix (without including the liquidity indicators). The dependent variable $y_t \in [-1, +1]$ is a binary variable which has a value of 1 when the return or trade price trend of the next period is positive, and a value of -1 otherwise. As a base case, we worked with a model calculated with the most common parameters for each indicator as used by practitioners and listed in the appendix. The classification of the base case is a majority vote of the respective rules of the indicators used.

Our method includes the recalculation of the technical indicators with different parameters. Besides the base case, we tried three other variations where each parameter is tested with three, six and nine different versions and where their values increase by six, three and two units, respectively. The main calculation of these indicators for the model calibration and for the trading agent is done using R, its financial engineering, order book, and analytics package called Rmetrics, RTAQ and RWeka, respectively.⁶ OneTick <<http://www.onetick.com>> is used for the generation of the order book.

Logitboost is used to classify the future trade price trend. To check for the possibility that the Logitboost results could be improved because of the characteristic instability of boosting, we ran bagging on top of Logitboost (bagged boosting). We also compared Logitboost's results with Adaboost and logistic regression to evaluate the difficulty of the classification task. We performed 10-, 50- and 100-fold cross-validation experiments to evaluate classification performance on held-out experiments. Logitboost, Adaboost and bagging are run with 20 iterations. We compared our results using the test error and the Sharpe ratio. For this test of model calibration, the Sharpe ratio is calculated using a very simple calculation of return based on the price difference between periods and the trend forecast. We did not include trading costs because we were not testing any trading strategy. We were only evaluating the trend forecast and its impact in risk adjusted return. Its application to a trading strategy is reviewed in the next section.

⁵See appendix for an explanation.

⁶See <<http://cran.r-project.org>> and <<http://www.rmetrics.org>> for information about R and Rmetrics respectively.

6.2. Trading algorithm

The trading agent algorithm presented in Figure 3 is applied to the FESX and FDAX futures. This algorithm, implemented with Logitboost, reweights the participation of each expert according to individual performance. Additionally, the risk management module holds or liquidates positions when they are not profitable or become too risky. Every day, the trading agent processes new information and takes investment decisions based on the experts generated the previous day. During the day, new experts are generated about every half hour (d) and substitute older experts maintaining a maximum of 25 experts (E). In our simulation, the first day was used only to generate experts and the agent started to trade only on the second day. Also about every half hour (g), the agent evaluates the trend of the market and sends limit orders according to the trading algorithm. The limit orders are revised every f periods of 10 seconds each. We evaluated f with the following values: 1, 2, 3, 4, 5, 6, 9, 12, 18, 24, 30, 36, 42, 48, 54, and 60. The threshold to eliminate very weak expert weights (γ_0) is set to 0.20, and the threshold to restrict trading (γ_1) is set to 0 as suggested by Creamer and Freund [19].

We decided to use boosting, specifically Logitboost, as our learning algorithm because of its feature selection capability, its error bound proofs [28], its interpretability, and its capacity to combine quantitative and qualitative variables. Additionally, Logitboost performed as well or better than the rest of the algorithms tested during our model calibration experiments. We used the implementation of boosting in Java included in the MLJAVA package or in its current version called JBoost <<http://jboost.sourceforge.net>>.

A group of well-known technical indicators, and investment signals introduced in the appendix are the inputs for the machine learning algorithm: a simple and exponential moving average, Bollinger bands, acceleration, momentum, rate of change, moving average convergence divergence, relative strength index, on balance volume, negative and positive volume index, price-volume trend and several liquidity indicators. We also included ratios and trading rules suggested by the practice of technical analysis.

The initial value of the parameters used to calculate the technical indicators are those recommended in the literature. Additionally, most of the technical indicators are calculated three extra times with parameters that increase by six units every time. Based on the result of our calibration model, the difference of performance with other combinations was not very significant and our choice reduced the computation period.

The results of the trading algorithm are aggregated in 21 trading days. We tested our results with trading costs per futures contract of 0.3 EUR and 0.5 EUR for FESX and FDAX, respectively. These values are consistent with Eurex's prices. We compared our results with a buy and hold (B&H) position using the Sharpe ratio as a risk adjusted return measure.

7. Results

Table 1 shows that all the algorithms tested with FESX and FDAX data perform better than the base case does. With 100 subsets, the base case has a test error of 49.65% for FESX and 47.46% for FDAX, while most of the algorithms explored have test errors between 44% and 46%. The mean difference between the base case and all

the algorithms explored is significant at the 99% confidence level when all the technical indicators are used. Similar results are obtained using 50 and 10 subsets.

These results are also consistent with the calculation of the Sharpe ratio. The Sharpe ratio of the base case with 100 subsets is respectively -5.26 and 19.29 for FESX and FDAX. All the learning algorithms show a positive Sharpe ratio when all the indicators are used (see Table 2) and the difference with the base case is significant with a 99% confidence level. In all cases, the Sharpe ratio is lower for the simulations when Bollinger bands and momentum are tested independently.

Logitboost and logistic regression running with all the technical indicators show the highest Sharpe ratio for both FESX and FDAX. Additionally, an increase in the number of versions (three or six) of the same technical indicators improve the Sharpe ratio, while nine versions of each parameter leads to a deterioration of the Sharpe ratio in most cases. Boosting, the primary learning algorithm, is able to handle a large number of parameters until certain point. As Creamer and Freund [19] have pointed out, the use of boosting for financial prediction improves when relevant ratios that are known for their importance to investment decisions are employed instead of using the original indicators.

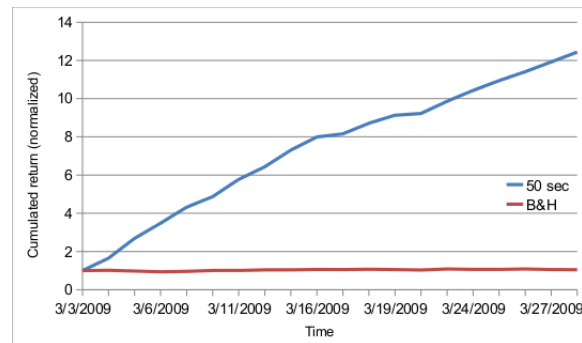


Figure 4: Cumulative return of algorithmic trading strategy and B&H for March 2009. Values are indexed to 1

These results lead us to prefer Logitboost for our trading agent algorithm. The annualized Sharpe ratio of this algorithm is reported in Table 3. It shows that the Sharpe ratio for FESX and FDAX outperforms a simple buy and hold strategy with a 99% confidence level. Additionally, orders submitted every 50 and 40 seconds leads to the maximum Sharpe ratio for either FSEX (40.17) and FDAX (41.59). Figure 4 graphically shows how the cumulated return for FESX using orders submitted every 50 seconds outperforms the buy and hold strategy.

It is more important from an algorithmic trading perspective to concentrate on a risk adjusted return indicator than on the accuracy of the algorithm. The simulations show that the evaluated algorithms may have very different Sharpe ratios even though they could have a very similar test error. This can be explained from the even weight given to all observations for the calculation of the test error, while a risk adjusted return indicator considers the return and the volatility of the investment decision. So a correct

Models	Rep.	Δ	All indicators			Bollinger bands			Momentum			
			100	50	10	100	50	10	100	50	10	
# subsets												
Base case	0	0	49.69	49.65	49.56	46.23	46.22	46.13	52.76	52.78	52.89	
Adaboost	0	0	45.69**	45.69**	45.68**	45.54**	45.83**	45.83**	45.66**	45.93**	45.93**	
	3	6	45.69**	45.68**	45.54**	45.83**	45.83**	45.66**	45.93**	45.93**	45.74**	
	6	3	45.69**	45.68**	45.54**	45.83**	45.83**	45.66**	45.93**	45.93**	45.74**	
	9	2	45.69**	45.68**	45.54**	45.91	45.93	45.81	45.93**	45.93**	45.74**	
Bagging	0	0	44.50**	44.48**	44.36**	45.81*	45.8**	45.64**	45.96**	45.96**	45.78**	
	3	6	44.31**	44.31**	44.22**	45.94	45.92*	45.76*	45.92**	45.92**	45.74**	
	6	3	44.34**	44.33**	44.23**	46.00	46.01	45.87	45.92**	45.92**	45.74**	
	9	2	44.65**	44.64**	44.54**	45.87*	45.86*	45.71*	45.93**	45.93**	45.75**	
Logistic regression	0	0	45.00**	44.96**	44.85**	45.83*	45.83**	45.65**	45.93**	45.93**	45.74**	
	3	6	45.78**	45.73**	45.74**	45.85**	45.85**	45.66**	45.92**	45.92**	45.73**	
	6	3	46.93**	46.93**	46.96**	45.94**	45.95*	45.78**	45.92**	45.92**	45.74**	
	9	2	45.95**	45.93**	45.93**	45.87**	45.87**	45.73*	45.94**	45.93**	45.75**	
Logitboost	0	0	44.53**	44.52**	44.43**	46.18	46.2	46.09	45.95**	45.97**	45.78**	
	3	6	44.52**	44.51**	44.43**	45.84*	45.83*	45.66*	45.84**	45.83**	45.65**	
	6	3	44.36**	44.35**	44.30**	46.13	46.14	46.04	46.03**	46.03**	45.89**	
	9	2	44.41**	44.41**	44.35**	45.95	45.92	45.83	45.99**	46.00**	45.74**	

(a) FESX

Models	Rep.	Δ	All indicators			Bollinger bands			Momentum		
			100	50	10	100	50	10	100	50	10
# subsets											
Base Case	0	0	47.46	47.32	47.22	49.1	49.1	48.99	51.61	51.58	51.73
Adaboost	0	0	44.74**	44.73**	44.69**	47.68**	47.68**	47.55**	47.30**	47.18**	47.18**
	3	6	44.59**	44.58**	44.55**	47.41**	47.41**	47.29**	46.25**	46.17**	46.11**
	6	3	44.80**	44.79**	44.66**	47.54**	47.51**	47.39**	46.26**	46.23**	46.11**
	9	2	44.59**	44.58**	44.55**	47.46**	47.45**	47.34**	46.23**	46.15**	46.13**
Bagging	0	0	44.61**	44.59**	44.50**	47.51**	47.49**	47.39**	47.33**	47.21**	47.22**
	3	6	44.54**	44.50**	44.42**	47.19**	47.17**	47.06**	46.10**	46.04**	46.00**
	6	3	44.45**	44.42**	44.37**	47.26**	47.23**	47.08**	46.35**	46.29**	46.19**
	9	2	44.4**	44.38**	44.31**	47.22**	47.22**	47.07**	46.19**	46.14**	46.03**
Logistic regression	0	0	44.55**	44.50**	44.47**	47.60**	47.58**	47.48**	46.85**	46.74**	46.79**
	3	6	44.66**	44.61**	44.62**	47.34**	47.34**	47.22**	46.08**	46.01**	45.99**
	6	3	44.38**	44.29**	44.25**	47.12**	47.12**	46.97**	46.29**	46.23**	46.21**
	9	2	44.65**	44.61**	44.60**	47.10**	47.10**	46.96**	46.25**	46.21**	46.08**
Logitboost	0	0	47.3**	47.31**	47.44**	47.69**	47.65**	47.58**	47.31**	47.17**	47.16**
	3	6	44.70**	44.66**	44.51**	47.21**	47.21**	47.10**	46.31**	46.24**	46.21**
	6	3	44.63**	44.60**	44.48**	47.32**	47.32**	47.18**	46.40**	46.33**	46.24**
	9	2	44.72**	44.71**	44.60**	47.34**	47.34**	47.22**	46.17**	46.13**	46.01**

(b) FDAX

Table 1: Test error (%) by algorithm for a) FESX and b) FDAX. Each algorithm is tested with 0, 3, 6 and 9 repetitions (Rep.) and with a modification of the original parameter (Δ) by 0, 6, 3, and 2 steps respectively. * 5%, ** 1% significance level (p-value) of t-test difference between the base case and each algorithm.

Models # subsets	Rep. Δ	All indicators			Bollinger bands			Momentum		
		100	50	10	100	50	10	100	50	10
Base case	0	-5.26	-5.41	-7.28	6.13	6.37	5.82	0.03	-0.2	-2.6
Adaboost	0	8.41 **	8.84 **	9.12 **	3.79 **	3.43 **	3.65 **	-1.08 **	-1.58 **	-3.06 **
	3	8.41 **	8.84 **	9.12 **	3.79 **	3.43 **	3.65 **	-1.08 **	-1.58 **	-3.06 **
	6	8.41 **	8.84 **	9.12 **	3.79 **	3.43 **	3.65 **	-1.08 **	-1.58 **	-3.06 **
Bagging	9	8.41 **	8.84 **	9.12 **	12.39	11.65	9.90	-1.08 **	-1.58 **	-3.06 **
	0	18.42 **	18.75 **	17.68 **	5.15 *	4.61 **	3.95 **	-1.02 **	-1.59 **	-3.09 **
	3	20.65 **	19.98 **	18.77 **	6.12	5.84 *	4.93 *	-1.05 **	-1.56 **	-3.03 **
Logistic regression	6	20.69 **	20.02 **	18.81 **	6.41	5.48	3.95	-1.05 **	-1.55 **	-3.02 **
	9	17.46 **	17.16 **	16.57 **	6.71 *	6.09 *	4.59 *	-1.13 **	-1.67 **	-3.12 **
	0	13.04 **	12.65 **	11.17 **	3.07 **	2.48 **	1.78 **	-1.08 **	-1.58 **	-3.06 **
Logitboost	3	13.64 **	12.99 **	11.04 **	3.81 *	3.17 **	2.19 **	-0.93 **	-1.45 **	-2.93 **
	6	12.19 **	12.00 **	9.70 **	2.71 *	2.02 *	0.68 **	-0.94 **	-1.46 **	-2.95 **
	9	18.06 **	17.85 **	16.16 **	3.24 *	2.04 *	0.59 *	-1.12 **	-1.54 **	-3.05 **
Logitboost	0	18.6 **	18.58 **	17.43 **	4.29	3.91	2.84	-0.63 **	-1.29 **	-2.54 **
	3	18.22 **	18.06 **	17.30 **	6.53 *	6.22 *	5.50 *	-0.81 **	-1.26 **	-3.21 **
	6	21.41 **	20.9 **	19.34 **	4.78	4.12	2.01	-1.21 **	-1.67 **	-3.85 **
9	20.26 **	20.19 **	18.71 **	4.67	4.32	2.40	-5.56 **	-6.72 **	-10.52 **	

(a) FESX

Models # subsets	Rep. Δ	All indicators			Bollinger bands			Momentum		
		100	50	10	100	50	10	100	50	10
Base Case	0	19.29	18.51	16.88	-7.83	-9.53	-11.57	-1.13	-2.33	-3.37
Adaboost	0	40.89 **	40.41 **	36.99 **	10.49 **	8.75 **	7.56 **	22.81 **	22.61 **	21.20 **
	3	41.27 **	40.7 **	37.58 **	14.52 **	12.83 **	12.62 **	29.53 **	29.18 **	26.68 **
	6	39.14 **	39.09 **	37.14 **	13.28 **	11.32 **	11.09 **	31.1 **	31.59 **	29.58 **
Bagging	9	41.27 **	40.7 **	37.58 **	13.88 **	12.06 **	11.9 **	31.25 **	30.9 **	29.30 **
	0	40.76 **	40.14 **	37.92 **	11.4 **	9.35 **	8.08 **	22.54 **	22.34 **	20.91 **
	3	40.47 **	39.78 **	37.13 **	16.66 **	14.33 **	13.43 **	31.65 **	31.98 **	29.76 **
Logistic regression	6	41.74 **	41.03 **	37.99 **	15.73 **	13.29 **	12.44 **	30.95 **	31.46 **	29.7 **
	9	42.29 **	41.53 **	38.33 **	15.74 **	13.37 **	12.25 **	31.81 **	31.83 **	30.67 **
	0	42.2 **	41.87 **	38.59 **	10.07 **	8.69 **	7.66 **	23.91 **	23.99 **	21.98 **
Logitboost	3	42.94 **	42.94 **	40.13 **	15.14 **	12.87 **	11.07 **	32.03 **	32.11 **	29.53 **
	6	44.55 **	44.42 **	41.69 **	15.97 **	13.4 **	12.35 **	32 **	32.47 **	30.34 **
	9	41.15 **	41.41 **	38.74 **	15.77 **	13.29 **	11.96 **	31.99 **	31.65 **	29.95 **
Logitboost	0	11.85 **	11.71 **	8.59 **	11.52 **	10.05 **	8.97 **	22.48 **	22.31 **	21.02 **
	3	39.78 **	39.14 **	36.54 **	16.41 **	14.33 **	12.48 **	30.44 **	30.25 **	28.07 **
	6	40.19 **	39.43 **	36.75 **	15.81 **	13.63 **	12.05 **	30.58 **	30.88 **	29.22 **
9	38.62 **	37.83 **	35.89 **	15.91 **	13.54 **	11.78 **	31.41 **	31.16 **	29.88 **	

(b) FDAX

Table 2: Sharpe ratio by algorithm for a) FESX and b) FDAX. Each algorithm is tested with 0, 3, 6 and 9 repetitions (Rep.) and with a modification of the original parameter (Δ) by 0, 3, and 2 steps respectively. * 5%, ** 1% significance level (p-value) of t-test difference between the base case and each algorithm.

Seconds between orders	FESX	FDAX
10	23.55	21.43
20	39.33	39.69
30	38.06	38.73
40	35.12	41.59
50	40.17	35.08
60	36.01	34.77
90	36.86	34.61
120	37.48	40.32
180	34.53	35.56
240	34.61	38.42
300	34.36	39.37
360	34.04	36.83
420	34.29	36.83
480	34.24	35.24
540	33.22	36.83
600	33.56	35.24
B&H	1.42	3.02

Table 3: Annualized Sharpe ratio for FESX and FDAX using 21 trading days of March 2009. t-test of mean difference between the underlying return series for the trading strategy and B&H has a significance level (p-value) of 1%.

forecast for two observations that have price changes of 2% and 0.002% will have the same impact in the calculation of the test error, while the first observation adjusted by risk will have more importance in the calculation of the Sharpe ratio.

Logitboost also slightly outperforms bagging in the case of FESX, while bagging outperforms Logitboost using FDAX data. So it does not seem necessary to incur the additional computational cost of bagging when it does not always generate superior results than Logitboost. In general, Logitboost is the best algorithm for FESX while logistic regression and bagging may show better results in some cases for FDAX. Anyway, FESX is the most important index and is much more liquid and traded than FDAX.

We can also observe that the Sharpe ratios of the trading algorithm for FESX are larger than those obtained from the model calibration process. These differences are partially explained by different samples, and the additional use of liquidity indicators. However, the most important considerations are related to the trading strategy itself:

1. The combination of many experts that are reweighted according to their performance, and
2. The trading algorithm that uses the forecast of the price trend to profit from the bid-ask spread.

8. Final comments and conclusions

This paper shows that a trading agent can be calibrated generating multiple versions of the same parameter or technical indicator and then selecting the optimal combination of these parameters. The optimization method could be a learning algorithm with feature selection capability such as Logitboost. Our tests indicate that Logitboost outperforms a model that uses the typical parameters recommended by investment practitioners.

The simulations show that the evaluation of a trading algorithm should be based on risk adjusted return indicators instead of only using accuracy indicators such as the test error. When a risk adjusted return indicator is utilized, every individual investment decision is evaluated by considering its return contribution and its risk impact while the test error gives the same weights to all the observations. So it might be possible that a learning algorithm has a low test error under normal conditions while it may not be able to anticipate major market changes. As a result, its Sharpe ratio might be unsatisfactory. Another algorithm might be able to take advantage of the instability of the market to capture profits even though it may not show the best performance under normal market conditions.

The trading agent algorithm introduced in this paper generated positive returns for two major European index futures (FESX and FDAX) using high frequency data. This trading agent was able to obtain these results combining a learning algorithm that makes the prediction, an expert weighting algorithm that combines experts, a risk management layer that minimizes risky trades, and a trading strategy that uses the prediction to profit from the bid-ask spread.

High-frequency trading has a different market structure and dynamic than daily or monthly trading. Hence, an automated trading strategy that works for longer periods of

time may not work for high-frequency data or it must be adjusted accordingly. Intraday data may not change much in very short periods of time unless it is affected by special events. As a result, a trading agent may need many experts that capture changes in the market behavior in different time periods. Additionally, the trading agent may have to review the order periodically (e.g. every five or thirty minutes) and define minimum quality standards to reduce excessive trading.

Our algorithm can be enriched by the introduction of new types of indicators that capture either events or market relationships that affect market behavior. Recently, several data providers offer machine readable news, parsed and transformed into investment signals, that can be used as new predictive variables or as an input to uncover relationships between companies and economic trends that may improve the forecasting capability of our predictive model.

Acknowledgements

The author thanks Stephanie Hammer, Axel Vischer and the Eurex team in Chicago for providing the data and for discussing initial versions of this research. The author also thanks Ionut Florescu, H. Eugene Stanley, Jeff Nickerson, participants of the Eurex workshop at the University of Chicago, and at the High Frequency data conference at Stevens Institute of Technology for suggestions and informal discussions about the trading algorithm; to OneMarketData, Maxim Chernobayev, Maria Belianina, and Ross Dubin for making OneTick available for this research, and to Patrick Jardine for proof-reading the article. The opinions presented are the exclusive responsibility of the author.

References

- [1] F. Allen, R. Karjalainen, Using genetic algorithms to find technical trading rules, *Journal of Financial Economics* 51 (1999) 245–271.
- [2] J. Arifovic, The behavior of the exchange rate in the genetic algorithm and experimental economies, *Journal of Political Economy* 104 (1996) 510–541.
- [3] W.B. Arthur, J.H. Holland, B. LeBaron, R. Palmer, P. Talyer, Asset Pricing Under Endogenous Expectations in an Artificial Stock Market, in: W. Arthur, S. Durlauf, D. Lane (Eds.), *The Economy as an Evolving Complex System II*, "Addison-Wesley", "Reading, MA", 1997, pp. 15–44.
- [4] J. Barr, T. Tassier, L. Ussher, Symposium introduction, *Eastern Economic Journal* 34 (2008) 421–422.
- [5] J. Barr, T. Tassier, L. Ussher, Introduction to the symposium on agent-based computational economics, *Eastern Economic Journal* 37 (2010) 1–5.
- [6] R. Bates, M. Dempster, Y. Romahi, Evolutionary reinforcement learning in FX order book and order flow analysis, in: *Proceedings of the IEEE International Conference on Computational Intelligence for Financial Engineering*, Hong Kong, March 20-23, 2003, IEEE, Hong Kong, 2003, pp. 355–362.

- [7] D.J. Bodas-Sagi, P. Fernández, J.I. Hidalgo, F.J. Soltero, J.L. Risco-Martín, Multiobjective optimization of technical market indicators, in: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, GECCO '09, ACM, New York, NY, USA, 2009, pp. 1999–2004.
- [8] T. Bollerslev, Generalized autoregressive conditional heteroscedasticity, *Journal of Econometrics* 31 (1986) 307–327.
- [9] J.A. Bollinger, *Bollinger on Bollinger Bands*, McGraw-Hill, New York, 2001.
- [10] L. Breiman, Bagging predictors, *Machine Learning* 24 (1996) 123–140.
- [11] N. Cesa-Bianchi, Y. Freund, D. Haussler, D.P. Helmbold, R.E. Schapire, M.K. Warmuth, How to use expert advice, *Journal of the ACM* 44 (1997) 427–485.
- [12] N. Chan, B. LeBaron, A. Lo, T. Poggio, Agent-based models of financial markets: A comparison with experimental markets, Technical Report 124, MIT Artificial Markets Project, Cambridge, MA, 1999.
- [13] T. Chan, *Artificial markets and intelligent agents*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2001.
- [14] T.S. Chande, S. Kroll, *The New Technical Trader: Boost your profit by plugging into the latest indicators*, John Wiley & Sons, Inc., New York, 1994.
- [15] J.H. Choi, M.K. Lee, M.W. Rhee, Trading s&p500 stock index futures using a neural network, in: Proceedings of the 3rd Annual International Conference on Artificial Intelligence Applications on Wall Street, pp. 63–72.
- [16] J.F. Clayburg, *Four Steps to Trading Success: Using everyday indicators to achieve extraordinary profits*, John Wiley & Sons, Inc., New York, 2001.
- [17] M. Collins, R.E. Schapire, Y. Singer, Logistic regression, adaboost and Bregman distances, *Machine Learning* 48 (2004) 253–285.
- [18] G. Creamer, Y. Freund, A boosting approach for automated trading, *The Journal of Trading* 2 (2007) 84–95.
- [19] G. Creamer, Y. Freund, Automated trading with boosting and expert weighting, *Quantitative Finance* 10 (2010) 401–420.
- [20] M. Dempster, V. Leemans, An automated FX trading system using adaptive reinforcement learning, *Expert Systems with Applications: Special issue on financial engineering* 30 (2006) 534–552.
- [21] M. Dempster, T.W. Payne, Y. Romahi, G. Thompson, Computational learning techniques for intraday FX trading using popular technical indicators, *IEEE Transactions on neural networks* 12 (2001) 744–754.

- [22] L.S. Duke, J.A. Long, An application of the aq machine learning methodology on the stock market, in: S. for Worldwide Interbank Financial Telecommunication S. C (Ed.), Adaptive intelligent systems: proceedings of the BANKAI Workshop, Brussels, Belgium, 12-14 October 1992, Elsevier, Amsterdam, 1993.
- [23] J.F. Ehlers, Rocket Science for Traders: Digital signal processing applications, John Wiley & Sons, Inc., New York, 2001.
- [24] N. Ehrentreich, The Santa Fe Artificial Stock Market Re-Examined - Suggested Corrections, Technical Report, econWPA, 2002.
- [25] N. Fosback, Stock Market Logic: A sophisticated approach to profits on Wall Street, Dearborn Trade Publishing, Chicago, 1991.
- [26] Y. Freund, Y. Mansour, R. Schapire, Generalization bounds for averaged classifiers, *The Annals of Statistics* 32 (2004) 1698–1722.
- [27] Y. Freund, L. Mason, The alternating decision tree learning algorithm, in: Machine Learning: Proceedings of the Sixteenth International Conference, Morgan Kaufmann Publishers Inc., San Francisco, 1999, pp. 124–133.
- [28] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences* 55 (1997) 119–139.
- [29] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: A statistical view of boosting, *The Annals of Statistics* 38 (2000) 337–374.
- [30] D.K. Gode, S. Sunder, Allocative efficiency of markets with zero intelligence traders: Market as a partial substitute for individual rationality, *Journal of Political Economy* 101 (1993) 119–137.
- [31] J. Granville, Granville’s New Key to Stock Market Profits, Prentice Hall, Upper Saddle River, 2000.
- [32] S. Grossman, J. Stiglitz, On the impossibility of informationally efficient markets, *American Economic Review* 70 (1980) 393–408.
- [33] S. Hamid, Z. Iqbal, Using neural networks for forecasting volatility of s&p 500 index futures prices, *Journal of Business Research* 57 (2004) 1116–1125.
- [34] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, Springer, New York, 2003.
- [35] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
- [36] J. Katz, D. McCormick, *The Encyclopedia of Trading Strategies*, McGraw-Hill, New York, 2000.

- [37] K. Kim, Artificial neural networks with feature transformation based on domain knowledge for the prediction of stock index futures, *Int. J. Intell. Syst. Account. Financ. Manage.* 12, 3, Jul. 2004 12 (2004) 167–176.
- [38] K. Kim, Artificial neural networks with evolutionary instance selection for financial forecasting, *Expert system with applications* 30 (2006) 519–526.
- [39] J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, 1992.
- [40] B. LeBaron, An evolutionary bootstrap method for selecting dynamic trading strategies, in: Refenes, A.-P. N., Burgess, A. and Moody, J. eds., *Decision Technologies for Computational Finance*, Proceedings of the Fifth International Conference Computational Finance, Springer-Verlag, New York, 1998, pp. 141–160.
- [41] B. LeBaron, Agent based computational finance: Suggested readings and early research, *Journal of Economic Dynamics and Control* 24 (2000) 679–702.
- [42] B. LeBaron, Empirical regularities from interacting long and short memory investors in an agent-based financial market, *IEEE Transactions on Evolutionary Computation* 5 (2001) 442–455.
- [43] B. LeBaron, W.B. Arthur, R. Palmer, The time series properties of an artificial stock market, *Journal of Economic Dynamics and Control* 21 (1998) 1487–1516.
- [44] M. Lettau, Explaining the facts with adaptive agents: The case of mutual funds flows, *Journal of Economic Dynamics and Control* 21 (1997) 1117–1148.
- [45] N. Littlestone, M. Warmuth, The weighted majority algorithm, *Information and Computation* 108 (1994) 212–261.
- [46] A. Lo, H. Mamaysky, J. Wang, Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation, *Journal of Finance* 4 (2000) 1705–1765.
- [47] J. Moody, M. Saffell, Learning to trade via direct reinforcement, *IEEE Transactions on Neural Networks* 12 (2001) 875–889.
- [48] K. Moriyama, M. Matsumoto, K. ichi Fukui, S. Kurihara, M. Numao, Reinforcement learning on a futures market simulator, *Journal of Universal Compute Science* 14 (2008) 1136–1153.
- [49] L.N. nez, Fitting the control parameters of a genetic algorithm: an application to technical trading systems design, *European Journal of Operational Research* 179 (3) (2007) 847–868.
- [50] M. Pring, *Technical Analysis Explained*, McGraw-Hill, New York, 4 edition, 2002.
- [51] B.R. Routledge, Genetic algorithm learning to choose and use information, *Macroeconomic dynamics* 5 (2001) 303–325.

- [52] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, Boosting the margin: A new explanation for the effectiveness of voting methods, *The Annals of Statistics* 26 (1998) 1651–1686.
- [53] C.J. Sherry, *The New Science of Technical Analysis*, Probus publishing, Chicago, 1994.
- [54] Kyung-shik Shin, Kyoung-jae Kim, I. Han, Financial data mining using genetic algorithms technique: Application to KOSPI 200, in: *Proceedings of the Korea Intelligent Information System Society Conference, 2*, Korea Intelligent Information Systems Society, 1998, pp. 113–122.
- [55] T. Stridsman, *Trading Systems and Money Management*, McGraw-Hill, New York, 2003.
- [56] F. Tay, L. Cao, Application of support vector machines in financial time series forecasting, *Omega* 29 (2001) 309–317.
- [57] N. Towers, A.N. Burgess, Implementing trading strategies for forecasting models, in: Y. S. Abu-Mostafa, B. LeBaron, A. W. Lo and A. S. Weigend (eds.) *Computational finance. Proceedings of the Sixth International Conference on Computational Finance*, New York, USA, January 6-9, 1999, The MIT Press, Cambridge, MA, 2000, pp. 313–325.
- [58] R.R. Trippi, D. DeSieno, Trading equity index futures with a neural network, *The Journal of Portfolio Management* 19 (1992) 27–33.
- [59] R. Tsaih, Y. Hsu, C.C. Lai, Forecasting s&p 500 stock index futures with a hybrid ai system, *Decision support Systems* 23 (1998) 161–174.
- [60] D. Witkowska, D. Marcinkiewicz, Construction and evaluation of trading systems: Warsaw index futures, *International Advances in Economic Research* 11 (2005) 8392.
- [61] E. Zivot, J. Wang, *Modeling Financial Time Series with S-Plus*, Springer, New York, 2003.

Appendix. Investment signals

Technical indicators quantify market trends. We follow Creamer and Freund [19] and [61] in describing the technical analysis indicators. Additional useful references about technical analysis and trading are [36, 50, 14, 53, 55, 23, 16].

We index the trading period by $t = 1, 2, \dots$. We denote by P_t^t , P_t^b , P_t^a , P_t^m , VOL_t^t , VOL_t^b , and VOL_t^a the trade, bid, ask, and mid price, and the trade, bid, and ask volume. We are trying to predict the trend of P_t^t .

We eliminate the lower index when we wish to refer to the whole sequence, i.e. P^t refers to the whole sequence P_1^t, P_2^t, \dots

Many of the technical indicators incorporate the following simple or exponentially weighted moving averages of prices. Let \mathbf{X} denote a time sequence X_1, X_2, \dots . The **simple moving average** is defined as

$$\mathbf{SMA}_t(\mathbf{X}, n) = \frac{1}{n} \sum_{s=0}^{n-1} X_{t-s},$$

and the **exponentially weighted moving average** is defined as

$$\mathbf{EMA}_t(\mathbf{X}, n) = \lambda \sum_{s=0}^{\infty} (1 - \lambda)^s X_{t-s}; \quad \lambda = \frac{2}{n + 1}.$$

$\mathbf{EMA}_t(\mathbf{X}, n)$ can be calculated using a simple update rule:

$$\mathbf{EMA}_t(\mathbf{X}, n) = \lambda X_t + (1 - \lambda) \mathbf{EMA}_{t-1}(\mathbf{X}, n).$$

Rules associated to technical indicators are denoted by “rule” and follow by an identification number. The input to our trading agent algorithm includes both signals and *normalized* indicators.

Additionally, we recalculate a selected group of indicators and their rules with three different values of the main parameters that are close to the industry practice. So, our learning system should be able to select the optimal combination of indicators and parameters. We also include ratios of the indicators which generally are calculated as the indicator divided by its moving average. Most of these ratios are part of the trading rules. However, we include the ratios by themselves so that our learning system finds its own rules.

In the following table we describe the technical indicators. The parameters of each indicator are in parentheses. Most of the parameters used refer to the length of the period (n) selected to calculate the indicator. In case of exponential moving average, the parameter used is λ which also depends of n . We have assigned parameters which are typically used in the industry for each indicator.

Investment and technical indicators

Variable	Description	Calculation detail
Price indicators:		
$EMA_t^c(\lambda)$	Exponential moving average of a time series P^t .	$EMA_t(P^c, \lambda)$ where $\lambda = 0.9, 0.84, \text{ and } 0.78$
$rule1_t$	Exponential moving average to price (P_t^t).	$\frac{EMA_t^c(\lambda)}{P^c}$
$SMA_t^c(n)$	Simple moving average of the last n observations of a time series P^t .	$SMA_t(P^c, n)$ where $n = 10, 16 \text{ and } 22$
$rule2_t$	Simple moving average to P_t^t	$\frac{SMA_t^c(n)}{P^c}$ where $n = 10, 16 \text{ and } 22$
Bollinger bands:	Using the moving average or the median band ($Boll_t^m(n)$) as the reference point, the upper and lower Bollinger [9] bands ($Boll_t^u(n)$ and $Boll_t^d(n)$ respectively) are calculated in function of s standard deviations. When price crosses above (below) the upper (lower) Bollinger band, it is a sign that the market is overbought (oversold). Technical analysts typically calculate Bollinger bands using 20 periods for the moving average and 2 standard deviations.	$Boll_t^m(n) = SMA_t^c(n)$
$Boll_t^u(n)$	Upper Bollinger band	$Boll_t^m(n) + s\sigma_t^2(n)$ where $s=2, n=20, 26 \text{ and } 32$
$Boll_t^d(n)$	Lower Bollinger band	$Boll_t^m(n) - s\sigma_t^2(n)$ where $s=2, n=20, 26 \text{ and } 32$
$PBoll_t^u(n)$	Price to upper Bollinger band	$\frac{P_t^t}{Boll_t^u(n)}$
$PBoll_t^d(n)$	Price to lower Bollinger band	$\frac{P_t^t}{Boll_t^d(n)}$
$rule3_t$	Bollinger trading rule	$\begin{cases} Buy & \text{if } P_{t-1}^t \geq Boll_t^d(n) \text{ and } P_t^t < Boll_t^u(n) \\ Sell & \text{if } P_{t-1}^t \leq Boll_t^d(n) \text{ and } P_t^t > Boll_t^u(n) \\ Hold & \text{Otherwise} \end{cases}$
Momentum and oscillation indicators:		
$MOM_t(n)$	Momentum: price (P_t^t) change in the last n periods. When it crosses above (below) zero, it indicates that trend is up (down). The default value of n is 12.	$P_t^t - P_{t-n}^t$ where $n = 12, 18, \text{ and } 24$
$MomEMA_t(n, \lambda)$	Momentum $EMA_t(MOM_t(n), \lambda)$	to $\frac{MOM_t(n)}{EMA_t(MOM_t(n), \lambda)}$ where $n = 12, 18, \text{ and } 24$ and $\lambda = 0.75$

$rule4_t$	Momentum trading rule	$\left\{ \begin{array}{l} Buy \quad \text{if } MOM_{t-1}(n) \leq \mathbf{EMA}_t(MOM_t(n), \lambda) \\ \quad \text{and } MOM_t(n) > \mathbf{EMA}_t(MOM_t(n), \lambda) \\ Sell \quad \text{if } MOM_{t-1}(n) \geq \mathbf{EMA}_t(MOM_t(n), \lambda) \\ \quad \text{and } MOM_t(n) < \mathbf{EMA}_t(MOM_t(n), \lambda) \\ Hold \quad \text{Otherwise} \end{array} \right.$
$ACCEL_t(n)$	Acceleration: difference of price change. The default value of n is 12.	$MOM_t(n) - MOM_{t-1}(n)$ where $n = 12, 18, \text{ and } 24$
$rule5_t$	Acceleration trading rule	$\left\{ \begin{array}{l} Buy \quad \text{if } ACCEL_{t-1}(n) + 1 \leq 0 \\ \quad \text{and } ACCEL_t(n) + 1 > 0 \\ Sell \quad \text{if } ACCEL_{t-1}(n) + 1 \geq 0 \\ \quad \text{and } ACCEL_t(n) + 1 < 0 \\ Hold \quad \text{Otherwise} \end{array} \right.$
$ROC_t(n)$	Rate of change: rate of change of P_t^t . Technical analysts recommend using 10 periods to calculate this indicator.	$\frac{P_t^t - P_{t-n}^t}{P_{t-n}^t} \cdot 100$ where $n = 10, 16, \text{ and } 22$
$rule6_t$	ROC trading rule	$\left\{ \begin{array}{l} Buy \quad \text{if } ROC_{t-1}(n) \leq 0 \text{ and } ROC_t(n) > 0 \\ Sell \quad \text{if } ROC_{t-1}(n) \geq 0 \text{ and } ROC_t(n) < 0 \\ Hold \quad \text{Otherwise} \end{array} \right.$
$MACD_t(s, f)$	Moving average convergence divergence: difference between two moving averages of slow and fast periods (s, f). $MACD_t(s, f)$ is regularly calculated using 26 (s) and 12 (f) periods.	$\mathbf{EMA}_t(P^t, s) - \mathbf{EMA}_t(P^t, f)$ where $f = 12, \text{ and } s = 18, 24, \text{ and } 30$
$MACDS_t(s, f, n)$	MACD signal line: moving average of $MACD_t(s, f)$ of past n periods. A buy (sell) signal is generated when the $MACD_t(s, f)$ crosses above (below) the signal line or a threshold.	$\mathbf{EMA}_t(MACD_t(s, f), n)$ where $f = 12, n = 9, \text{ and } s = 18, 24, \text{ and } 30$
$rule7_t$	MACD trading rule	$\left\{ \begin{array}{l} Buy \quad \text{if } MACD_{t-1}(s, f) \leq MACDS_t(s, f, n) \\ \quad \text{and } MACD_t(s, f) > MACDS_t(s, f, n) \\ Sell \quad \text{if } MACD_{t-1}(s, f) \geq MACDS_t(s, f, n) \\ \quad \text{and } MACD_t(s, f) < MACDS_t(s, f, n) \\ Hold \quad \text{Otherwise} \end{array} \right.$
$MACDR_t(s, f, n)$	$\frac{MACD_t(s, f)}{MACDS_t(s, f, n)}$	to $\frac{MACD_t(s, f)}{MACDS_t(s, f, n)}$

$RSI_t(n)$	Relative strength index: compares the periods that stock prices finish up against those periods that stock prices finish down. Technical analysts calculate this indicator using 9, 14 or 25 periods. A buy signal is when $RSI_t(n)$ crosses below a lower band of 30 (oversold), and a sell signal when $RSI_t(n)$ crosses above an upper band of 70 (overbought)	$100 - \frac{100}{1 + \frac{SMA_t(\mathbf{P}_n^{up}, n_1)}{SMA_t(\mathbf{P}_n^{dn}, n_1)}}$ <p>where $n_1 = 8, 14, \text{ and } 20$ and n is the length of the time series</p> $P_t^{up} = \begin{cases} P_t^t & \text{if } P_t^t > P_{t-1}^t \\ \text{empty} & \text{Otherwise} \end{cases}$ $P_t^{dn} = \begin{cases} P_t^t & \text{if } P_t^t < P_{t-1}^t \\ \text{empty} & \text{Otherwise} \end{cases}$ $\mathbf{P}_n^{up} = (P_{t-n}^{up}, P_{t-n+1}^{up}, P_{t-n+2}^{up}, \dots, P_t^{up})$ $\mathbf{P}_n^{dn} = (P_{t-n}^{dn}, P_{t-n+1}^{dn}, P_{t-n+2}^{dn}, \dots, P_t^{dn})$
$rule8_t$	RSI trading rule	$\begin{cases} \text{Buy} & \text{if } RSI_{t-1}(n) \geq 30 \text{ and } RSI_t(n) < 70 \\ \text{Sell} & \text{if } RSI_{t-1}(n) \leq 30 \text{ and } RSI_t(n) > 70 \\ \text{Hold} & \text{Otherwise} \end{cases}$
Volatility and return indicators:⁷		
$\hat{r}_{t+1}, \sigma_t^2$	Next period return and volatility calculated using $GARCH(1, 1)$ [8].	
Sharpe ratio	Risk adjusted return.	\hat{r}_t / σ_t^2
Volume indicators:		
OBV_t	On balance volume: this indicator was developed by Granville [31] to evaluate the impact of positive and negative volume flows. OBV_t adds the volume when the close price has increased and subtracts it when the close price has decreased. A sign of market reversal is when OBV_t diverges with the price movement.	$\begin{aligned} \text{if } P_t^t > P_{t-1}^t & \quad OBV_t = OBV_{t-1} + VOL_t^t \\ \text{if } P_t^t < P_{t-1}^t & \quad OBV_t = OBV_{t-1} - VOL_t^t \end{aligned}$
$NVI_t \text{ and } PVI_t$	Negative and positive volume index: these indicators were introduced by Fosback [25] as signals of bull markets. NVI_t (PVI_t) concentrates on periods when volume decreases (increases). The rationality is that “informed” investors take positions on periods when volume decreases, while the “uninformed” investors take position on periods when the volume increases. NVI_t (PVI_t) is calculated as the cumulative sum of $ROC_t(n)$ when volume decreases (increases). Fosback maintains that there is 95% probability that a bull market is going to develop when NVI_t crosses above its one year moving average, and 67% probability of a bear market when PVI_t crosses below its one year moving average.	$\begin{aligned} \text{if } VOL_t^t < VOL_{t-1}^t & \quad NVI_t = NVI_{t-1} + ROC_t(n)NVI_{t-1} \\ & \quad PVI_t = PVI_{t-1} \\ \text{if } VOL_t^t \geq VOL_{t-1}^t & \quad PVI_t = PVI_{t-1} + ROC_t(n)PVI_{t-1} \\ & \quad NVI_t = NVI_{t-1} \end{aligned}$ <p>where $n=1$</p>

$rule9_t$	Negative volume index trading rule.	$\begin{cases} Buy & \text{if } NVI_{t-1} \leq \mathbf{SMA}_t(NVI, l) \\ & \text{and } NVI_t > \mathbf{SMA}_t(NVI, l) \\ Hold & \text{Otherwise} \end{cases}$
$rule10_t$	Positive volume index trading rule.	$\begin{cases} Buy & \text{if } PVI_{t-1} \leq \mathbf{SMA}_t(PVI, l) \\ & \text{and } PVI_t > \mathbf{SMA}_t(PVI, l) \\ Sell & \text{if } PVI_{t-1} \geq \mathbf{SMA}_t(PVI, l) \\ & \text{and } PVI_t < \mathbf{SMA}_t(PVI, l) \\ Hold & \text{Otherwise} \end{cases}$ <p>where $l = 10$</p>
$nviSMA_t(l)$	NVI_t to $\mathbf{SMA}_t(NVI, l)$	$\frac{NVI_t}{\mathbf{SMA}_t(NVI, l)}$, where $l = 10$
$pviSMA_t(l)$	PVI_t to $\mathbf{SMA}_t(PVI, l)$	$\frac{PVI_t}{\mathbf{SMA}_t(PVI, l)}$, where $l = 10$
$PV_t(n)$	Price-volume trend: this indicator is similar to OBV_t . It calculates a cumulative total of volume where the portion of volume added/subtracted is given by the increase or decrease of close prices in relation to the previous period.	$\sum_{t=1}^n VOL_t^t \cdot ROC_t(n_1)$ <p>where $n_1 = 1$ and n is the length of the time series.</p>
Liquidity indicators⁸	indicators	
ES_t	Effective spread	$2 * D_t * (P_t^l - P_t^m)$, where D_t if trade is buy 1, sell -1
RS_t	Realized spread	$2 * D_t * (P_t^l - P_{t+300}^m)$, where time is in seconds
VT_t	Value trade	$P_t^l * VOL_t^t$
SVT_t	Signed value trade	$D_t * P_t^l * VOL_t^t$
STS_t	Signed trade size	$D_t * VOL_t^t$
dDI_t	Depth imbalance as difference	$\frac{D_t * (VOL_t^a - VOL_t^b)}{VOL_t^a + VOL_t^b}$
DI_t	Depth imbalance as ratio	$(\frac{VOL_t^a}{VOL_t^b})^D$
PES_t	Proportional effective spread	$\frac{100 * ES_t}{P_t^m}$
PRS_t	Proportional realized spread	$\frac{100 * RS_t}{P_t^m}$
PI_t	Price impact	$\frac{ES_t - RS_t}{2}$
PPI_t	Proportional price impact	$\frac{100 * PI_t}{P_t^m}$
HTS_t	Half traded spread	$D_t * (P_t^l - P_{t+300}^m)$, where time is in seconds
$PHTS_t$	Proportional half traded spread	$\frac{HTS_t}{P_t^m}$
$RetSQ_t$	Squared log return on trade prices	$(\log(P_t^l) - \log(P_{t-1}^l))^2$

$RetAbs_t$	Absolute log return on trade prices	$ \log(P_t^t) - \log(P_{t-1}^t) $
QS_t	Quoted spread	$P_t^a - P_t^b$
PQS_t	Proportional quoted spread	$\frac{QS_t}{P_t^m} * 100$
$LogQS_t$	Quoted spread	$\log(\frac{P_t^a}{P_t^b})$
$LogSize_t$	Log quoted spread	$\log(VOL_t^a) - \log(VOL_t^b)$
$QSlope_t$	Quoted slope	$\frac{QS_t}{LogSize_t}$
$LogQSlope_t$	Log quoted slope	$\frac{\log(QS_t)}{LogSize_t}$
$MQRetSQ$	Squared midquote returns	$(\log(P_t^m) - \log(P_{t-1}^m))^2$
$MQRetAbs$	Absolute midquote returns	$ \log(P_t^m) - \log(P_{t-1}^m) $

⁷GARCH is also another indicator of volatility.

⁸Only used for trading algorithm. Liquidity indicators are not used for the model calibration tests described in section 6.1.