# Data Mining Methods on Time Price Series for Algorithmic Trading Systems

Cristian PĂUNA
Economic Informatics Doctoral School
Bucharest Unversity of Economic Studies
cristian.pauna@ie.ase.ro

*Buy cheap and sell more expensive. This is the main principle to make a profit on capital markets for hundreds of years. The rule is simple but to apply it in practice has become a very difficult task nowadays, with very high price volatility in the financial markets. Once electronic trading was widespread released, reliable solutions can be found using algorithmic trading systems. This paper presents a data mining method applied to the time price series in order to generate buy and sell decisions using computational algorithms. It was found that an original data mining method based on the price cyclicality function gives us an important profit edge when it is about the capital investments on the short and medium term. The Cyclical Trading Method will be presented together with the main principles and practices to design and optimize trading software. Test results are also included in this article in order to compare the presented method with other known methodologies to trade the capital markets.*

***Keywords:*** *Data mining, Time price series, Capital markets, Cyclical trading method, Trading algorithms, Trading software*

## 1 Introduction

The main condition to make a profit on the capital markets is to buy cheap and to sell more expensive. This simple principle was explained even in the oldest book about trading. In the year of 1688 José de la Vega present this rule in the "*Confusion de Confusiones*" [1] together with other two very important aspects regarding the financial investments: the unpredictability of the stock price evolution and how important is to have patience in order to make the profit. Even from the beginning, the investors knew that they can lose money and they have to be patient in order to recover and to make a profit. After more than three hundred years nothing was changed about these simple truths of the capital investments.

Even the basic principles remain the same, the tools and methods we are using today to manage the price evolution and to build the trading decisions are different. The electronic trading systems globally implemented in all stock exchanges in the world permit us today to have fast access to the price evolution, to analyze the price behavior in real time, to build almost instantly the trading decisions and to send the trading orders using computers and low latency informatics systems.

The questions we are asking today about the investment in the capital markets are more elaborated: how to analyze the real-time price evolution in order to have a profit edge? How to decide when to enter the capital market? How much to buy and how to manage and limit the capital risk? How to build exit decisions based on price behavior? How to organize all of these into an informatics system? And how to optimize all computing processes in order to maximize the profit and to minimize the involved risk? This paper will try to answer to all these questions presenting a complete trading methodology that can be tracked and applied by anyone in order to build an algorithmic trading system.

One particular specificity of this paper is that all presented methods, algorithms and rules will be referred for short and medium time trades. This means the trades are open and kept for different periods of time, between minutes to even hundreds of hours. The purpose of this methodology is to make a profit on a longer period of time speculating some special market conditions as we will see. Starting from this fact, some readers can consider this paper dedicated rather to investment methods on the capital markets

DOI: 10.12948/issn14531305/23.1.2019.03

than to the short trading methodologies. Regardless of the used term, the presented system will make the profit from both short and medium term trades. The efficiency level obtained will prove the relevance of the presented algorithms which can be adapted and optimized for any financial market.

To answer the question when are the best entry moments and how to analyze the time price series in order to build the entry decisions, this paper will present the Cyclicality Trading Method, an original method based on the "Price Cyclicality Function" [2]. The new model will be compared with two other known methods that can be applied in the same market conditions. One was introduced by Larry Connors and Cezar Alvarez in 2009 using the 2-periods Relative Strength Index, presented as "the trades' holy grail of indicators" [3] and the second method is based on the very well-known Bollinger Bands [4].

To build the exit decisions this paper will introduce a simple and proved closing strategy based on the price behavior. The Combined Exit Method includes two profit levels and real-time price action in order to optimize trading efficiency. The presented exit methodology can also be optimized for any capital market to maximize the profit and to reduce the capital exposure level regarding the investor's requirements.

In order to build an algorithmic trading software, the entry and exit methods presented will be assembled into buyng and sell trading signals, which are Boolean variables used to automate the trading decisions. The logical structure of the trading informatics system will be presented together with the real-time data streams used to access and compute the time price series and to convert it into low-latency trading signals.

Analyzing the comparative results included in the last part of the article we can classify the Cyclicality Trading Method used with Combined Exit Method as a reliable trading methodology. Together with the suitable risk management procedure, the presented algorithms can be used in order to design a profitable trading software for any capital market.

## 2 Algorithmic trading system

An algorithmic trading system, in the acceptance of this paper, is a software implemented by the investor in direct connection with the informational brokerage system. We can call this on short as trading software. It is specially designed to receive and analyze the real-time price and the capital data streams, to build the trading signals using different data mining models, to decide about the trading volume using a capital management procedure and to build and to send the trading orders to the brokerage system.

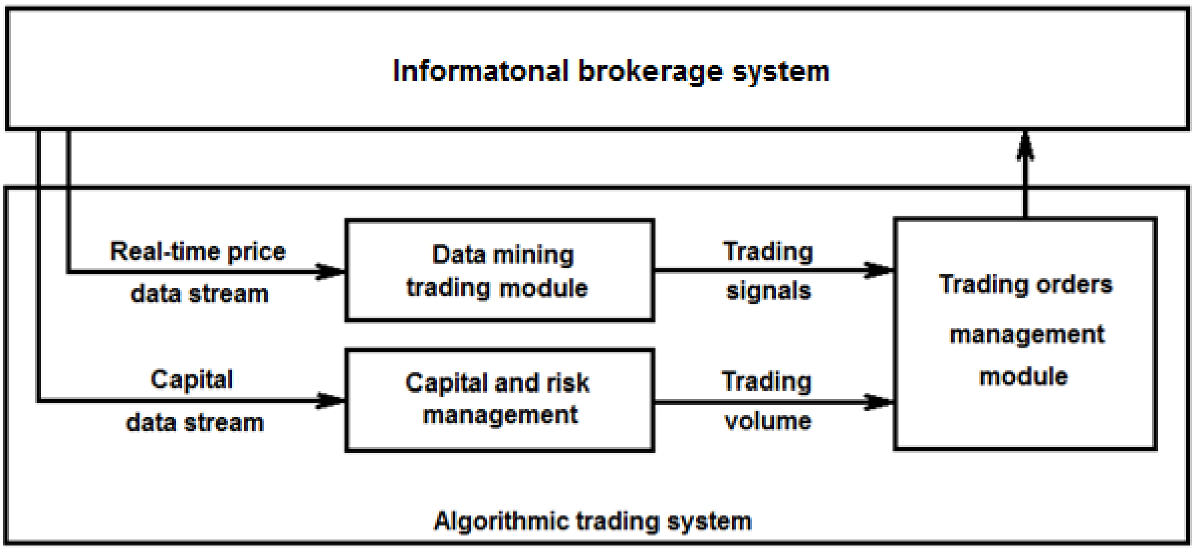

Informatonal brokerage system
Real-time price data stream
Data mining trading module
Trading signals
Trading orders management module
Capital data stream
Capital and risk management
Trading volume
Algorithmic trading system

**Fig. 1.** Data streams and logical modules of an algorithmic trading system.

DOI: 10.12948/issn14531305/23.1.2019.03

The logical modules and data streams included in an algorithmic trading system are presented in figure 1. This paper will present a mathematical model for the data mining module. The capital and risk management to decide the traded volume can be made using the “Global Stop Loss” [5] method. This methodology permits to have several simultaneous open positions made by the same trading model and to manage and optimize the capital exposure for each traded market and for a specified risk level.

One of the main requests for trading software is the low-latency response. The speed to assemble and send the trading orders is the most important factor in order to have a reliable trading software. The orders must arrive in the informational brokerage system before a significant price change, otherwise, the trading orders will be ignored. The trading orders are assembled based on the trading signals and the trading volume using a data structure provided and accepted by the brokerage system. Using fast speed data connections or different application programming interfaces, the orders are automatically sent to the brokerage systems. Usually, the accepted delay between the price data reception and the order sent is between 10 and 100 milliseconds. When the buy and sell decisions are also made automatically, without any human intervention, we will have an automated trading system.

## 3 Trading with the time price series

To trade on the capital market, in the acceptance of this paper, means to buy a specified volume of shares at a certain price in a precise moment of time and to sell the same volume of equities at another moment of time with a different price. From a mathematical point of view, a trade is defined by:

$$\left(t_{buy}, p_{buy}, V, p_{sell}, t_{sell}\right) \qquad (1)$$

where $t_{buy}$ and $p_{buy}$ are the buy time and price, $V$ is the traded volume and $t_{sell}$ and $p_{sell}$ are the sell time and price. Note that using a contract for difference, the sell operation can be made before the buy transaction. In the formula (1) the time order between the buy and the sell operations is not important.

Three decisions are involved to complete the variables in the formula (1). They are about the time and price for the buy operation, about the volume traded and about the sell time and price. As we mentioned, the decision about the trading volume can be made using the “Global Stop Loss” method [5]. This will provide the exact value for each traded volume. This paper will present hot the buy and sell decisions can be automated.

To decide about the buy and the sell variables we have to analyze the time price evolution. This is usually provided by the brokerage informational system as a repetitive stream including:

$$\left(Ask, Bid, time\right) \qquad (2)$$

where *Ask* is the current buy quote, *Bid* is the current sell quote and the *time* is the current moment of time of the stream component. The data array (2) is repeated for each moment of time; the difference between two moments of time being variable and very short, of the order of a few milliseconds. To facilitate the low-latency data processing, the history of the time price evolution is stored by the brokerage or by the trading system in data collections with a different structure:

$$\left(O, C, H, L, time, timeframe\right) \qquad (3)$$

where *O* is the open price, meaning the quote level at *time*; *C* is the close time, meaning the quote level at the moment *time+timeframe*; *H* is the highest value of the price and *L* is the lowest value of the price between *time* and *time+timeframe*. Using the structure (3) the data transfer can be made the fastest possible. In the data array given by (3) there are included a large number of information given by the stream arrays delivered by formula (2) between *time* and *time+timeframe*. Changing the *timeframe* we can have an accurate description of the price evolution for each interval. The *timeframe* can be any time interval between seconds, minutes, hours to days, weeks, month and even years. In addition, the close price value of the last

DOI: 10.12948/issn14531305/23.1.2019.03

interval is the current quoted price.

The stream data (2) is transformed and stored into the stream data (3) which will be the data input for the data mining process presented in this paper. The trading signals are functions of the data stream (3) taking Boolean values. For example, a buy signal will be described as:

$$BuySingal(time) = f(O, C, H, L, time, timeframe) = \begin{cases} true \text{ if buy condition is true} \\ false \text{ if buy condition is false} \end{cases} \tag{4}$$

For the exit decisions, an exit condition must be tested for each time moment, based on the price evolution included in the stream (3). The exit signals are assembled similarly as:

$$ExitBuySingal(time) = g(O, C, H, L, time, timeframe) = \begin{cases} true \text{ if exit condition is true} \\ false \text{ if exit condition is false} \end{cases} \tag{5}$$

Usually, the signals are noted depending on the index of the data array in the price stream (3) using indices as $BuySignal_i = BuySignal(time)$ and $ExitSignal_i = ExitBuySingal(time)$. Note that the stream (3) includes the current quote level once it is received in real-time. This will permit to build the buy and exit signals with low latency in order to have an instant decision system. For the markets where sell trades can be considered before the buy operations, the sell signals and the exit from the sell trades can be assembled in the same way. In this paper, we will develop and present only buy trades. The main objective of the presented method is to trade shares and indices on the stock markets. For this case, the long trades give us better efficiency. The data mining methods will mathematically define the conditions included in the formulas (4) and (5) in order to complete the trading methodology.

## 4 The Cyclical Trading Method

This trading method is developed starting from the three basic ideas about the capital investments mentioned at the beginning of this paper. The method purpose is to find those price levels where we can buy cheap and sell more expensive after a period of time. The patience principle will be inserted into the presented methodology by keeping the open positions for a longer period of time until an efficiency level is achieved. The unpredictability of the price behavior makes hard to evaluate the precise time interval to make a profit and to meet the efficiency level. This fact excludes the time interval to be a design requirement. Consequently, this method is not suitable for high-frequency trading systems.

The Cyclical Trading Method starts from the idea that in any market the price is increasing for a period of time. After an increased interval, the investors start to mark their profit and to close their positions. The selling orders will produce a price decreasing which determines a descending interval. After a time, when the price is lower, more and more investors start to buy again, determining accordingly a new increasing interval. The cyclic behavior of the quoted price can be observed in any capital market for any timeframe. A mathematically function describing the price cyclic behavior is the “Price Cyclicality Function” (*PCY*) introduced in [2]. Looking in figure 2. we can see the price evolution together with the *PCY* function. It can be observed the *PCY* function is ascending when the price is growing and decreasing when the price is decreasing. The minimal points of the *PCY* function are located where the price makes local minimum values.

To entry in the markets when the price is lower means to buy near the minimal point of the *PCY* function. This will generate cyclic entries using the simple trading condition made by:

$$BuySignal_i = \left(PCY_i < \xi\right) \qquad (6)$$

The $\xi$ parameter will define the maximal value of the *PCY* function when the algorithmic trading system will start to buy shares. This functional parameter can be optimized for each financial market depending on different efficiency criteria. Looking at the graph presented in figure 2. we can see that the price can decrease more after the $\xi$ value is achieved in the intervals where the *PCY* function is approaching to make a minimum value. This means that better entry points can be found after the first moment when $PCY_i < \xi$. To speculate all these better entries, the algorithmic trading software will initiate several buy trades with the lower price once the (6) condition is still valid. In order to distribute capital exposure, the additional trades will be decided by the additional trading signal:

$$BuySignal_k = \left(PCY_k < \xi\right) \wedge \left(p_k - p_{k-1} < \delta\right) \text{ with } 1 < k < M \qquad (7)$$

where *k* define each price level $p_k$ when the quotes decreased with $\delta$ distance under the last entry and *M* is the maximal number of trades.

The functional parameters $\delta$ and *M* can be optimized for each traded market using different optimization methods.

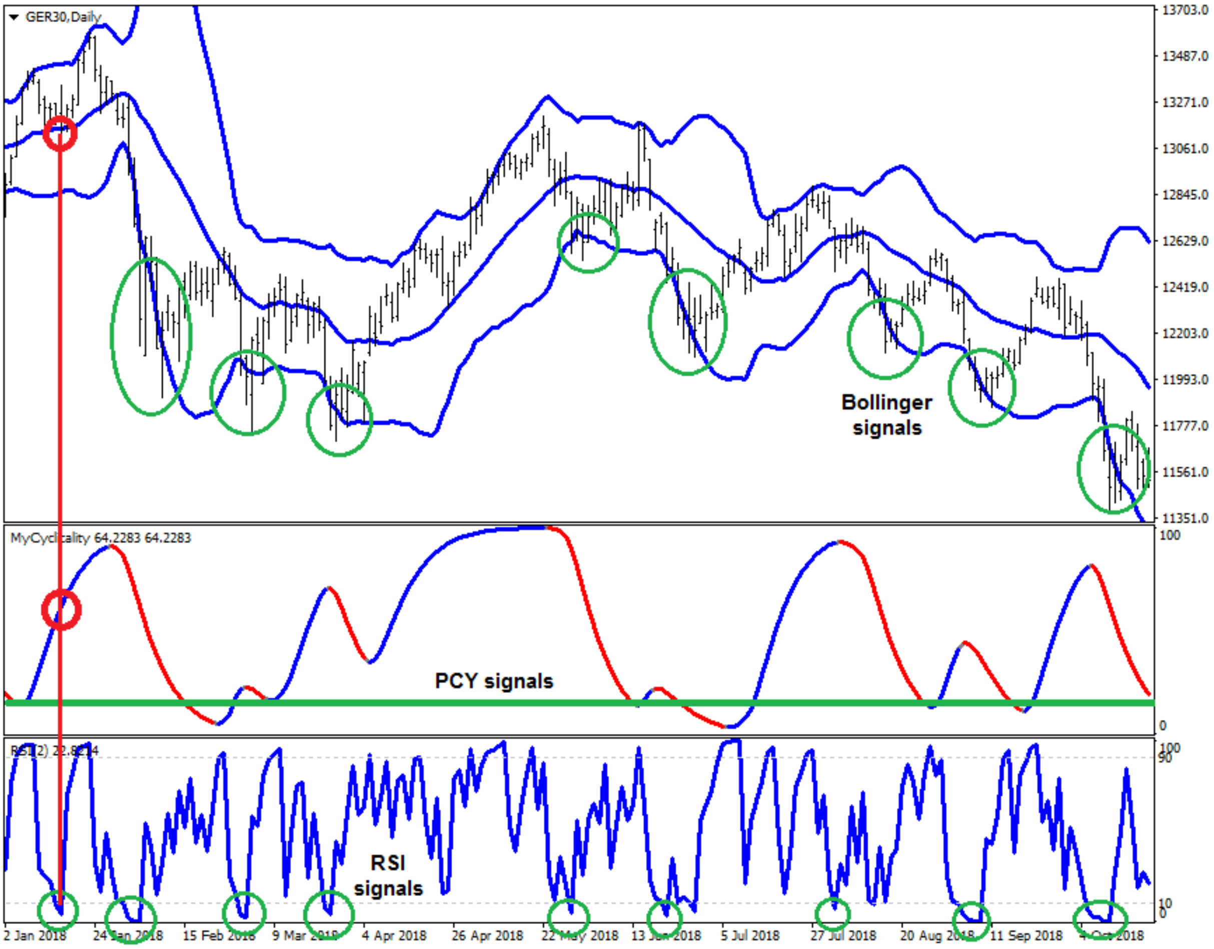


**Fig. 2.** Cyclicality price behavior. Extreme price levels detected by different methods.

The methodology implies not to trade and wait until the *PCY* function decrease under a specified level $\xi$. After this trigger is achieved, more trades will be initiated each time when the price quote decrease with one more $\delta$ distance under the last entry price. To limit the

DOI: 10.12948/issn14531305/23.1.2019.03

risk, a maximal number of open trades *M* is considered. All the trades will be kept open until a profitable condition will be met. The exit conditions will be discussed in one future chapter.

The cyclical trading method permits to initiate trades on the capital markets in those time intervals when the price makes minimal values. The advantage is the lower risk involved and the good trading efficiency obtained. The main disadvantage is the long time period when the model waits for a new trading opportunity; the numbers of the generated trades being small.

## 5 The 2-PERIOD RSI Method

The 2-period RSI [3] method uses the Relative Strength Index [6] of 2 time periods in order to build the buy decision. The buy signals are built when the RSI value decrease under a specified value:

$$BuySignal_i = (RSI_i < \alpha) \tag{8}$$

where α can take small values in order to locate those intervals when the price makes lower values. The *RSI* signals can be fallowed in figure 2. It was found that this methodology makes good trading signals but time to time it generates false signals in almost all capital markets. An example is also included in figure 2. On the left side of the graph, the *RSI* made a minimal value but the price continues to increase for another 10-12 intervals and after that significantly decreased under the entry point. In this case, the risk level involved in that trade is very high. This fact damaged the trading efficiency of the 2-period RSI method. These cases can be filtered using also the *PCY* function. Imposing a maximal value for the *PCY* function when the 2-period RSI signal is considered, we will have better results given by:

$$BuySignal_i = (RSI_i < \alpha) \wedge (PCY_i < \pi) \tag{9}$$

where *π* defines the *PCY* filter level. For most of capital markets, a common value for the π functional parameter is $\pi = 50$. In figure 2 is presented the price evolution of the Frankfurt Stock Exchange Deutscher Aktienindex DAX30 [7] for the first ten months of the 2018 year using the daily timeframe. The additional trading signals will be generated by:

$$BuySignal_k = (RSI_k < \alpha) \wedge (PCY_k < \pi) \wedge (p_k - p_{k-1} < \delta) \text{ with } 1 < k < M \tag{10}$$

where *M* is the maximal number of open trades and *δ* is the considered price step.

## 6 Lower price with Bollinger Bands

Another function that can be used in order to locate the time intervals when the price takes local minimum values is the Bollinger function [4]. In figure 2. are figured over the price graph the 2 standard deviation Bollinger bands for 20 time intervals period. Time to time the price values are below the lower Bollinger band. The trade trigger will be activated the first time when the price decreases under the lower band. The trading signal is given by:

$$BuySignal_i = (p_i < LBB_i(\sigma)) \tag{11}$$

where $p_i$ is the current ask price and $LBB_i(\sigma)$ is the lower Bollinger band calculated for σ standard deviation. The parameter σ can have any positive value and it will be the subject of an optimization process in order to increase the trading efficiency for each traded market.

In order to filter the same false trading signals explained in the previous chapter, an additional *PCY* filter will be added. In the same way, the trading signal will be generated by:

DOI: 10.12948/issn14531305/23.1.2019.03

$$BuySignal_i = \left(p_i < LBB_i(\sigma)\right) \wedge \left(PCY_i < \pi\right) \qquad (12)$$

and the repetitive trading entries will be considered using the condition:

$$BuySignal_k = \left(p_k < LBB_k(\sigma)\right) \wedge \left(PCY_k < \pi\right) \wedge \left(p_k - p_{k-1} < \delta\right) \text{ with } 1 < k < M \qquad (13)$$

where *δ* is price step used to add new trades and the *M* is the maximal number of the trades.

## 7 Combined Exit Method

Trading on the capital market involves an exit decision. The profit made by the algorithmic trading system is also the difference between the buy price and the selling price. In order to build the exit decision, several exit methodologies can be found in the literature. A comparative presentation is included in [3]. Some of the most used methods are fixed time exit strategy, fixed profit level exit strategy, first up-close exit strategy, seven up-close exit strategy, new high exit strategy, close above the moving average exit strategy and 2-period RSI exit strategy. All of these are presented and compared in [3].

This paper will present the Combined Exit Method using the 2-period RSI function in order to integrate the current price action and two profit levels for different cases. It was found that the best trading efficiency using the entries presented above is obtained with the exit condition:

$$ExitBuySignal_i = \begin{cases} Profit_i \geq N \cdot \varphi \text{ if } RSI_i < \Omega \\ Profit_i \geq N \cdot \psi \text{ if } RSI_i \geq \Omega \end{cases} \qquad (14)$$

where *N* is the number of the current open trades, *1 < N < M*, *φ* and *ψ* are two desire profit levels for the two cases, usual *φ > ψ* and *Ω* is a level of the 2-period RSI function where the market is considered overbought. In the formula (14)

$$Profit_i = \sum_{k=0}^{N} V_k \left(Ask_i - Bid_k\right) \qquad (15)$$

is the total profit in the *i* moment of time, computed as sum of the individual profit of all the *N* open trades, $V_k$ is the volume of each trade, $Bid_k$ is the entry price for each trade and $Ask_i$ is the asked price level in the current moment of time.

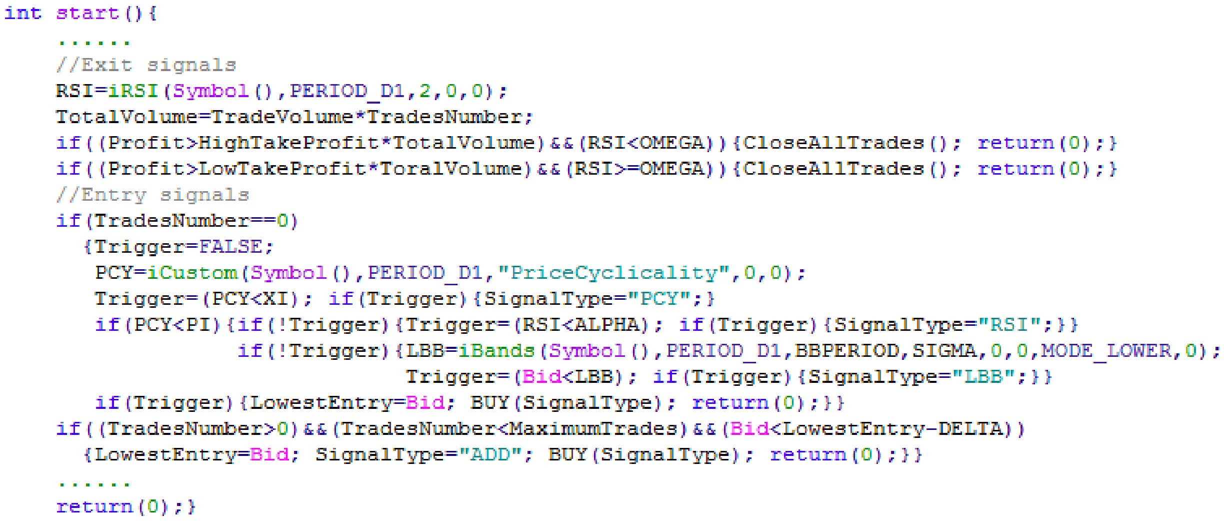
The logical description for this exit methodology derives from the principle of patience needed in the stock market in order to make a profit. Once initiated the trading process, on the first time interval until the market is not overbought $\left(RSI_i < \Omega\right)$, the algorithmic trading system waits to catch the φ profit level for each open trade. If this purpose was not achieved and the market evolved in such way to be considered overbought $\left(RSI_i \geq \Omega\right)$, a lower ψ profit level is considered and marked in order to close all opened positions before the market to turn into a descending interval. Usually the 50 < Ω < 90, depending on the traded market.

The most important advantage of the exit conditions made by formula (14) is that it can be optimized for any financial market using the functional parameters *φ, ψ* and Ω, and it can be applied for any number *N* of the open trades. The exit strategy permits to have longer intervals in order to record higher profit levels, as consequence to the patience principle included.

## 8 Code samples

Sample code for the algorithm integration is presented in figure 3. The used language is MetaQuotes programming language [8] designed to implement trading algorithms under MeraTrader platform [9].

In the presented code, the *start*() procedure is repeated at any time when the current quote price is changed. No trade is opened until the *EntryTrigger* gets the *TRUE* value which is initiated by the trading signals generated with formula (6), (9) or (12). Once a trade was opened, the *DELTA* distance from the last open price is tested in order to add new positions. Before the entry process, the current profit is computed and compared with the desired profit levels. Once a profit level is achieved, regarding the formula (14), all trades are closed and the trading software entry into the waiting phase until a new *Trigger* condition is met. As we can see in figure 3, implementing all the presented formulas can be done in simple steps.



```
int start(){
    ......
    //Exit signals
    RSI=iRSI(Symbol(),PERIOD_D1,2,0,0);
    TotalVolume=TradeVolume*TradesNumber;
    if((Profit>HighTakeProfit*TotalVolume)&&(RSI<OMEGA)){CloseAllTrades(); return(0);}
    if((Profit>LowTakeProfit*ToralVolume)&&(RSI>=OMEGA)){CloseAllTrades(); return(0);}
    //Entry signals
    if(TradesNumber==0)
      {Trigger=FALSE;
       PCY=iCustom(Symbol(),PERIOD_D1,"PriceCyclicality",0,0);
       Trigger=(PCY<XI); if(Trigger){SignalType="PCY";}
       if(PCY<PI){if(!Trigger){Trigger=(RSI<ALPHA); if(Trigger){SignalType="RSI";}}
                  if(!Trigger){LBB=iBands(Symbol(),PERIOD_D1,BBPERIOD,SIGMA,0,0,MODE_LOWER,0);
                               Trigger=(Bid<LBB); if(Trigger){SignalType="LBB";}}
       if(Trigger){LowestEntry=Bid; BUY(SignalType); return(0);}}
    if((TradesNumber>0)&&(TradesNumber<MaximumTrades)&&(Bid<LowestEntry-DELTA))
      {LowestEntry=Bid; SignalType="ADD"; BUY(SignalType); return(0);}}
    ......
    return(0);}
```

**Fig. 3.** Algorithm integration sample code in MetaQuotes programming language.

## 9 Comparative results

In this section are presented trading results obtained with the entry signals (6), (9) and (12) together with the exit signals generated by the formula (14). The results were obtained using TheDaxDealer [10], an automated trading system that uses all the presented methodologies in order to trade the Frankfurt Stock Exchange Deutscher Aktienindex DAX30 [7]. The results presented in table 1 were obtained between 01.12.2016 and 30.11.2018.

The DAX30 index was traded as a contract for differences with a spread of 1 point. The exposed capital involved and the risk management were accomplished using the "Global Stop Loss" [2] method with a maximal exposure capital level of 1%. All the data mining methods were applied for a daily timeframe. The test trading capital considered was 1.000.000 $ and an additional condition was imposed regarding the hourly intervals of the executed trades between 7:00 and 21:00 coordinated universal time (UTC) to trade on a liquid market.

**Table 1.** Trading results for DAX30 index in a time interval of 24 months.

| Entry type | Profit | Drawdown | Risk to Reward | Number of trades | Longest trade |
|---|---|---|---|---|---|
| Price Cyclicality | 30,369 | 8,583 | 1:3.54 | 249 | 424 hours |
| Relative Strength Index | 11,307 | 9,469 | 1:1.19 | 45 | 214 hours |
| Bollinger Bands | 18,293 | 8,851 | 1:2.07 | 47 | 141 hours |
| All signals together | 43,834 | 9,713 | 1:4.51 | 282 | 424 hours |

DOI: 10.12948/issn14531305/23.1.2019.03

In table 1. are presented the trading results obtained for the next values of the functional parameters:

- Price Cyclicality signals given by (6) with $\xi$=16;
- 2-period Relative Strength Index signals given by (9) with $\alpha$=4 and $\pi$=50;
- Bollinger Bands trading signals given by (12) with $\sigma$=2 and $\pi$=50;
- all signal together with the same values for the functional parameters.

The exit decisions were assembled using the formula (14) with $\Omega$=50, $\varphi$=150, $\psi$=20 and M=10. The capital evolution into the test trading account for all signals assembled together is presented in figure. 4. for 24 months interval.

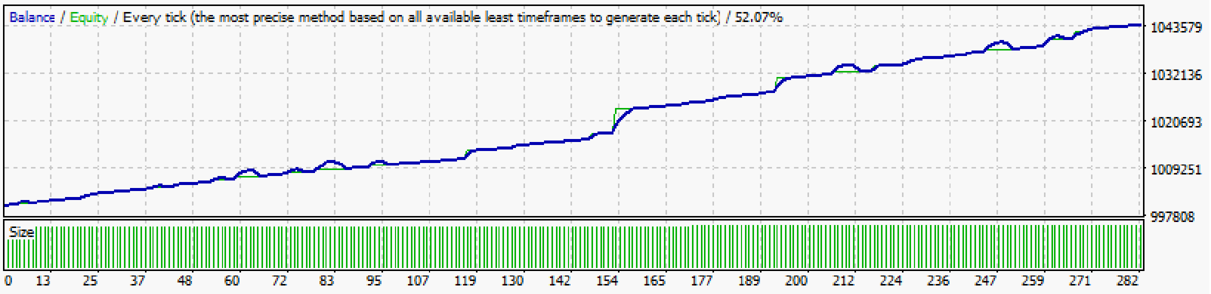


**Fig. 4.** Capital evolution for DAX30 trading index in a time interval of 24 months.

As it was already mentioned, the main disadvantage of the presented methodology is the reduced number of trades. Looking for those time intervals when the price takes minimal values, the presented trading methodology does not make any trade for long periods of time. In order to have a real image of this aspect, the no open trade time intervals were computed and presented in table 2.

**Table 2.** Trading results for DAX30 index in a time interval of 24 months.

| Entry type | Maximal no trade time | Total no trade time | Total trading time | Trading capital time efficiency |
|---|---|---|---|---|
| Price Cyclicality | 2,351 hours | 14,864 hours | 2,656 hours | 15.16% |
| Relative Strength Index | 3,289 hours | 15,629 hours | 1,891 hours | 10.79% |
| Bollinger Bands | 3,528 hours | 15,529 hours | 1,991 hours | 11.36% |
| All signals together | 2,351 hours | 14,073 hours | 3,447 hours | 19.67% |

## 10 Comments and conclusions

To compare the trading results obtained with all three entry models we have to analyze the numbers presented in table 1. The trading efficiency and the number of trades made with the Cyclical Trading Method are higher. Even the longest time trade is higher, the risk to reward ratio obtained with the cyclicality methodology is 1:3.54, a considerable value for a 24-month investment with the maximal capital exposure of only 1%. We have to observe that the longest time trade is 424 hours, between two and three weeks, a good time interval for a medium-term strategy.

Comparing the results obtained with the same capital and risk conditions, we can say that the Cyclicality Trading Method is better than the 2-period RSI method which was for many years one of the most preferable trading strategies. The entry model using the Bollinger bands makes the shortest trades. From this point of view, this is the method that places the trigger entry to the nearest point from the minimum local prices. In the test presented, the method using the Bollinger bands obtained a better reward for the same risk level as the 2-period RSI model. Played together, all three methods will produce a better profit for the same capital exposure. The total risk to reward ratio obtained with all three methodologies is 1:4.51, a good value when it is about an automated trading software set up with the maximal capital risk of 1%. Having all of these considerations and results, the question how to analyze the time price series and how to decide when is better to entry in the capital market got the answer: using the Cyclicality Trading Model. The 2-period RSI and the Bollinger bands are still valid trading models with reasonable risk to reward ratio, methods which can increase the total trading efficiency of the algorithmic trading system.

As we mentioned, risk management was made using the Global Stop Loss method. The model permits to have more open trades at the same time. This is a sustained answer to the question about how to manage the risk and the exposed capital. In addition, the Combined Exit Method offers us the answer to the question regarding exit decisions. The presented exit strategy permits to insert and implement into the mathematical algorithms the principle of patience in trading and the behavior and the current price action. All presented models can be optimized using different methodologies in order to maximize the trading efficiency and to minimize the risk level accordingly the investor's requirements. A special note is regarding the simplicity of the trading models presented and the reduced number of the functional parameters. This will permit easy adaptability of this methodology for any financial markets and will permit the use of this model in any algorithmic trading environment. In addition, the reduced number of calculations needed will contribute to the low-latency response with small resources.

The Cyclicality Trading Model together with the Combined Exit Model includes all the basic trading principles. The models buy when the capital market is near the minimum values, include the patience principle waiting on markets for higher profit opportunities and also include the unpredictability of the price behavior closing the longest trades when the market is approaching the overbought intervals.

The methodology presented in this paper was tested, implemented and used with the same good results for a representative number of financial markets: Deutscher Aktienindex (DAX30), Dow Jones Industrial Average (DJIA30), Financial Times London Stock Exchange (FTSE100), Cotation Assistée en Continue Paris (CAC40), Swiss Stock Exchange Market Index (SMI20), Australian Securities Exchange Sydney Index (ASX200), Tokyo stock Exchange Nikkei Index (Nikkei225), NASDAQ100 Index, Standard & Poor's Index (S&P500) and Small Capitalization US Index (Russell2000). Also with good and stable results, the Cyclical Trading Method presented in this article together with the Combined Exit Model was applied with good and sustained results for Gold and Bent Crude Oil markets.

Looking at the numbers in table 2. We can have some conclusions about capital usage time. From the total time allocated in the test results, the capital is involved in a trade for 15,16% of the time using the Cyclical Trading Model, higher than the 2-period RSI and Bollinger methodologies. Even so, using all the presented methods, the capital is involved in the trading activity for only 19,67 of the time. This means more than 80% of the time the capital is not used. This is the most important disadvantage of the presented methodology. To avoid and improve this aspect, the presented methods are used in practice together with other trading models to make additional trades when the capital is not used by current methods.

As final conclusions, the Cyclical Trading Method together with the Combined Exit Method gives us an important profit edge for investing in the capital markets. These methods include all the basic principles of the financial trading: the entry price is near the minimum values, the trades are kept in order to catch higher profit levels and the price action is also included to close the trades when the price behavior is changing. The low-risk level involved by entering in the markets near the minimum price intervals permits to obtain a reliable risk to reward ratio achieved in short and medium trading time. The presented methodology can be adapted for algorithmic trading systems and can be optimized for any capital markets.

## References

[1] De la Vega, J. (1688). *Confusion de Confusiones, Portions Descriptive of the Amsterdam Stock Exchange*, translated by Hermann Kellenbenz. Barcelona, Spain: Profit Editorial (2009), ISBN13: 9788496998957.

[2] Păuna, C., Lungu, I. (2018). *Price Cyclicality Model for Financial Markets. Reliable Limit Conditions for Algorithmic Trading*, Studies and Economic Calculations and Economic Cybernetics Journal, vol. 4/2018. ISSN: 0585-7511.

[3] Connors, L., Alvarez, C. (2009). *Short Term Trading Strategies That Work. A Quantified Guide to Trading Stocks and ETFs*, US: TradingMarkets Publishing Group. ISBN: 978-0-9819239-0-1.

[4] Bollinger, J. (2002). *Bollinger on Bollinger Bands. The Seminar. The essentials*. US: Bollinger Capital Management. DVD. ISBN: 978-0-9726111-0-7.

[5] Păuna, C. (2018). *Capital and Risk Management for Automated Trading Systems*. Proceedings of 17th International Conference on Informatics in Economy, Iași, Romania. Available at: https://pauna.biz/ideas

[6] Wilder, J.W. (1978). *New Concepts in Technical Trading Systems*, Greensboro, NC: Trend Research. ISBN 0-89459-027-8

[7] DAX30. (2018). *Frankfurt Stock Exchange Deutscher Aktienindex Components. Online presentation*, Wikipedia Encyclopedia. Available at: https://en.wikipedia.org/wiki/DAX

[8] MQL4. (2018). *MetaQuotes programming language online presentation*. Available at: https://www.metaquotes.net/en/metatrader4/automated_trading/mql4

[9] MT4. (2018). *MetaTrader platform online presentation*. Available at: https://www.metaquotes.net/en/metatrader4

[10] Păuna, C. (2018). *TheDaxDealer automated trading system online presentation*. Available at: https://pauna.biz/thedaxdealer

**Cristian PĂUNA** graduated the Faculty of Cybernetics, Statistics and Economic Informatics of the Economic Studies Academy in 1999 and he is also a graduate of the Aircraft Faculty from the Bucharest Polytechnic University in 1995. He got the title of Master of Science in Special Aerospace Engineering in 1996. In the last decades, he had a sustained activity in the software development industry, especially applied in the financial trading domain. Based on several original mathematical algorithms, he is the author of more automated trading software for financial markets. At present, he is Research and Development Manager of Algorithm Invest company and he is involved as a Ph.D. student in the Economic Informatics Doctoral School from the Bucharest University of Economic Studies.

DOI: 10.12948/issn14531305/23.1.2019.03