Investigating Optimal Regimes for Prediction in the Stock Market

Rodrigo Corbelli
Department of Eletric Engineering
PUC-Rio
Rio de Janeiro, Brazil
rodrigocorbelli@gmail.com

Marley Vellasco
Department of Eletric Engineering
PUC-Rio
Rio de Janeiro, Brazil
marley@ele.puc-rio.br

Álvaro Veiga
Department of Eletric Engineering
PUC-Rio
Rio de Janeiro, Brazil
alvf@ele.puc-rio.br

***Abstract*—Forecasting stock prices in the market its known to be an extremely difficult task, where even the predictability of the series itself is a controversial matter. The present study investigates the existence of periods within the series more suitable for prediction, and whether the identification and exploitation of those periods could be learned from data. In order to do that, the Predictability Crawler (P-Craw) framework is proposed. The technique uses optimizations routines such as the Particle Swarm Optimization (PSO) or Genetic Algorithms (GA) to select subsets of historical data where statistical learning algorithms can be more efficiently trained. When tested against simulated data, The P-Craw is able to reliably identify the optimal subsets in scenarios ranging from 40% to 100% of predictable samples in the data. To access if the framework brings any improvement when used in a real world scenario, it is tested in a dataset containing intraday data from the Brazilian stocks exchange (BOVESPA). When benchmarked against training with all the samples for the series in the BOVESPA dataset the use of the framework is able to significantly raise the Correct Directional Changes (CDC) of the trained models while reducing the Mean Absolute Error (MAE) in up to 19%.**

***Index Terms*—Stock Market, Predictability, Genetic Algorithms, Particle Swarm Optimization, Time Series**

I. Introduction

In the financial market, the prediction of price series means financial profit, so the matter of how to accurately compute those forecasts is always on the spotlight. In order to tackle this challenge, two schools of thought are generally invoked: Technical and Fundamentalist [1]–[3]. To the Fundamental school of analysis, the intrinsic value of a stock paper is the crucial factor, therefore this school pays attention to indicators that go from the financial health of the company to government regulation policies on the sector and macroeconomic data. The Technical branch, on the other hand, looks only at previous movements in price charts and graphics to guide their beliefs concerning future behavior.

In modern prediction models, both Technical and Fundamentalist indicators are used in machine learning setups for prediction [4]–[7]. Even with the use of sophisticated techniques, there is no clear winner in this task, and the most difficult benchmark to beat remains the random walk model.

A possible explanation for the inherent difficulty of the area lies in the Efficient Market Hypothesis [8]. The hypothesis states that all the information regarding an asset is already incorporated in it's price at any given time, therefore it is impossible to consistently beat the market. Not all literature agrees with this hypothesis though, and a more recent conjecture, the Adaptive Market Hypothesis (AMH), speculates a dynamic evolution of efficiency [9]. With this evolution it is possible that a specific strategy or model work at some periods but perceives only noise at others. This reality can be a problem for the regular supervised learning procedures used in machine learning [10]. Namely, the use of all available data in training can incorporate noisy samples that don't offer any structure to be learned. Those samples can worsen the final performance instead of helping. Noisy observations can degrade not only the training stage, but also mask the performance when used in out-of-sample evaluation. To study this issue, different proxies for market efficiency have been proposed [11]–[14] and enhancement has been uncovered when training and evaluation happen in respect to a selected "predictable" sub-set of data [11], [13].

This work proposes a general framework to address the selection of predictable periods. This framework is called the Predictability Crawler (P-Craw). The P-Craw is meant to be an additional step to be performed when training a statistical learning model. This step filters which data points to be used in the training and evaluation of the model, removing noisy observations that can degrade performance. In the present study, a predictable series is one where a model can be trained on past data to decrease uncertainty about future movements. This definition describes predictability of a series in regard to a specific statistical model. It can be the case that the series is considered predictable in respect to one model but unpredictable to another one.

The rest of this paper is organized as follows. Section II discusses how the concept of predictability and multiple regimes evolved in the price series literature. Section III presents the proposed framework. Section IV describes the experiments using simulations to test the framework's accuracy in finding predictable intervals. Tests with real world data are presented in Section V. Finally, the paper is concluded in Section VI.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

II. Background

In his seminal paper [8] Fama introduced the Efficient Market Hypothesis (EMH). In the argument, the fierce competition for profits assures that any new knowledge is immediately incorporated. With the information already reflected in price, is impossible for a trader to create a strategy that consistently beats the market. An efficient market is an unpredictable one.

Not all studies agree with this statement though. According to [15] there are costs inherent to information gathering and arbitrage, and if there was no excess return to be made in exchange for this cost there would be little reason for traders to trade, causing the market to eventually collapse. The payoff of information gathering would be directly linked to the predictability of the market, and an equilibrium would arise from players paying less attention to saturated markets, lowering those markets efficiency levels, and focusing on new assets, raising those assets efficiency.

More Recently, Lo [9] proposed a new paradigm, the Adaptive Market Hypothesis (AMH). In the AMH each agent has constrained knowledge acquired through past experiences. Those agents would then find the best solution constrained to their current beliefs, fighting for resources (economic gains), learning and shifting their preferences towards different financial assets and arbitrage strategies. This evolution entails different predictability in different markets at different times in respect to different strategies, and thus allows for profitable opportunities to exist in certain periods. The theory has gained force in light of evidence such as changing correlation coefficients over time in price series and trading techniques that showed vanishing performance once they were published in academic studies [16].

The modelling of the market as a constant switching between different regimes is not new, and import studies are mentioned next to contextualize how the idea evolved into the formulation presented in the current work.

A. Regime Switching for Trend Identification

A milestone of the regime-switching paradigm is in the work of Hamilton and Engel [17], whose objective was to correctly identify regimes of price trends. A simple Markov Chain model was proposed to address the identification of those trends in the dollar exchange rate series. In their proposal is postulated the existence of a latent variable S which can take the values 1 or 2. The conditional distribution of returns is then $N(\mu_1, \sigma_1)$ or $N(\mu_2, \sigma_2)$ depending on the current value of S, which follows a first-order Markov process. Although the study incorporates the idea of regime switching, it assumes one ever-functioning strategy and constant predictability of the series.

B. Regime Switching for Strategy Identification

In [18] the price change at the dollar exchange rate series e_t after the reveal of new information at instant τ is studied. One of the goals of the study is to develop a estimator to detect whether this new data will influence the equilibrium in the market or not. The authors arrive at a biased yet consistent estimator to be used after the event has occurred, which depends of two interval sizes ρ_1 and ρ_2, with $\rho_2 > \rho_1$.

$$E[\Delta e_{t+1}] > 0, \text{ if } \frac{1}{\rho_1}\sum_{j=0}^{\rho_1} e_{t-j} > \frac{1}{\rho_2}\sum_{j=0}^{\rho_2} e_{t-j} \tag{1}$$

Equation (1) is the moving average indicator of the Technical Analysis, and the derivation is only valid for periods where new information is incorporated. According to the study, those moments would be better predicted by the indicator in (1), while Fundamental Analysis would be preferred otherwise. To identify when to use each strategy, the authors use the variance of the time series, with the regimes detected by the same modelling proposed in II-A.

In their experimental setup, it is found that the technical indicator influence is statistically significant in the assigned regimes. In this approach the market is modelled as passing by different dynamics, where one given strategy can only work if coupled with the right timing.

C. Regime Switching for Predictability Identification

In [11] the nature of regime switching in the market receives a different interpretation. The predictability of the prices series itself is postulated to evolve, and the Hurst Exponent is used as a proxy for it. Periods with Hurst Exponent equal to 0.5 are labelled as random, while values above 0.65 are considered to exhibit a trend-reinforcing pattern which can be exploited for prediction.

In this study, the Hurst Exponent is computed for the whole dataset. The data is split into two groups, one classified as predictable and the other one as unpredictable. Both groups are split into training and test sets. In each group a Neural Network is trained in the training set and evaluated in the test set, and the Mean Absolute Scaled Error (MASE) is used as an error metric of the process. The group considered predictable by the Hurst Exponent proxy achieved statistically significant lower MASE than the one considered to be random.

III. Predictability Crawler

The Predictability Crawler (P-Craw) is defined to adapt the training of a statistical model to a situation where some data points might not be beneficial to the process. In the context of stock market prediction, it searches for predictable moments in the series and can be compared to the approach of II-C. In contrast, it allows for greater flexibility in choosing how to search for the predictable periods. The P-Craw is composed of a series of building blocks.

First, the selector S chooses a subset of the available data composed of predictable periods. Once the best subset $\mathbb{P}$ is defined, it is used in two different training tasks. The first is to fit a prediction model M using only the samples in the selected subset. The second is to train a classifier C, responsible for labeling out-of-sample observations into predictable or unpredictable ones. The latter uses all the available samples, with the target to be learned being whether they are present in the selected subset or not. By training to reproduce the

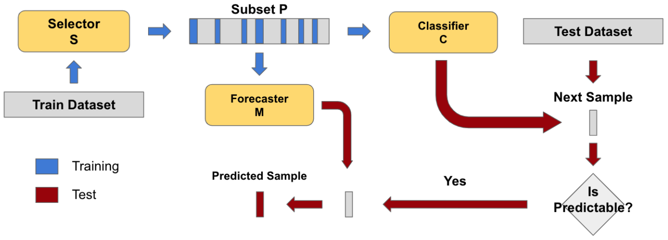


Fig. 1. Proposed Architecture

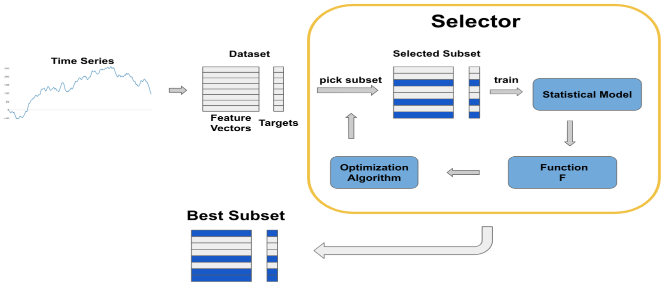


Fig. 2. Optimization Procedure

classification of the selected samples, the classifier learns to differentiate predictable samples from the rest.

To test the performance of the method, the P-Craw turns the out-of-sample observations into features and feed them to the classifier. The vector is then selected as predictable or not. In the positive case, the trained forecaster M is used to perform the actual prediction. If the classification returns negative, the process discards that samples, moving to analyze the next one. Fig. 1 depicts the process, and the following sections describe each building block in details.

A. Selector

The selector's task is to find the subset $\mathbb{P}^{\star}$ in the set of all possible subsets $\mathbb{S}$ of $\mathbb{D}$ that represents the "most predictable" periods available. Implicit in this task is the necessity to compare two possible subsets and be able to say which one better represents predictable periods. This calls for a function $F : \mathbb{S} \in \mathbb{D} \mapsto \mathbb{R}$, used to rank possible sets. With this function defined, the problem now becomes the optimization procedure described in (2).

$$\mathbb{P}^{\star} = \arg\max_{\mathbb{S} \in \mathbb{D}} F(\mathbb{S}) \tag{2}$$

This function needs to score subsets in respect to their ability to "be explained" by a statistical model. In order to carry that, the time series is represented as feature and target matrices. The scoring process must be paired with a specific statistical model to be trained on, and the trained model is used to compute F. Once this scoring mechanism is settled, an optimizer search amongst the space of possible subsets to find the solution of (2). Fig. 2 illustrates the process, which is composed of three main entities. The statistical model chosen, the fitness function F, and the optimization algorithm. The statistical model can be any machine learning procedure suited to the forecasting task. The other two parts are described next.

B. The Fitness Function

With the statistical model trained at a specific subset, an error metric is used to access how much the model could adjust to the trained data. The MASE error metric is selected and computed using a 3-fold cross-validation scheme [10], with the resulting metric being referred to as $MASE_{CV}$. Using the $MASE_{CV}$ directly as a fitness score might lead the optimization to search a period where the model can learn correctly, but it does not provides incentive to explore further and find all the possible predictable moments. To incorporate this search, a measure of subset size is added. The chosen measure is the proportion of the size of subset $\mathbb{P}$ in respect to the whole dataset $\mathbb{D}$, $\frac{|\mathbb{P}|}{|\mathbb{D}|}$. This choice not only has the same order of magnitude as the $MASE_{CV}$, but also ensures a metric unrelated to the dataset size. The constant multiplying this new term influences a lot on the performance of the framework, and was calibrated by experiments with different values between 0.1 and 1. Equation (3) displays the proposed function.

$$F(\mathbb{P}, \mathbb{D}) = (1 - MASE_{CV}) + 0.3 * \frac{|\mathbb{P}|}{|\mathbb{D}|} \tag{3}$$

Besides the mathematical representation, the definition of the fitness function also needs a minimum number of samples n in the chosen subset. If a model trained with less than n samples is evaluated, it is assigned the smallest value possible, so that any interval with a number of samples greater than n is preferred over it.

C. Optimization Algorithm

Gathering what was exposed, it is possible to assign fitness scores to each possible candidate subset. The selector then proceeds to choose which new subsets to evaluate, using what it learned from the previous choices. The optimization procedure is the algorithm responsible to make this choice. The P-Craw admits a number of different optimization procedures, such as Genetic Algorithms (GA) [19], Ant Colony Optimization (ACO) [20], Intelligent Water-Drops Algorithm (IWD) [21] and others. In the present study the Particle Swarm Optimization (PSO) [22] is used in two versions, which are compared against each other.

1) Particle Swarm Optimization (PSO): In the Particle Swarm Optimization, each solution is represented by a position and assigned a velocity vector, and so the solutions "wander" through the parametric topology. At each iteration the particles update their velocities based on the best position they have already encountered and the best position encountered by their neighborhood.

In order to use the PSO, the time-series is divided in groups of size equivalent to 2% the dataset size. Each group is assigned a probability between 0 and 1. A particle to the PSO algorithm in the present context is then a vector of probabilities

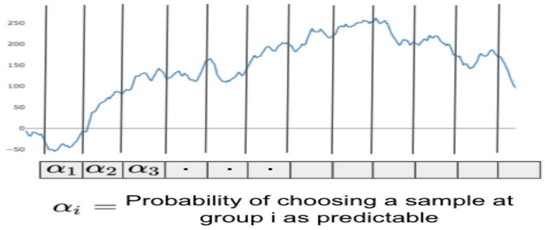


Fig. 3. PSO Particle Representation

representing each group. Every time the fitness value of a particle is computed, a sampling occurs where each sample is considered predictable with a probability equal to the one assigned to the group it belongs. The particles sampled as predictable at that sampling form the subset chosen to train the statistical model and compute the fitness function. This way, if a interval has a parameter of 0.5, each sample in that group has a 50% probability if being included in the subset. Every time that solution is evaluated, this sampling occurs to every group. This way, evaluating a particle different times will produce different results, as different samplings will arise. When the final particle is elected as the best solution, every sample in every group with probability above 0.5 is chosen as predictable. Fig. 3 illustrates the representation.

2) Perturbed PSO: In the classical PSO, the optimization occurs in a parameter space that represents a possible solution in a deterministic fashion. In the present context, due to the categorical nature of the problem (groups being only allowed to be either present or not), the parameters optimized are used to define probability functions from which the true parameters are sampled. The parameter in this case is the probability of choosing an individual sample as predictable within a group. This approach is used to propose a modification to the PSO algorithm. In this version, the optimization is not performed on the problem parameters. Instead, those parameters are represented by probability functions, and the parameters describing those functions are optimized instead.

In the context of the proposed PSO, the sampling happens every time a particle position is evaluated. When that happens, a particle with a value of 0.35 for a given group may, for example, choose half the samples of the group to be included in the subset. In the Perturbed PSO variation, the position of the particle is altered to match the percentage of predictable samples chosen in each group at the last sampling. For example, in the previous case with half the samples in the group chosen, the value of that group would be altered from 0.35 to 0.5 after the sampling. This way, the particle suffers a random perturbation in it's position every time it has it's value computed. The general approach is represented in Fig. 4. The perturbation on the probability and particle position on step 2 is what differentiates the Perturbed and Regular PSO in the present context.

D. Forecaster

With the chosen subset at hand, the samples are used to train the forecasting module in a supervised learning setup. The task is to train a statistical model only in the set proposed by the selector. This model doesn't necessarily has to be the same as the one used in the selector's optimization procedure. The predictability identification and series forecasting are totally separated tasks in the P-Craw methodology. Not only the model can be different, but even the feature representation can change.

E. Classifier

Once the best subset has been identified, a label is assigned to each row of the feature matrix representing whether it is present in it or not (in other words, if it is considered as predictable or not). This label is used as a target for a new task. This time, supervised learning is used to train a model capable of identifying new samples on the test phase which posses the same dynamic as the ones in the selected subset. As in the forecaster, the features used don't need to be the same used in any other building block of the framework. Fig. 5 displays the training process.

IV. Simulation Benchmark

A simulated series is created to help investigate the accuracy of the proposed framework in identifying predictable periods. In this simulation, three different regimes are concatenated together. The first one follows an ARIMA(1, 1, 0) process when transformed by the logarithm function, with coefficient $\phi = .7$ and $\sigma = 2$. The process is displayed in (4), where y_t are the values of the time series.

$$\begin{aligned} \Delta \log y_t &= \phi \Delta \log y_{t-1} + \epsilon_t \\ \epsilon_t &\sim N(0, \sigma) \end{aligned} \tag{4}$$

The other two regimes are random walks when transformed by the logarithm function. The innovations of those regimes

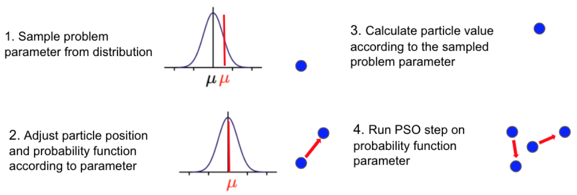

1. Sample problem parameter from distribution
μ μ
3. Calculate particle value according to the sampled problem parameter
2. Adjust particle position and probability function according to parameter
μ
4. Run PSO step on probability function parameter

Fig. 4. Perturbed PSO Steps

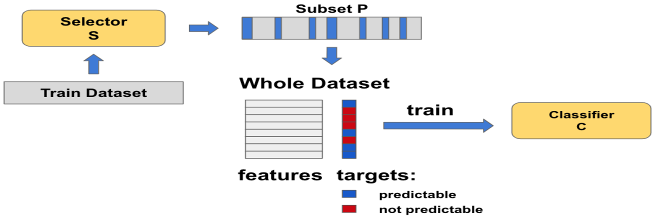


Fig. 5. Training of the Classifier

have mean $\mu = 0$ and variances $\sigma_1 = 2$ and $\sigma_2 = 6$ respectively, as summarized in (5).

$$\Delta \log y_t = \epsilon_t$$
$$\epsilon_t \sim N(0, \sigma_i),\ i = 1, 2 \tag{5}$$

To create the feature representation, a vector of past returns is assembled using the logarithm of the last 15 observations. Every simulation generated has 2015 observations, creating a feature and target representations of 2000 entries.

Before proceeding to the next discussions, the concept of purity needs to be introduced. In the current study, the purity of a dataset will be defined as the percentage of predictable regimes present in it (AR(1) in the simulation case). If a simulation has 300 predictable samples and 700 unpredictable ones, it has a purity of 30%. Subsets of a given simulation can have a different purity and so it is important to differentiate between subset and simulation purity. When the simulation purity is discussed, it will be referred to as total purity to emphasize its calculation regarding all the samples in the simulation. Simulations with different total purity are generated. In each new simulation, a reference signal is generated together with the X and Y feature matrices, with the predictable rows classified as 1 and the others as 0.

As discussed previously, the fitness function needs to be associated with a specific statistical model. For the simulation benchmark a simple linear regression is used together with a feature selection procedure performed using the LASSO algorithm [10].

A. Optimization Setup

On a normal evaluation of an optimization procedure, two quantities are of interest: the best and mean fitness value of the population for each iteration. In this study, two metrics will be added. The first is the purity for the best solution at each generation (again, not to be misinterpreted as the total purity of the simulation) and the second is the relative size of the best solution.

The relative size is the ratio between the subset size and the number of predictable periods present in the simulation, being different from the quantity $\frac{|\mathbb{P}|}{|\mathbb{D}|}$. If a simulation possesses 2000 observations but only 600 predictable ones and a given subset has size 300, the relative size is $\frac{300}{600} = 0.5$. This makes it possible to identify the optimal subset in any situation as the one with purity 1 and relative size 1, regardless of the total purity and size of the simulation. The relative size can be greater than 1 if the subset size is greater than the number of predictable samples.

The purity and relative size metrics relate with the exploitation and exploration properties of the algorithm. The purity metric at each generation conveys how effectively the method was to discover predictable samples, but a high purity might be found in a relatively small portion of the data. The relative size is a metric of how extensively the algorithm explored the possible subsets. For example, if the purity is close to 1 but the relative size is significantly bellow 1, the procedure was able to achieve a predictable solution, but not to sufficiently explore the possible subsets.

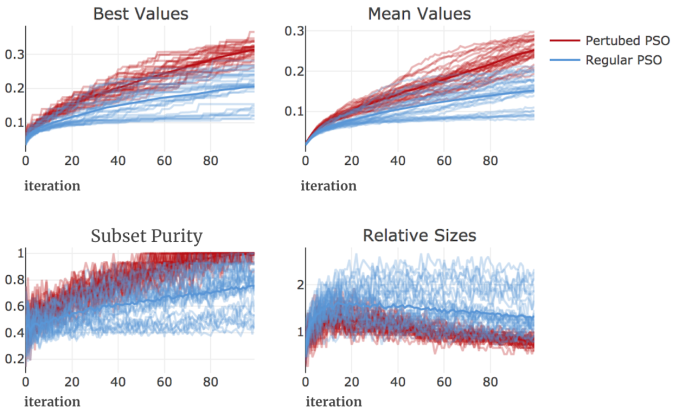


Fig. 6. PSO Evolution - Perturbed vs Regular

In the subsequent comparisons, every configuration is run 30 times, and best fitness value, mean fitness value, purity and relative size are compared.

B. Particle Swarm Optimization

The PSO implementation used is based on [23], using the update equation described in (6).

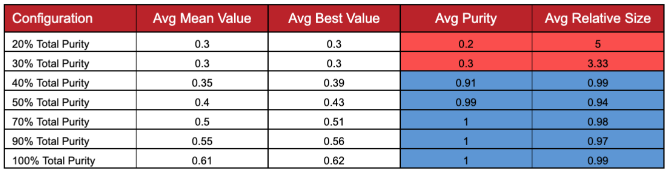
$$v_{t+1} = wv_t + U(0, c_p)(p_t - x_t) + U(0, c_g)(l_t - x_t) \tag{6}$$

Where v_t is the velocity at iteration t, x_t is the position at iteration t, p_t is the best know position so far by that specific particle, $U(0, c)$ is a uniform distribution in the range 0 and c and l_t is the best know position found in the neighborhood of that particle. The values for $w = 0.5$, $c_p = 0.5$, and $c_g = 1.0$ were defined by experimentation. An adaptive random topology with a neighborhood size of 3 is adopted, and a synchronous update with 100 particles running for 100 iterations. The position of each dimension in each particle is initialized by a sampling of a uniform distribution between 0.0 and 0.4, while the velocities are sampled uniformly between -1 and 1. The minimum number of samples is set to 10% of the simulation sizes (200 samples).

1) Perturbed PSO vs Regular PSO: The PSO and the Perturbed PSO are tested separately, and Fig. 6 displays the comparison of their outcome. The Perturbed PSO was superior in every metric while showing a smaller spread between the paths of each run. With better overall results and consistency, the Perturbed PSO is elected the preferred version, and is the one used in the following benchmark.

2) Total Purity Variation: The previous benchmark was computed in respect to only one simulation. To better understand how the optimization could take place in different scenarios, simulations with different total purity were generated, and the Perturbed PSO was again tested in each of them. Table I summarizes the results for each total purity configuration, with results for purity and relative size painted in blue if considered satisfactory and red otherwise.

TABLE I
PERTURBED PSO - TOTAL PURITY VARIATION



Configuration	Avg Mean Value	Avg Best Value	Avg Purity	Avg Relative Size
20% Total Purity	0.3	0.3	0.2	5
30% Total Purity	0.3	0.3	0.3	3.33
40% Total Purity	0.35	0.39	0.91	0.99
50% Total Purity	0.4	0.43	0.99	0.94
70% Total Purity	0.5	0.51	1	0.98
90% Total Purity	0.55	0.56	1	0.97
100% Total Purity	0.61	0.62	1	0.99

The Perturbed PSO showed to be very effective for scenarios where the predictable regime prevailed. For total purity of 40% or greater, it was able to correctly classify nearly every sample on the simulations, specializing the method in high total purity situations.

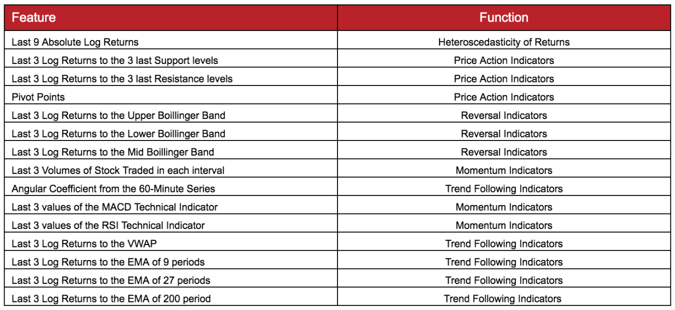
In every scenario where the algorithm failed to search the optimal subset, it diverged toward choosing the whole dataset. This is convenient because in all cases where the optimization failed to output the right answer it just outputted what would have been used if the P-Craw had not been employed. In other words, the procedure ”cannot hurt”. The rule of thumb is to use the framework and, should it output the whole dataset, just proceed without the P-Craw. In all other cases, it seems reasonable to trust the selected subset.

V. REAL CASE STUDY - BRAZILIAN STOCK MARKET INTRADAY DATA

In the real case study, the dataset used was from the Brazilian Stock Market (Bovespa). The data was obtained from the official FTP site (ftp://ftp.bmf.com.br/MarketData/) with help from the R package GetHFData [24]. The files consist of trade information from every transaction in the Limit Order Book from 2018-07-02 to 2019-03-3, compromising 8 months of data. The time series of interest is the log return of the price in the 5-minute window interval, and has approximately 9.300 observations for each stock.

The next step is to define a feature vector capable of summarizing enough information about each moment in the stock dynamics. For this purpose, a number of technical indicators were studied, and Table II compiles all the ones chosen for the construction of the feature matrix.

TABLE II
FEATURE VECTOR



Feature	Function
Last 9 Absolute Log Returns	Heteroscedasticity of Returns
Last 3 Log Returns to the 3 last Support levels	Price Action Indicators
Last 3 Log Returns to the 3 last Resistance levels	Price Action Indicators
Pivot Points	Price Action Indicators
Last 3 Log Returns to the Upper Boillinger Band	Reversal Indicators
Last 3 Log Returns to the Lower Boillinger Band	Reversal Indicators
Last 3 Log Returns to the Mid Boillinger Band	Reversal Indicators
Last 3 Volumes of Stock Traded in each interval	Momentum Indicators
Angular Coefficient from the 60-Minute Series	Trend Following Indicators
Last 3 values of the MACD Technical Indicator	Momentum Indicators
Last 3 values of the RSI Technical Indicator	Momentum Indicators
Last 3 Log Returns to the VWAP	Trend Following Indicators
Last 3 Log Returns to the EMA of 9 periods	Trend Following Indicators
Last 3 Log Returns to the EMA of 27 periods	Trend Following Indicators
Last 3 Log Returns to the EMA of 200 period	Trend Following Indicators

Unless explicitly written, those metrics were all computed in reference to the 5-minute interval windows. The target for each row is the next log return of the price series after 30 minutes in the future in respect to the last value observed.

The 20 most traded papers at the first day of the available data are chosen to be used in the study. This choice guarantees series with good amounts of daily transactions. Furthermore, the chosen stocks represent a diverse collection, with companies from sectors like Food, Oil, Education, Beverage, Commodities, Banking ... among others.

For each stock, the time period is split in a training and testing window and the PCrawl is used on the training windows to probe for predictable samples. The Boosted Trees implementation of [25] is used as a statistical model. In order to make the learning of the model fast during the optimization phase it is restricted to train only up to the first 10 trees. The parameters of the model are chosen in a grid-search in the training set, using cross validation to assert the best combination.

The final series has a total of 9.543 observations, but in order to mimic the simulation setup, only the first 2000 observations are probed for predictable periods. The Perturbed PSO is used as the optimization mechanism. Next, labels are generated from the subset selected subsets and used to train the classifier as described in the P-Craw section. The same feature representation is used, and the chosen model is the Boosted Classification Trees [25]. For this task, the parameters were chosen in order to maximize the cross-validated accuracy of the training predictions, with the same grid-search used formerly.

The outputs of the classifiers are set to be the probability of the sample being predictable, and so is bounded between 0 and 1. The predictable samples are the ones in the test sets at which the output of the classifier is above 0.99.

With the samples in the training and test windows identified for each stock, the efficiency of the PCrawl can be tested. In order to do that, a forecasting model is selected and evaluated in each stock in two different ways. First, it is trained on the whole training set, and the error metrics are computed using the whole test set. Second, the same model is trained only in the selected samples in the training set, and only the selected samples in the test set are used to compute the errors. For each series, three metrics are evaluated. Those metrics are: the Mean Absolute Error (MAE), MASE and Mean Directional Accuracy (MDA). The MDA asserts how often the model was able to correctly predict if the price would rise or fall, and is computed as in (7).

$$\begin{gathered} MDA = \frac{1}{n}\sum_{i=1}^{n} D_t \\ D_t = 1 \text{ if } \hat{y}_t * y_t > 0, 0 \text{ otherwise} \end{gathered} \tag{7}$$

In the first run, the comparison is carried using the same boosted trees models employed in the optimization phase, the results are summarized in Table III.

TABLE III
BOOSTED TREES - RESULTS FOR EACH STOCK

Stock	Selected Fraction	Whole MAE	Selected MAE	Whole DMA	Selected DMA	Whole MASE	Selected MASE
BRFS3	0.8	0.00394	0.00422	0.54	0.54	1.1	1.11
PETR4	0.68	0.00384	0.00433	0.52	0.53	1.15	1.33
BBAS3	0.2	0.00352	0.00359	0.51	0.49	1.05	1.09
BBDC4	0.75	0.00332	0.00315	0.5	0.52	1.07	1.04
ITSA4	0.26	0.00291	0.00268	0.55	0.57	1.06	1.06
VALE3	0.31	0.00344	0.00348	0.52	0.51	1.07	1.11
ITUB4	0.56	0.00285	0.00288	0.51	0.51	1.04	1.04
ABEV3	0.45	0.00317	0.00307	0.53	0.52	1.08	1.07
RENT3	0.65	0.00453	0.00432	0.52	0.5	1.12	1.09
KROT3	0.66	0.00516	0.00476	0.56	0.56	1.19	1.08
B3SA3	0.66	0.00355	0.00363	0.52	0.54	1.02	1.04
CCRO3	0.42	0.00472	0.00432	0.55	0.55	1.11	1.11
JBSS3	0.44	0.00448	0.00388	0.55	0.58	1.2	1.11
GGBR4	0.53	0.00387	0.00399	0.53	0.52	1.07	1.08
ESTC3	0.66	0.00508	0.00468	0.52	0.54	1.12	1.05
EMBR3	0.17	0.00361	0.00304	0.51	0.53	1.08	1.05
ELET3	0.58	0.00574	0.00521	0.52	0.53	1.12	1.08
CMIG4	0.3	0.00529	0.00428	0.53	0.56	1.33	1.17
ELET6	0.94	0.00533	0.00554	0.52	0.5	1.08	1.14
CSNA3	0.37	0.00456	0.00424	0.56	0.55	1.05	1.04

The MAE metric is highlighted, with the winning variant being displayed in blue for every series. As it can be seen in the Selected Fraction column of Table III, the choices of predictable samples for each stock yielded very different results. This points towards different structures being exploited for each series, instead of a common market dynamic.

To test whether or not the framework brought significant improvements a Wilcoxon signed rank paired test is performed [25]. The non-parametric test pairs the computed metrics for both cases and take the differences from the numbers obtained when using the whole dataset and the ones obtained with the selected samples. The null hypothesis is that those differences have zero mean, and an one-sided test is performed to assert if the framework brought significant improvement for each one of them. Table IV compiles the p-values for each test. Even though every metric displayed better results when the framework was employed, the only significant change confirmed is the enhancement in the MAE. Actually, if the Bonferroni correction [26] is employed, none of the tests pass the corrected confidence level (1.6 %). The correction assumes the worst case scenario of independence between the tests, which is hardly the case. The procedure can be a little too conservative in this context, but is used to ensure high confidence in the conclusions.

The scatter plot between MAEs computed using the whole dataset and the P-Craw is displayed in Fig. 7. A linear relationship can be inferred, with the metrics evaluated at the selected intervals being approximately 0.81 times the ones using the whole dataset. This signifies an reduction of almost 20% the original value, demonstrating the effect of the methodology.

TABLE IV
PAIRED TESTS P-VALUES FOR THE BOOSTED TREES MODEL

MAE	MDA	MASE
0.023	0.123	0.21

In the previous results, the same model was used for both the optimization stage and the forecasting of select samples, although with different parameters. A question that arises then is if this improvement is restrict to this model or if the selected signal indeed unveils a structure that can be exploited by other techniques. To answer that question, the signals computed are tested again with two different models. The first, the Random Forest (RF), another tree based variant. The second, a linear regression paired with the LASSO for feature selection [10], the same used in the simulation benchmark. The parameters presented in those models are computed separately for each series as in the previous case. For the RF model this

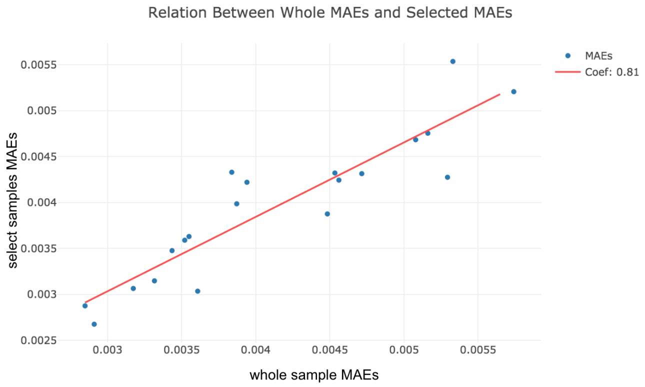


Fig. 7. Regular and Enhanced MAEs for the Boosted Trees model

TABLE V
MODELS RESULTS

Model	MAE P-value	MDA P-value	MASE P-value	MAE Enhancement
Boosted Trees	0.023	0.123	0.21	19%
Lasso	0.003	0.092	0.957	3.3%
Random Forests	0.025	0.144	0.956	3.3%

TABLE VI
PAIRED TESTS P-VALUES FOR THE AGGREGATED MODELS RESULTS

MAE	MDA	MASE
0.0001	0.022	0.237

computation is done with the aid of cross validation, and for the LASSO the Bayesian Information Criterion (BIC) is used. The comparison regarding those models is carried on exactly as it was with the Boosted Trees, and Table V contemplates the results.

To increase the power of the conclusions, the combined 60 pairs of the three models are compiled together (20 pairs for each model) and the paired tests are performed on the aggregated data. Table VI summarizes the results. The outcomes reinforce the enhancement in the MDA metric, although the final p-value for this metric don't pass the confidence level if the Bonferroni correction is employed. The aggregated results show strong evidence of reduction in the MAE metric even when conservative corrections for multiple comparisons are employed, and point evidence in the decrease of the corrected directional changes. The conclusion, overall, favor the Adaptive Markets Hypothesis, with the results shedding more light in the efficiency discussion.

VI. CONCLUSION

The present work discussed the efficiency of the market as well as the studies that inspired the proposed idea. The simulations performed showed that the P-Craw is capable of performing well in a large variety of conditions, with the Perturbed PSO consistently discovering the right patterns in scenarios ranging from 40% to 100% of predictable samples.

When applied to the brazilian stock market, the 20 most traded stocks at the beginning of the studied period were used as a benchmarks and the MAE, MDA and MASE metrics were evaluated with and without the use of the proposed framework. The P-Craw significantly increased the performance on the MAE, and displayed evidence towards an enhancement in the MDA. The effect was the greatest when the same model was used for both the probing and forecasting of samples, achieving a reduction of MAE as high as 19%.

Although the framework was not able to improve the MASE metric, the decrease in the correct directional changes showed that the selected intervals had a more predictable price dynamic. The outcome is favorable to the AMH, aggregating more evidence in favor a floating efficiency level in the market.

According to the AMH, different markets can have different efficiency levels, and so a natural direction to future works is extending the analysis to other markets to understand how the results behave.

REFERENCES

[1] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "A systematic review of fundamental and technical analysis of stock market predictions," *Artificial Intelligence Review*, pp. 1–51, 2019.

[2] S. Islam *et al.*, "Fundamental and technical analysis: Tools used in stock market," 2019.

[3] G. De Zwart, T. Markwat, L. Swinkels, and D. van Dijk, "The economic value of fundamental and technical information in emerging currency markets," *Journal of International Money and Finance*, vol. 28, no. 4, pp. 581–604, 2009.

[4] R. Efendi, N. Arbaiy, and M. M. Deris, "A new procedure in stock market forecasting based on fuzzy random auto-regression time series model," *Information Sciences*, vol. 441, pp. 113–132, 2018.

[5] S. Lahmiri, "Minute-ahead stock price forecasting based on singular spectrum analysis and support vector regression," *Applied Mathematics and Computation*, vol. 320, pp. 444–451, 2018.

[6] J. Kaur and K. Dharni, "Predicting daily returns of global stocks indices: Neural networks vs support vector machines," *Journal of Economics, Management and Trade*, pp. 1–13, 2019.

[7] S.-L. JIANG and W.-J. WU, "Forecasting stock market with social media sentiment based on adaptive network fuzzy inference system," *DEStech Transactions on Economics, Business and Management*, no. ssemr, 2019.

[8] E. F. Fama, "The behavior of stock-market prices," *The journal of Business*, vol. 38, no. 1, pp. 34–105, 1965.

[9] A. W. Lo, "The adaptive markets hypothesis," *The Journal of Portfolio Management*, vol. 30, no. 5, pp. 15–29, 2004.

[10] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.

[11] B. Qian and K. Rasheed, "Hurst exponent and financial market predictability," in *IASTED conference on Financial Engineering and Applications*, 2004, pp. 203–209.

[12] L. Molgedey and W. Ebeling, "Local order, entropy and predictability of financial time series," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 15, no. 4, pp. 733–737, 2000.

[13] E. Maasoumi and J. Racine, "Entropy and predictability of stock market returns," *Journal of Econometrics*, vol. 107, no. 1-2, pp. 291–312, 2002.

[14] M. Lettau and S. Van Nieuwerburgh, "Reconciling the return predictability evidence: The review of financial studies: Reconciling the return predictability evidence," *The Review of Financial Studies*, vol. 21, no. 4, pp. 1607–1652, 2007.

[15] S. J. Grossman and J. E. Stiglitz, "On the impossibility of informationally efficient markets," *The American economic review*, vol. 70, no. 3, pp. 393–408, 1980.

[16] G. W. Schwert, "Anomalies and market efficiency," *Handbook of the Economics of Finance*, vol. 1, pp. 939–974, 2003.

[17] C. Engel and J. Hamilton, "Long swings in the dollar: Are they in the data and do markets know it?" *American Economic Review*, vol. 80, pp. 689–713, 02 1990.

[18] S. Reitz, "On the predictive content of technical analysis," *North American Journal of Economics and Finance*, no. 17, pp. 127–137, 2006.

[19] J. H. Holland *et al.*, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[20] M. Dorigo and M. Birattari, *Ant colony optimization*. Springer, 2010.

[21] H. Shah-Hosseini, "The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm," *International Journal of Bio-inspired computation*, vol. 1, no. 1-2, pp. 71–79, 2009.

[22] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proceedings of the IEEE international conference on neural networks*, vol. 4. Citeseer, 1995, pp. 1942–1948.

[23] M. Clerc, "Beyond standard particle swarm optimisation," in *Innovations and Developments of Swarm Intelligence Applications*. IGI Global, 2012, pp. 1–19.

[24] M. Perlin and H. Ramos, *GetHFData: A R Package for Downloading and Aggregating High Frequency Trading Data from Bovespa*, 2016. [Online]. Available: https://ssrn.com/abstract=2824058

[25] R. Woolson, "Wilcoxon signed-rank test," *Wiley encyclopedia of clinical trials*, pp. 1–3, 2007.

[26] E. W. Weisstein, "Bonferroni correction," 2004.