The investment strategy based on behavior of artificial earthworm for use in algotrading

Antoni Wiliński, Aneta Bera, Maciej Jarlaczyński, and Piotr Błaszyński

West Pomeranian University of Technology, Faculty of Computer Science,
Zolnierska 49, 71210 Szczecin, Poland
{awilinski,abera,mjarlaczynski,pblaszynski}@wi.zut.edu.pl
http://wi.zut.edu.pl

Abstract. The paper presents an innovative concept of investment strategy derived from the general ideas of artificial intelligence. The strategy has been tested in a number of simulations on historical data as well as data outside learning periods. Tests were performed mainly in the selected currency markets, including the primary currency pair of EURUSD. Presented approach opens long and short positions alternately during candle closing events. Closing a position is based on recommendation which is deduced from changes in characteristics of time series. The strategy has parameters that vary through time and are adapted to a market changes. To present the strategy a metaphor of an innovative artificial earthworm that feeds on a time series was used. The earthworm has its own artificial intelligence which enables it to analyse „consumed" candles, change its preferences about movement direction or to stay in neutral state. The goal was to implement presented strategy for trading platform to achieve the automatic trading effect. The strategy was tested in two development environments — MATLAB environment and trading platform MetaTrader after conversion of M-file into MQL4 file. Implemented strategy enables to achieve interesting results.

Keywords: investment strategy, machine learning, forecasting, time series, algotrading, mql4

1 Introduction

Algotrading is a new rapidly growing form of trading on financial markets where investor (that makes decisions) is replaced with trading bot (computer program) [1, 2]. Such machine can perform operations (opening and closing positions, placing orders to open positions that meet certain conditions) directly in an automated brokerage platform environment. It may also occur as a recommendation system that helps investor with making correct decisions.

Such computer applications (bots) require carefully prepared and tested software that most often needs to adapt to the changing situations on the observed markets. Modern trends, in the development of this software, include sophisticated methods of: artificial intelligence, pattern recognition, classification and

machine learning [1, 3–5]. Those methods are interpenetrating and complementary and are also expanded with information based on fundamental analysis e.g. automatically extracted from the Internet [6, 2, 7]. Many strategies are based on relatively simple rules. Efficiency of these rules is amplified by using of machine learning [3, 8–12].

Strategy presented in this paper is an example of a solution that can be used fully automatically or with assistance of investor who makes the decision manually. Using such softwaremakes sense only if there are reasonable premises of achieving the success. These conditions can be achieved through inductive thinking [13, 14], involving the assumption that multiple validation of decision on data from the past substantiates the correctness of the decision in the future. If a certain set of rules in the strategy used to seek the pattern, leads to the success then these rules are a pattern that is encouraged for repeated use. If repeated use of mentioned pattern results in more successes than failures, then the basic premise to consider the use of the pattern is rational [15, 14]. Unfortunately, changing market is causing the pattern to become outdated therefore it is necessary to constantly update it. Finding historically effective patterns that could be also effective in a data space not used for their determination (to identify these patterns) is a major challenge for machine learning which is applied in the paper. The article is organized as follows: at first authors describe an authored strategy based on an entirely innovative concept of an artificial worm. Subsequently extensive studies of effectiveness of presented strategy in terms of prediction and the aspect of investment are carried out. At the end the conclusions resulting from the study are presented and also cited bibliographic sources are included.

2   Strategy characteristics

In presented strategy, a metaphorical procedure is often used to explain heuristic algorithms of artificial intelligence. Said procedure is a reference to an inspiration derived from the nature.

Let's consider an artificial earthworm, which behaves like its natural inspirer. Let this artificial worm have a length of $p1$ candles (vectors representing the knowledge about events occurring within the contractual part of the time series). Let her movement come down to the successive „eating" one candle and simultaneous „discarding" of one candle (the last one) — always having $p1$ candles. Thus the earthworm moves incrementally at length of time called the period of the candle. $i - th$ candle is defined as a row of data of values O_i, H_i, L_i, C_i (Open, High, Low and Close). These values represent (in order): the value at the beginning of this period, the maximum value within the period, the minimum and the closing values - at the end of the candle.
Let's assume that candle can have color (often in a number of brokerage platforms, this approach is used). Therefore candles may be red (when the opening

of the candle is higher than its closing $O_i > C_i$) and green (when the situation is opposite $O_i < C_i$).
Let's also assume — it is a primary strategy rule - that when an artificial earthworm has a significant dominance of red candles, a short position is opened or continued. When earthworm has the significant dominance of green candles, a long position is opened or continued. Imprecise term „significant" is defined by the use of the second parameter $p2$ according to the rules:

$$\text{If } Lr > p1/2 - p2 \text{ then } short; \tag{1}$$

opening of short position will be suggested;

$$\text{If } Lg > p1/2 - p2 \text{ then } long; \tag{2}$$

opening of long position will be suggested;

$$\text{If } Lg < p1/2 - p2 \text{ and } Lr < p1/2 - p2 \text{ then } none; \tag{3}$$

the continuation of state will be suggested (position or no opened position).

In the rules above Lr and Lg are the number of candles, respectively „red" and „green" in the worm of $p1$ length, wherein $p1 = Lr + Lg$.
When dominance determined by another parameter $p2$ is undecided, as in (3), then worm takes the color white and the position will not be opened. When the worm continues to open position, investor does not incur costs. On many platforms, also in literature, symbolic marking of growing candles in white and decreasing in black is used. In this article, due to the occurrence of three values, green-red-white colors are used.

Earthworm is therefore supplied with artificial intellect, which allows the accumulation of knowledge and decision-making:

- right after closing of the current candle, worm „swallows" the candle and „knows" what color is it;
- earthworm knows how many candles it is composed with;
- after absorbing and discarding one candle, worm knows how many candles inside it are red or green and how to determine its status — red, green or white;
- knowing its status, earthworm knows how to make a decision about opening or continuing position; the opening of a position (long or short) means the simultaneous closing of the current position (opposite) if such position is opened;
- earthworm observes cumulative profit curve and if it notices unacceptable drawdown, then opened position is changed to the opposite;
- unacceptable drawdown is defined by two other parameters $p3$ and $p4$. $p3$ is the number of previous candles, which sets a reference point in time series, in which the value of the accumulated profit for $i - p3$ is checked for comparison

with the current value of the accumulated profit. $p4$ is permissible decrease of cumulative profit. If it is exceeded, the SL (Stop Loss — closes the currently open position) mechanism is activated;

– worm also observes current course changes during each next candle and reacts to unprofitable changes. If the current exchange rate causes a loss on the opened positions greater than $p5$ then the SL mechanism is triggered — the position is closed and till the end of the candle worm remains in the state without an open position. $p5$ is therefore the fifth strategy parameter.

These features of the worm's intelligence may be extended, however, for this work they were limited to the foregoing.

It is assumed that the time series is S_i where $i = 1, 2, ...$, and each element is a vector $S_i = (O_i, H_i, L_i, C_i)$ called candle.

Artificial earthworm executes the following algorithm.
The strategy is built on the principle of classical machine learning. Before starting to run the algorithm, on data located outside the environment that investor knows (like on the current market segments — out-of-sample), it is necessary to „teach" the values of parameters. This principle is presented in Figure 1, which shows examples of learning periods and following them testing periods. Data from testing period was not used during learning.
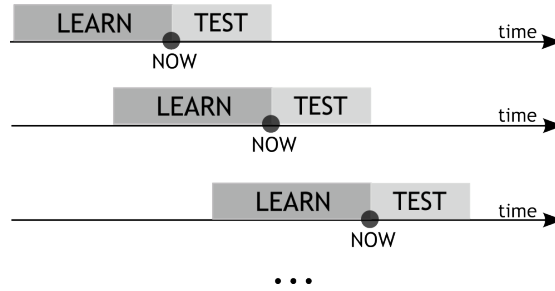

Fig. 1: An example of the arrangement of learning and testing periods of strategy parameters in time series.

This repeatable algorithm, located in both periods - learning and testing, has the following structure. In learning period the optimal value for five parameters $p1$, $p2$, ..., $p5$ was determined. In test period these designated values of parameters were used to perform the algorithm in the new environment. In each $i - th$ step (candle) of time series following activities were performed:

– For each $i = p1, p1 + 1, p1 + 2, ..., I$ and for $j = 1, ..., p1$ check whether the candle located at a distance $i - j$ backwards from the current candle is: green when $O_{i-j} < C_{i-j}$,

red when $O_{i-j} > C_{i-j}$,
where $p1$ — is the first parameter which is the length of the artificial worm — the number of candles of different colors that make up the „body" of the worm;
O_{i-j} — is the opening value of (i-j)-th candle;
C_{i-j} — is the closing value of (i-j)-th candle;
I — is the length of the test window, after which it is necessary to change the parameters values to the new, learned in the machine learning phase;

— In each $i - th$ step, the loop $j = 1, ..., p1$ is performed in which a number of green Lg, red Lr candles is calculated so the sum of Lg and Lr is $p1$ (to avoid the situation when $O_i = C_i$, it is assigned arbitrarily to one of the candles sets, for example green);
— A condition for the existence of a Rg recommendation to open a long position (or to continue it if it is open) is checked. This condition is arbitrarily adopted (one can of course consider different one):

$$\text{If } Lg > p1/2 - p2 \text{ then } Rg = 1 \tag{4}$$

If this condition is fulfilled, then depending on the current position status $PositionState$ (where $PositionState$ is a variable which takes values from $\{long, short, none\}$) at the opening of the next candle, a position is opened or continued as follows:

$$\text{If } Rg = 1 \text{ and } PositionState = long \text{ then } PositionState = long; \tag{5}$$

(costless continuation of opened position);

$$\text{If } Rg = 1 \text{ and } PositionState = short \text{ then } PositionState = long; \tag{6}$$

$$\text{If } Rg = 1 \text{ and } PositionState = none \text{ then } PositionState = long; \tag{7}$$

Similarly check the condition of recommendations Rr to open or continue short position:

$$\text{If } Lr > p1/2 - p2 \text{ then } Rr=1. \tag{8}$$

If this condition is fulfilled, then depending on what the current position status PositionState is, at the opening of the next candle, position is opened or continued as follows:

$$\text{If } Rr = 1 \text{ and } PositionState = long \text{ then } PositionState = short; \tag{9}$$

$$\text{If } Rr == 1 \text{ and } PositionState == short \text{ then } PositionState = short; \tag{10}$$

(costless continuing of opened position)

$$\text{If } Rr == 1 \text{ and } PositionState == none \text{ then } PositionState = short; \tag{11}$$

In formulas above there is a parameter $p2$. It is the second parameter of the strategy which is an arbitrarily separated part of the worm, which depending on size causes the direction of investment not to change. Large $p2$ causes long

positions to change into short positions (or vice-versa) rarely. Small value of $p2$ leads to a higher frequency of these changes. The optimum value of $p2$ is chosen similarly like $p1$.

– When opening long position, profit after one candle (this does not mean that position was closed after a single candle) can be calculated as:

$$Z_i = C_{i+1} - C_i - spread; \qquad (12)$$

where the $spread = 0$, when opened long position is continued;
and $spread = cost$, when long position was opened after closing or opening a short position in case when there was no open position;
$cost$ is a variable that depends on the particular brokerage platform that was used for testing the strategy; in this case, for the currency pair EURUSD $cost = 0.00016$ was assumed and for the currency pair GBPUSD $cost = 0.00028$.

– When short position is opened, profit after one candle (it does not mean that position is closed) can be calculated as:

$$Z_i = C_i - C_{i+1} - spread; \qquad (13)$$

where the $spread = 0$, when opened short position is continued;
and $spread = cost$, when short position was opened after closing or opening a long position in case when there was no open position.

– If as the result of the test the number of red and green candles in the body of the worm had not obtained any dominance from the color that would recommend to open defined position (worm was white), it is assumed to continue the position, which is open, without any cost of the transaction (with the value of $spread = 0$). If at the beginning of the candle, at the time of checking the recommendation, there was no open position then it can be assumed to continue without any position.

– Each of the profits calculated above is verified in terms of exceeding the allowable local position drawdown (within one candle). This additional risk management is widely used in automated trading and is called limiting losses — Stop Loss (SL). In this algorithm, SL was introduced as the fifth parameter of the strategy — $p5$.

Presented strategy was tested if:

– in the case of opening at the beginning of the $i - th$ candle or continuation of a long position presented condition did not occur:

$$\text{If } L_i - O_i < SL \text{ then } Z_i = -SL \qquad (14)$$

where L_i — the lowest rate value in the $i - th$ candle; SL — absolute value Stop Loss appearing as $p5$ parameter;

– in the case of opening at the beginning of the $i-th$ candle or continuation of a short position presented condition did not occur:

$$\text{If } O_i - H_i < SL \text{ then } Z_i = -SL \tag{15}$$

where H_i — the highest rate value in the $i-th$ candle.

This way $p5$ parameter was used for securing in strategy the strong local changes in exchange rate within a single candle.
The meaning of $p5$ parameter to the behavior of the worm and the occurrence of the closing position is explained in Figure 2.
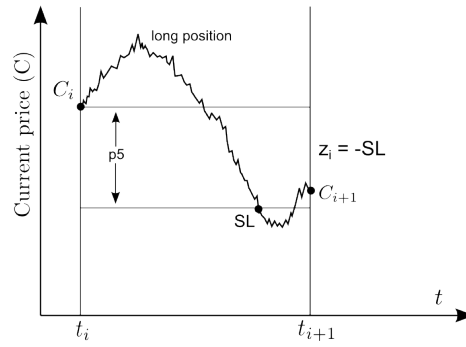

Fig. 2: Method of calculating the profit (here: loss) in the case of necessity to use $p5$ parameter actuating mechanism SL for long positions.

– To determine the current quality of the strategy and risk management summation of current gains/losses (the accumulation) is performed

$$Zc_i = sum(Z_i) \text{ from } i = 1 \text{ to the current } i \text{ value}; \tag{16}$$

– There is a possible local subsidence Zc_i capital by introducing two additional parameters to manage the risk.
$p3$ — the number of steps back on the curve of the cumulative profit for the determination of Zc_i — $p3$ for comparison with the present value of Zc_i;
$p4$ — maximum value measured in pips of the capital drawdown (eg for the currency pair EURUSD), which if exceeded will trigger the closure position:

$$\text{If } (Zc_i - p3 - Z_i) < -p4 \text{ and } PositionState \neq 0 \text{ then } PositionState = 0 \tag{17}$$

where $PositionState$ — is the type of open position (long or short) and $PositionState$ equal 0 means resignations from the opening position to the beginning of the next candle.

So, the above-described algorithm uses 5 parameters and 3 of them are devoted to risk management of the strategy.

The complexity of the algorithm is illustrated in Figure 3, which shows the individual steps and decisions made in the iteration on a single candle in the form of a decision tree. This iteration was carried out both in the search for optimal parameters $p1, ..., p5$ and in the process of testing the parameters on data that was not used in the learning period.
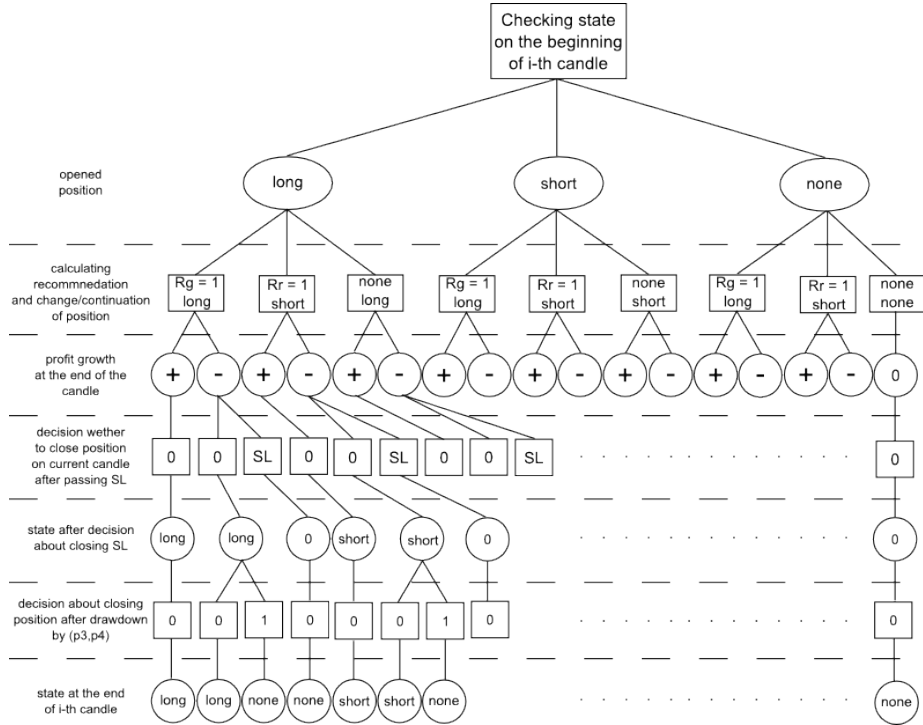

Fig. 3: Decision tree for the states and decisions in one step (candle) in the learning and testing phase.

The figure shows in a tree form further decisions taken at the beginning of the candle and then reactions to any unacceptable change in course. The first step is to examine the state of the position which is taken from the closing of the previous one. The next step is to make decisions / recommendations as to the direction of the opening from the set {long, short, none}. After opening position the return of the open position is observed — it can be positive or negative. If it is positive, no any additional decisions are necessary. In the case of a negative return from the position both the level of SL and a condition resulting from the use of parameters $(p3, p4)$ are checked. The state at the end of tree is a beginning of the next candle.

Described algorithm was implemented in MATLAB environment and in Metatrader platform using the MQL4 language. The study was conducted both in the historical data (of course with the principles of machine learning — testing was carried out using data not used in the learning phase) and a real „live" market. Between the environments, there are important differences that affect the diversity of simulation results.

3   The study of strategy

The study was conducted primarily on the basic currency pair EURUSD at 1h candles. This market was chosen because of its popularity, a huge liquidity and also due to the fact that the authors posses many experiences in other machine learning strategies for this particular market such as in [2, 3]. These experiments suggested initially, carrying out many tests with a long learning period, about 1000 candles, then 500 candles and testing time windows of 50 to 200 candles. All these attempts have proven to be ineffective and led to continuing losses. After months of tests in many markets it was noted that over long periods of learning, e.g. the order of 1000 candles, variability in parameters was very large. Therefore, the tests started with a decidedly shorter periods of learning and testing.

Finally, the authors decided to present the results for the two markets — currency pairs EURUSD and GBPUSD sampled at 1h — in machine learning with ratios adjusted to the reduced scope of both the learning and testing phase.

Table 1: The test results of the algorithm for different configurations of learning and testing periods for a currency pair EURUSD.

Learning\Test	8	10	15	20
40	-0.0261	-0.0759	-0.1424	-0.0653
60	0.0044	-0.0156	0.0275	0.1647
80	-0.0234	0.0989	**0.2401**	0.0618
100	0.0290	0.0529	-0.0104	0.0018

Table 1 summarizes the results for different time periods for the first of the currency pairs. The study was conducted at approximately constant number of candles in testing period, e.g. if the testing period was 10 candles duration, to perform the test on 1300 candles, optimal parameters were calculated 130 times. If the testing period was 20 candles long the parameters were changed 65 times, etc..

The table shows that the best results were obtained for the learning to testing period ratio of 80/15. For this ratio tests were performed on the EURUSD market and shown in the Figure 4. For the 800 one hour candles the result 0.2401

(2401 pips) is obtained. Chart is showing a growing course with a maximum fall of 600 pips, which gives a very good, according to the authors, the criterion of Calmar (ratio of maximum profit to maximum drawdown) nearly about 4.0.
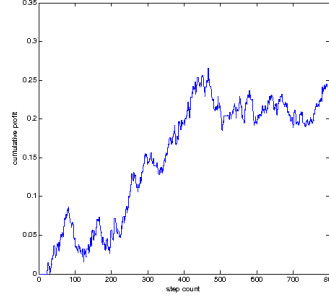

Fig. 4: Accumulation of capital for horizon of 800 steps for the currency pair EURUSD 1h. This horizon was divided into sections with constant parameter values. Test sections were preceded by learning sections with a length of 80 steps.

Tests were also made for another great important currency pair GBPUSD sampled every 1h. Results for this currency pair are worse than for EURUSD, however, they reaffirm the overall efficiency of the algorithm.
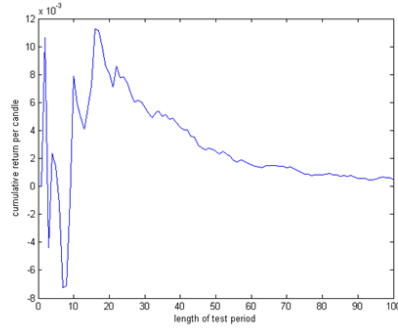

Fig. 5: EURUSD Results per candle.

The best results are obtained in both cases (for both markets) for the learning to testing period ratio of 80/15. Result for EURUSD means that the profit higher

than 100 pips per month (approximately 550 hours of trading) was achieved.

In addition, effectiveness studies were carried out and optimized for different test periods. For the purpose of comparison, the results were combined. Achieved results were divided by the length of the testing period (number of candles).

Figure 5 presents the results for EURUSD currency pair. From these results it can be concluded that the strategy is working with a short term duration characteristics. The best results were achieved in periods containing 10-20 candles — unlike in any other strategy presented in [2], where the best results were achieved for the test periods with a length of 80-100 candles.

All the results presented above refer to simulations conducted in MATLAB.

4   Strategy Verification Using Online Trading Platform

From a conceptual point of view, implementing automated trading strategies in MATLAB environment is one of proper approaches. MATLAB provides a number of features to rapidly prototype such algorithms, for e.g. built-in computation functions and charting tools. What is more, MATLAB enables also intuitive operations on vectors and matrices as well as flexible insight into variables values that are produced by the algorithm which greatly facilitates computing process management.

However, in such approach, it is very important to make sure that an algorithm, that has been developed in MATLAB, can be used also regarding to real market conditions, which means a number of rules, validations and limitations, which obviously are not natively implemented in MATLAB, as it is just an open computing environment.

To make sure that those market conditions are met in the algorithm, MATLAB script was converted manually to MQL4 expert advisor for trading platform of MetaTrader 4, which is commonly used to perform the real market operations.

The main differences between the implementation of the artificial earthworm in MATLAB and in MetaTrader are listed below:

- MATLAB
    - Calculation of balance and other standard market statistics is implemented manually, including also calculation of profit and loss related with opening / closing position;
    - SL handling is implemented manually, including trailing stop;
    - Lack of built-in market validations for opening / closing position;
    - Calculations are done using only 4 discrete price values for each candle;
    - Implementation approach from technical point of view is a matter of developer decisions;
- MetaTrader

- Balance and other standard market statistics are calculated automatically, including also a strictly defined API for opening / closing position;
- SL handling is a built-in feature, including trailing stop;
- There are built-in market validations for opening / closing position;
- Prices are changing smoothly for current candle (not only final fixed OHLC values);
- Implementation approach from technical point of view is a not only a matter of developer decisions, but is also enforced by MetaTrader conventions (phases of initialization, candle execution, deinitialization etc.);

The elements listed above could cause some variations in the fragmentary as well as the final results of the algorithm execution.

In the article, implementation for MetaTrader4 has covered only the testing part of the algorithm, using fixed set of parameters that have been previously computed in MATLAB in the learning part of execution for every test period. It is important to note that implementing full algorithm, along with the learning part in MQL4, is obviously possible, however in the article the main objective was to confirm that the artificial worm can run on a real market with results similar to those achieved in MATLAB.

Below are presented balance comparisons in MATLAB and MetaTrader environments:

1. for EURUSD currency pair using 1h candles values in time period of 2012.01.20 - 2013.01.01 in two different approaches:
   - with fixed parameters: $p1=34$, $p2=4$, $p3=3$, $p4=-0.0060$, $p5=0.0060$. Optimal parameters in that case were computed in MATLAB for the first testing period and then were used for all testing periods.
   - with dynamic parameters that were changed every testing period, regarding to their values computed in MATLAB that were put to MetaTrader
   In both approaches following assumptions were done:
   - spread was 0.00016,
   - initial deposit was 10000 EUR,
   - size of the opening positions was 1 lot,
   - each testing period lasted 15 candles.
2. for GBPUSD currency pair using 1h candles values in time period of 2013.01.09 - 2013.12.30 in two different approches:
   - with fixed parameters: $p1=27$, $p2=4$, $p3=1$, $p4=-0.0060$, $p5=0.0060$ (Figure 6a and Figure 6b). Optimal parameters in that case were computed in MATLAB for the first testing period and then were used for all testing periods.
   - with dynamic parameters that were changed every testing period, regarding to their values computed in MATLAB that were put to MetaTrader (Figure 7a and Figure 6b)
   In both approaches following assumptions were done:
   - spread was 0.00028,

- initial deposit was 50000 EUR,
- size of the opening positions was 1 lot,
- each testing period lasted 15 candles.

Results of the algorithm execution in both environments for both currency pairs are presented in Table 2.

Table 2: Results of the algorithm execution in both environments for both currency pairs and various parameters values.

	Total net profit in MetaTrader [EUR]	Total net profit in MATLAB [EUR]	Difference [EUR]	Difference [%]
GBPUSD in time period 2013.01.09 - 2013.12.30				
Fixed $p1$-$p5$	-8 823.76	-7 987.00	836.76	10.48
Dynamic $p1$-$p5$	6 057.82	5 555.00	502.82	9.05
EURUSD in time period 2012.01.20 - 2013.01.01				
Fixed $p1$-$p5$	2 305,18	2 695,00	389,82	14,46
Dynamic $p1$-$p5$	13 669,40	17 260,00	3590,60	20,80
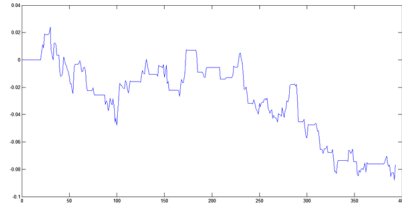
(a) Balance (relative value) in MATLAB.
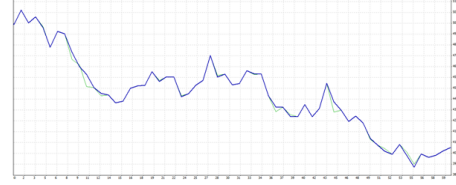
(b) Balance (absolute value) in Meta-Trader.

Fig. 6: Balance for GBPUSD [1h] in period 2013.01.09 - 2013.12.30 with fixed parameters $p1$-$p5$.

It is important to note that every first chart from each pair above was done in MATLAB and every second chart was done in MetaTrader. In MATLAB chart context, every point of abscissa axis represents testing period, while on ordinate axis there is a result of testing period (relative value), so that it can be assumed that chart was done in time domain. On the other hand, every abscissa on MetaTrader chart represents each opening of position, which means that chart is not linear related with time (however in cases presented in the article it is really similar to time domain), so that chart begins with first position opening and it is denoised in comparison to MATLAB chart as every line are determined by opening and closing position, no matter how long this position has lasted. Despite those differences, correlation between MATLAB and MetaTrader charts for the artificial worm algorithm executions is indisputable.
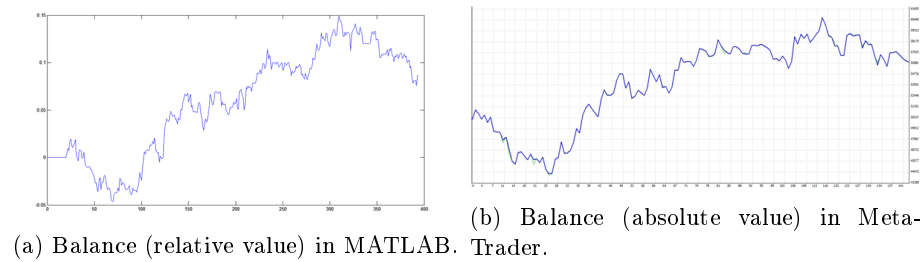
(a) Balance (relative value) in MATLAB.

(b) Balance (absolute value) in Meta-Trader.

Fig. 7: Balance for GBPUSD [1h] in period 2013.01.09 - 2013.12.30 with dynamic parameters $p1$-$p5$.

5   Conclusions

Presented in the paper the concept of an innovative investment strategy gave satisfactory results for a certain pair as the result of periods of machine learning. For example, in the studied data for two key market currency pairs sampled every hour, the best value is 80 hours of teaching and 15 hours of testing. One can assume that this relationship may change in a different environment (for other markets, for other periods) by becoming a sort of next parameter that optimal values should be sought. It should be emphasized that the study was conducted on data not used in the learning phase and for the basic principle of machine learning achieved results that would have satisfied many a practice. It should also be noted that outside research conducted in environment completely specified (for constant unchanging data stored in MATLAB) was also performed simulations in terms of actual brokerage platform. The results obtained in both environments are similar in the sense of direction changes and logic behavior caused by the user received by the algorithm implemented in the MATLAB and MetaTrader. Differences were explained in this work. The direction of further research will test the hypothesis that the variability periods to test the learner can improve the quality of prediction. This hypothesis seems to be interesting against the background of previous experiences and results, eg for different studied strategies, including [2][3][5][10] obtained surprisingly different optimal length of periods of learning and testing.

References

1. Leshik, E., Cralle, J.: An Introduction to Algorithmic Trading: Basic to Advanced Strategies (Google eBook). John Wiley & Sons (2011)
2. Wiliński, A., Bera, A., Nowicki, W., Błaszyński, P.: Study on the Effectiveness of the Investment Strategy Based on a Classifier with Rules Adapted by Machine Learning. ISRN Artificial Intelligence **2014** (2014) 1–10
3. Wiliński, A.: Prediction Models of Financial Markets Based on Multiregression Algorithms. The Computer Science Journal of Moldova **19**(2) (2011) 178–188
4. Wang, F., Dong, K., Deng, X.: Algorithmic trading system: design and applications. Frontiers of Computer Science in China **3**(2) (2009) 235–246

5. Sinclair, E.: Trade Evaluation. Wiley Trading. In: Volatility Trading. Wiley (2011)
6. Elder, A.: Come Into My Trading Room: A Complete Guide to Trading. Wiley Trading Advantage Series. Wiley (2002)
7. Schwager, J.D.: Technical Analysis. John Wiley & Sons (1996)
8. Person, J.: A Complete Guide to Technical Trading Tactics: How to Profit Using Pivot Points, Candlesticks & Other Indicators. Wiley Trading. Wiley (2004)
9. Tian, X., Quan, C., Zhang, J., Cai, H.: Optimization of intraday trading strategy based on acd rules and pivot point system in chinese market. Journal of Intelligent Learning Systems & Applications **4**(4) (2012) 279–284
10. Krutsinger, J.: Trading Systems: Secrets of the Masters. McGraw-Hill Professional Publishing (1997)
11. Lewis, H.G., Brown, M.: A generalized confusion matrix for assessing area estimates from remotely sensed data. International Journal of Remote Sensing **22**(16) (2001) 3223–3235
12. Murphy, J.: Intermarket Technical Analysis: Trading Strategies for the Global Stock, Bond, Commodity, and Currency Markets. Volume 6 of Wiley Finance. John Wiley & Sons (1991)
13. Provost, F., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: In Proceedings of the Fifteenth International Conference on Machine Learning, Morgan Kaufmann (1997) 445–453
14. Klęsk, P., Wiliński, A.: Market Trajectory Recognition and Trajectory Prediction Using Markov Models. In: ICAISC (1). (2010) 405–413
15. Bishop, C.M.: Pattern recognition and machine learning. Volume 1. springer (2006)