

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/320012598>

Deep Learning for Financial Time Series Forecasting in A-Trader System

Conference Paper · September 2017

DOI: 10.15439/2017F449

CITATIONS

26

READS

1,211

2 authors:



[Jerzy Korczak](#)

Wroclaw University of Economics and Business

131 PUBLICATIONS 791 CITATIONS

[SEE PROFILE](#)



[Marcin Hernes](#)

Wroclaw University of Economics and Business

71 PUBLICATIONS 222 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Real Time Omnichannel Marketing [View project](#)

Deep Learning for Financial Time Series Forecasting in A-Trader System

Jerzy Korczak, Marcin Hernes

Wrocław University of Economics, ul. Komandorska 118/120, 53-345 Wrocław, Poland

e-mail: {jerzy.korczak, marcin.hernes} at ue.wroc.pl

Abstract—The paper presents aspects related to developing methods for financial time series forecasting using deep learning in relation to multi-agent stock trading system, called A-Trader. On the basis of this model, an investment strategies in A-Trader system can be build. The first part of the paper briefly discusses a problem of financial time series on FOREX market. Classical neural networks and deep learning models are outlined, their performances are analyzed. The final part presents deployment and evaluation of a deep learning model implemented using H2O library as an agent of A-Trader system.

I. INTRODUCTION

OWING to fluctuations of return rates and risk accompanying investment decisions, forecasting financial time series constitutes a very difficult problem. Currently, most trading systems are based on one or a limited number of algorithms. The proposed prediction models use, for example, genetic algorithms [1], fundamental and technical analysis [2,3,4,5], neural networks and neuro-fuzzy computing [6], behavioral techniques [7]. There are also many multi-agent approach based solutions [8,9,10,11,12]. The trend is that in most cases multiple software agents that use different methods and techniques help provide trading advice.

The problem remains unresolved. Complexity of trading problems requires the use of increasingly more sophisticated models oriented towards non-linearity of financial time series. One of these models is based on deep learning [13,14].

According to [15], deep learning algorithms have already demonstrated their ability:

- to learn complex, highly varying functions, with a number of variations much exceeding the number of training examples,
- to learn with little human input low-level, intermediate and high level abstractions of input data,
- for being computationally scalable, with linear complexity,
- to learn from predominantly unlabeled data and work in a semi-supervised setting, where examples might have incorrect labels.

In stock trading, L. Takeuchi et al. [16] carried out a study similar to the one presented in our paper using an auto-encoder and various types of networks on data extracted from S&P 500, Nasdaq Composite and AMEX lists. G. Hinton et al. [17] have used deep learning methods to build deep belief networks (DBN) based stock decision support system, with training and test data from S&P500. They found out that their system outperforms the buy-and-hold strategy. Other approaches to deep learning using regression models on chaotic time series, showing good results, are presented in [18].

Our platform, called A-Trader [19,20,21], allows for implementation of various algorithms or trading decision support methods. Recently, A-Trader is aimed at supporting trading decisions on FOREX market (Foreign Exchange Market). Currencies are traded in pairs, for example USD/PLN, EUR/GBP on FOREX. In general, a trader on FOREX can open/close long/short positions. A long position relies on "*buying low and selling high*" in order to achieve a profit. On the other hand, a short position relies on "*buying high and selling low*". On FOREX, as one currency in a pair rises in value, other drops, and vice versa [19]. A-Trader mainly supports High Frequency Trading (HFT) [20], putting strong emphasis on price formation processes, short-term positions, fast computing, and efficient and robust indicators [21].

An agents' knowledge is represented by three-valued logic (where 1 denotes open long/ close short position, -1 denotes open short/close long position and 0 denotes "*do nothing*") in A-Trader system. Also fuzzy logic is used for agents' knowledge representation, where confidence of decisions is in the range [-1..1], with "-1" level denoting "strong sell" decision, "0" level denoting "*strong leave unchanged*" decision and "1" level denoting "*strong buy*" decision. The positions can be open/close with different levels of confidence of decision. For example, long position can be open, when the level of confidence is 0.6 or short position can be open, when the level of probability is 0.7. Therefore, the timeframe for the opening/closing position is wider than in the case of three-valued logic.

A-Trader architecture and description of different groups of agents have already been detailed [19]. In order to support trading decisions, agents apply technical,

fundamental and behavioral analysis as well as different methods for knowledge integration (e.g. consensus [22,23]).

The aim of this paper is to present an application of deep learning methods for financial time series forecasting in A-Trader system environment.

The first part of the paper briefly outlines a problem of financial time series on FOREX market. Classical neural networks and deep learning models are analyzed in the next part of the paper. The final part presents deployment and performance evaluation of implemented deep learning model compared with trading strategies already implemented in the system.

II. FINANCIAL TIME SERIES ON FOREX MARKET

Trading on FOREX relies on forecasting of opening or closing long or short positions. A long position is a situation in which one purchases a currency pair at a certain price hoping to sell it later at a higher price. This is also referred to as the notion of "*buy low, sell high*" in other trading markets. If a trader expects a currency pair to drop, he will sell it hoping to buy it back later at a lower price. This is understood as a short position, which is the opposite of a long position. A-Trader receives tick data grouped into minute aggregates (M1, M5, M15, M30), hourly aggregates (H1, H4), daily aggregates (D1), weekly aggregates (W1) and monthly aggregates (MN1).

High frequency traders are constantly taking advantage of very small fluctuations in quotation with a high rate of recurrence to arrive at significant profit margins. As many HFT experts have pointed out, traders seek profits from the market's liquidity imbalances and short-term pricing inefficiencies. Hence, minimization of time from the access to quote information, through the entry of an order right to its execution, is vital. On the whole, to be of any help to traders, systems must as quickly as possible provide advice as to the best move to be made: buy, sell or do nothing.

Fig 1 presents an example of visualization of open/close long/short positions. The green marks denote a long position, red ones denote a short position. The red dots denote open short/close long position, green ones denote open long/close short position. There are positions opened too early (example marked by yellow oval) or too late (example marked by brown oval). There also losses-generating positions (example marked by blue rectangle). Therefore, there is still a strong need to look for more efficient prediction methods or models, such as neural nets, that would generate more profitable decisions..

III. CLASSICAL NEURAL NETWORK MODELS

Classical neural networks will be examined in order to compare and demonstrate prediction quality of deep learning model. Artificial neural networks with backpropagation learning algorithm have been widely used in solving various prediction problems and have already shown great potential for discovering non-linear relationships in time-series [e.g. 24,25,26]. Until recently, the depth of a practical neural network was mostly limited to one or two hidden layers.

In our previous paper, Multi-Layer Perceptron (MLP) model was employed to build agents of fundamental analysis. A diagram of agent operation is schematically shown in Fig 2. MLP uses sigmoid activation function and back-propagation learning algorithm. Input vector contained long term indicators and the last sequences of S&P500, FTSE 100, oil and gold tics. Long and short-term fundamental indicators were taken into account. Prediction of GBP/PLN pair return rates, shifted by T_n units in time (in our case study M5 period and USD/GBP), constituted the output. Prediction is performed as $G(t_n+t_0)$.

To remind the definition of an agent, below the trading agent based on the fundamental analysis is specified:

```
Input:  $q_{FTSE100} = \langle q_{ftse_1}, q_{ftse_2}, \dots, q_{ftse_n} \rangle$ 
// FTSE100 quotations,
```



Fig. 1. Example of long/short position visualization in A-Trader.
Source: Own work.

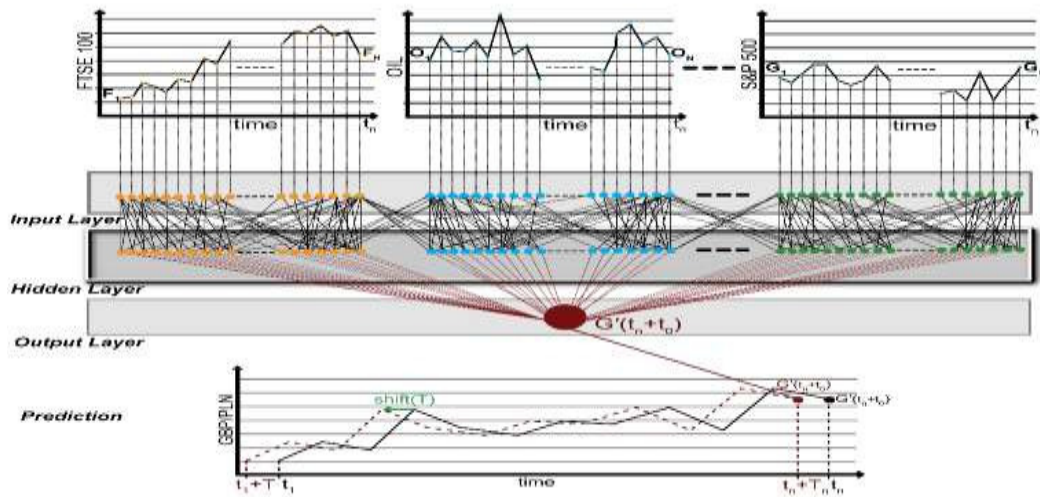


Fig. 2. MLP agent
Source: [21].

```

q_S&P500=<q_s&p1, q_s&p2, .... q_s&pM>
// S&P500 quotations
q_GOLD=<q_gold1, q_gold2, .... q_goldM>
// GOLD quotations
q_OIL=<q_oil1, q_oil2, .... q_oilM>
// OIL quotations
i_Inflation // M/M inflation change
i_NCI // Net Capital Inflows change
i_COE // Change of employment M/M
i_IP // Industrial Production M/M change
thresholdopen //threshold level for open
//long/close short //position
thresholdclose //threshold level for close
//long/open short //position
Output: Fuzzy logic recommendation D
// (range [-1..1]).
BEGIN
  if
    CheckPerformanceLevel()
    //If financial indicators are not acceptable
    //see Tab.1
  then
    BeginLearningProcess();
    p_USD/GBPM+3 := Multi-layerPerceptron
    (q_FTSE100, q_S&P500, q_GOLD,
    q_OIL, i_Inflation, i_NCI, i_COE, i_IP);
    //Prediction of USD/GBP value change
    if
      p_USD/GBPM+3 > thresholdopen
    then
      D:= heuristic_open(p_USD/GBPM+3);
    else if
      p_USD/GBPM+3 < thresholdclose
    then
      D:= heuristic_close(p_USD/GBPM+3);
    otherwise D:= heuristic_do_nothing
END

```

From a financial point of view, fundamental analysis agent is founded on money flow interpretation. For instance, if S&P 500 is falling and FTSE 100 is rising, one can expect investors to swap their S&P shares for USD, then swap USD

for GBP, and finally buy FTSE 100 shares. Or if they bought GBP for USD, the value of GBP to USD would rise.

While performing experiments [21] using MLP network it was found out that the values of input vectors of most of fundamental analysis ratios were relatively stable. The result was that in relation to high short-time fluctuation of currencies learning process was not always convergent and the outcomes were financially very weak. These were the main reasons to look for more relevant data and more complex forecasting models.

IV. DESIGN OF DEEP LEARNING MODELS

Deep Learning is a set of machine learning algorithms that attempt to model high-level abstractions in data by using architectures composed of multiple non-linear transformations [27, 28]. Incidentally, MLPs with just one hidden layer are not deep because they have no feature hierarchy and they operate on original input space. Various deep learning architecture, like deep neural networks, convolutional deep neural networks, deep belief networks, recurrent neural networks are used in financial forecasting.

The network architectures used in this research are based on Convolutional Neural Networks (CNN). CNNs are variations of MLP designed to use minimal amounts of preprocessing [29]. CNNs constitute a type of feed-forward artificial neural networks in which connectivity pattern between its neurons mimics animal visual cortex connection structure. Individual cortical neurons respond to stimuli in a restricted region of space known as receptive field. The receptive fields of different neurons partially overlap such that they tile the visual field. Response of an individual neuron to stimuli within its receptive field can be mathematically approximated by a convolution operation.

Standard MLP and Deep Learning Model differ both in architecture and in the training procedure. Prior to deep learning, MLPs were typically initialized using random numbers. MLPs currently in use apply the gradient of the network's parameters related to the network's error in order to affine parameters so as to improve prediction in each

training iteration. In back propagation, it is necessary to multiply each layer's parameters and gradients together across all the layers in order to evaluate this gradient. This involves intensive computation, especially for networks with more than two layers. Often, the gradient converges to machine-zero value and training stops or explodes into a huge value; ultimately training process becomes intractable [30,31].

Deep learning offers a new learning algorithm: to find the initial parameters for deep CNN, it uses a series of single layer networks - which do not suffer from vanishing or exploding gradients. The idea of DL is illustrated in the example shown on fig. 3 where inputs are marked violet, hidden layers are marked green and outputs are marked blue. This process makes it possible to create progressively initial features of CNN input data. It can be performed in the following way [32]:

1. An auto-encoder (a simple 3-layers neural network where output units are directly connected to the neurons of the next layer – top right column on Fig. 3) is used to find initial parameters for the first layer of a deep CNN (left column on Fig 3).
2. In the same way, a single layer auto-encoder network is used to find initial parameters for the second layer of a deep CNN, and likewise for the next layers.
3. Finally, a softmax classifier (logistic regression) is used to find initial parameters for the output layer of a deep CNN.

Deep Learning is able to efficiently discover and extract new features that can be tuned with more data but there are not easy to interpret. Deep learning models are highly flexible and configurable, however, its theory is still not well understood. For example, the problem to define a number of neurons and number of layers essentially remains unresolved; nevertheless, it is a known fact that more layers means non-linearity, while more neurons translates to more features.

V. DEPLOYMENT AND EVALUATION OF DL MODEL

Implementation is done in H₂O. H₂O is an open-source distributed in-memory data analysis and modeling platform [33]. It is interactive, scalable, extensible solution consist of several machine learning models (including Deep Learning Model), oriented to exploration of Big Data. The advantage of H₂O is that it integrates all programming environment and analytical platforms, e.g. Java, Python, JSON, R, Scala commonly in use [34]. In our project H₂O has been integrated with A-Trader. Detailed architecture of this system has been presented in [19,20,21]. In this paper, we have outlined only those components closely related to Deep Learning Model (Fig 4).

Notification Agent of A-Trader receives quotations, distributes messages and data to various agents, and controls system operation. The main objective of Supervisor Agent is to generate profit-generating, investment risk-minimizing trading advice. A Supervisor, by using different trading strategies, coordinates computing on the basis of decisions

generated by other agents, and gives the trader the final decision. Conflicts are resolved and integration of agents' pieces of advice is performed by the Supervisor, after factoring in the decisions of all the other agents, and evaluating their performance (on the basis of performance ratios described in the further part of this paper). The Supervisor determines the agents of whom the advice is to be taken into consideration when making an investment decision and those, the advice of whom is to be ignored. The DeepLearningH2O Agent runs in two modes (controlled by Supervisor):

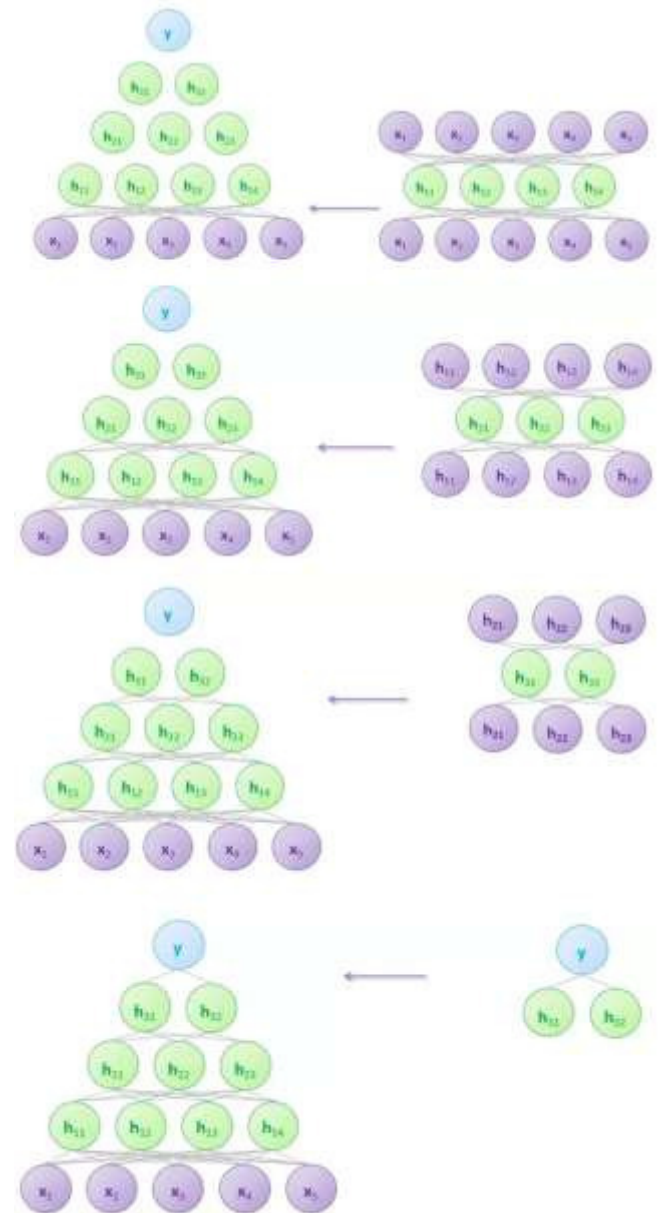


Fig.3. Deep learning process
Source: [32].

1. Learning mode (continuous dots) – in this mode following steps are performed:

- Import time series to H₂O platform (because H₂O constitutes A-Trader's external module, data are imported indirectly from A-Trader database, it does not use the data notified by Notification Agent),
 - Specification of Deep Learning (DL) model,
 - Parametrization of DL model – (such parameters as number of hidden layers, number of training epochs, stopping rounds, stopping metrics, etc),
 - Model building – on the basis of structure of imported data and determined parameters,
 - Learning-Testing –built model is trained on the basis of training time series and testing on the basis of validation dataset.
2. Forecasting mode (dashed dot) – the time series are continuously imported from A-Trader database and trained model is used for forecasting future rates of returns.

Other agents of A-Trader support the process, among other:

- Basic Agents - pre-process time series and compute the basic indicators; agents with own knowledge base can learn and change their parameters as well as their internal states,
- Intelligent Agents include all the agents based on artificial intelligence models (e.g. genetic algorithms, neural networks – including MLP, rule-based systems, etc.), agents analyzing text messages, agents observing market behavior. A decision transferred to the Supervisor Agent

constitutes the output of Basic Agents and Intelligent Agents.

Next part of the paper details the Deep Learning Model used in DeepLearningH2O Agent.

Formally, the model can be defined as follows:

$$Y_{t+1}^i = DLM(x^i, x^p \dots x^q), \quad (1)$$

where:

x^i is an input vector of rates of return related to main quotation.

x^p, \dots, x^q are inputs containing rates of return related to quotations correlated with main quotation (e.g., main quotation is USD/GBP and related quotations are oil quotes and gold quotes).

Long-return rates- are used in this model, they are calculated as follows:

$$r^i(t) = \log\left(\frac{S_t^i}{S_{t-1}^i}\right) \quad (2)$$

where S_t^i is a price of quotation i at time t .

Long-returns rates are normalized and projected by H₂O in the range from -1 to 1.

Formula of input vector related to main quotation:

$$x^i = \{r^i(t), r^i(t-1), \dots, r^i(t-k)\} \quad (3)$$

where k is a number of past quotations used as inputs – in this experiment we assume $k=30$.

Y_{t+1} assumes a value in the range $[-1, 1]$ (generated as fuzzy logic output) and it predicts a log-rates-of-return at time $t+1$ (normalized value).

Training set consists of input vectors x^i and inputs

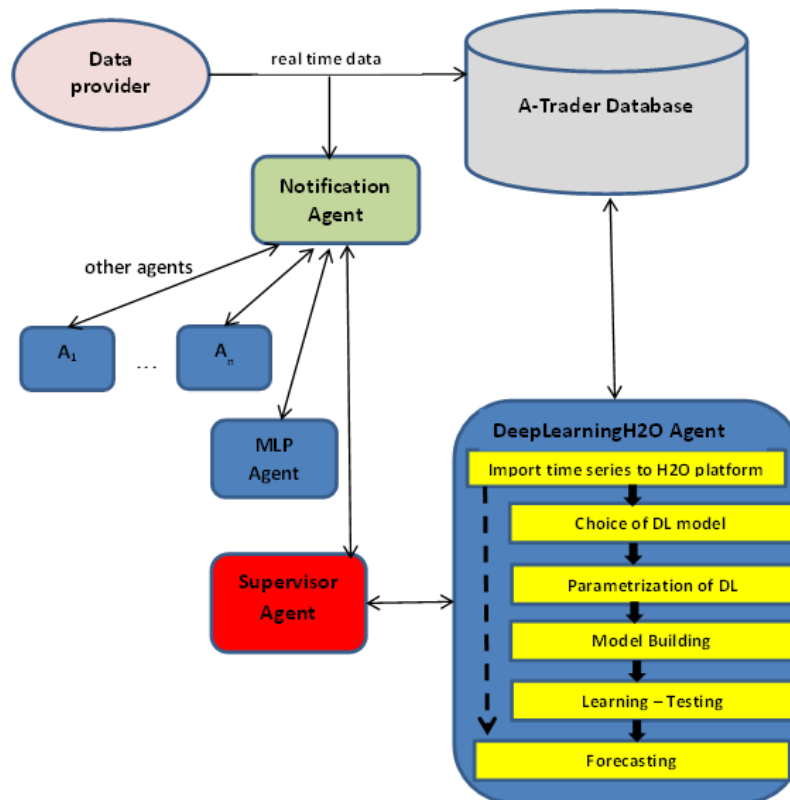


Fig. 4 Schema of processes.
Source: Own work

$x^p \dots x^q$ at time t , $t - 1$, etc. (in this experiment have been used about 1400 inputs quotations) and output at time $t + 1$ (the learning process is performed on the basis of historical time series, hence log-rate-of-return at time $t + 1$ is known).

On the basis of DeepLearningH2O Agent's output, the Supervisor Agent is able to use different strategies to generate opening/closing positions, for example by using genetic algorithm or consensus strategy. Supervisor takes into consideration thresholds for open/close positions determined by genetic algorithm. Supervisor also determines the mode of DeepLearningH2O Agent operation. Learning mode is started if performance (performance ratios are presented in further part of this paper) of this agent is low. If performance is high, then only forecasting mode is run using previously generated model.

The agent's performance analysis was carried out for data within the M5 period of quotations from FOREX market. For the purpose of this analysis, test was performed in which the following assumptions were made:

1. USD/GBP quotes were selected from randomly chosen periods (each - 1440 quotations), notably:
 - 15-03-2016, 0:00 am to 15-03-2016, 23:59 pm,
 - 16-03-2016, 0:00 am to 16-03-2016, 23:59 pm,
 - 22-03-2016, 0:00 am to 22-03-2016, 23:59 pm.
2. At the verification, the trading signals (for open long/close short position equals 1, close long/open short position equals -1) were generated by the Supervisor on the basis of DeepLearningH2O Agent and MLP agent.
3. It was assumed that decisions' probability levels for open/close position are determined by the genetic algorithm (on the basis of earlier periods).
4. It was assumed that pips (a pip is equivalent to the final number in a currency pair's price) constituted the unit of performance of analysis ratios (absolute ratios).
5. The transaction costs were directly proportional to the number of transactions.
6. The capital management - it was assumed that in each transaction the investor commits 100% of the capital held

at the leverage 1:1. The investor may define another capital management strategy.

7. The results obtained by tested agents were compared with the results of *Buy-and-Hold* benchmark (a trader buys a currency at the beginning and sells a currency at the end of a given period).

The DeepLearningH2O Agent was run in stand-alone computer with 4 processors Intel Xeon E52640 v3 2.6GHz 8C/16T, and 32GB memory with the following parameters:

- input: 30 return rates USD/GBP, oil and gold quotes, (sliding window),
- hidden layers [100,100,100,100] trained for 200 epochs,
- output 1 predicted,
- overwrite_with_best_model = true,
- stopping_rounds = 5,
- stopping_metrics = "MSE",
- stopping_tolerance = 5e-2,
- score_validation_sampling = "Stratified",
- score_duty_cycle=0.1,
- to reduce overfitting we have specified RectifierWithDropout, and the default values for Hidden_dropout_ratio = 0.15.

The final deep learning contained 4 hidden layers of 78, 64, 87, and 63 neurons respectively. The learning time of model is about 1 minute. The computing time of forecasting was approximately 0.14 sec.

Table 1 presents the results obtained in given particular periods. In general, it should be noted that not all decisions generated by agents were profitable. However, in A-Trader, performance evaluation does not compute rate of return alone, there are other ratios of significance, among them risk involved in trading (these measures have been described in detail in [20]).

The evaluation function provides the fast choice of the best agent. It may be noted that the values of efficiency

TABLE I.
PERFORMANCE ANALYSIS RESULTS

Ratio	DeepLearningH2O			MLP			B & H		
	Period 1	Period 2	Period 3	Period 1	Period 2	Period 3	Period 1	Period 2	Period 3
Rate of return [pips]	27	127	24	9	-66	28	-143	89	-160
The number of transactions	11	14	9	4	7	5	1	1	1
Gross profit [pips]	21	110	43	11	61	43	0	89	0
Gross loss [pips]	14	63	38	17	37	28	-143	0	-160
The number of profitable transactions	7	9	6	3	3	3	0	1	0
The number of profitable consecutive transactions	3	3	2	2	2	2	0	1	0
The number of unprofitable consecutive transactions	2	2	1	1	3	2	1	0	1
Sharpe ratio	0.83	1.64	1.17	0.55	1.8	0.78	0	0	0
The average coefficient of volatility [%]	1.12	0.74	1.92	1.66	0.86	1.94	0	0	0
The average rate of return per transaction	2.45	9.07	2.67	2.25	-9,43	5,6	-143	89	-160
Value of evaluation function (y)	0.45	0.51	0.42	0.38	0.17	0.53	0.06	0.32	0,03

ratios of particular agents differ in each period. Values of this function oscillate in the range [0.03, 0.53]. Therefore, use of this function makes it possible to reduce divergence of the values of the ratios.

The results of the experiment allow us to state that the ranking of agents' evaluation was different for different periods. In the first and second period, the DeepLearningH2O was the best agent, in the first period MLP Agent was ranked higher, but was ranked lower than B&H benchmark in the second period. In the third period MLP Agent was ranked the highest, with DeepLearningH2O being ranked higher than B&H.

The highest value of evaluation function of DeepLearningH2O Agent (in first and second period) results from the highest *Average Rate of Return per Transaction* and low values of risk measures. The B&H benchmark was ranked lowest in all periods, it generated losses in the first and third periods. It should be noted that an upward trend was observed in the second period, hence B&H's *Rate of Return* was positive. The first and the third periods show a downward trend, implying negative *Rate of Return* for B&H. Taking all the periods into consideration, it may be stated that DeepLearningH2O Agent was ranked highest the most time. DeepLearningH2O Agent always was characterized by greater number of transactions than MLP Agent. DeepLearningH2O is characterized by higher *Rate of Return* than MLP Agent. Both, DeepLearningH2O Agent and MLP Agent are characterized by low level of risk.

VI. CONCLUSION

The deep learning model implemented as agent in A-Trader is used for supporting trading decision on FOREX market. Deep Learning proved to be a powerful, efficient and robust financial time series forecasting model. This model helps achieve better performance than MLP model. Experiments have shown that the error rate of time series forecasting has dropped significantly with CNN compared to other models. An important advantage of deep learning applications that makes them attractive to stock trading practitioners and researchers is the wide availability of high quality open source software, libraries and computation facilities.

Learning mode constitutes the main disadvantage of using Deep Learning Model, it is a time consuming process with a negative effect when near real time trading is applied [31]. It can be reduced, for example, by using a distributed cloud computing architecture. It was also interesting to observe the ability of deep learning model not only to dynamically adapt to network architecture but also to discover unknown features in raw financial time series.

The further research works may be related, among others, to optimization deep learning model by tuning values of parameters, and evaluating its performance. Additionally, models with different inputs vector structure could be developed, e.g. by using another number of log-rates-of-return or using weighted inputs.

REFERENCES

- [1] L. Mendes, P. Godinho and J. Dias, "A Forex trading system based on a genetic algorithm", *Journal of Heuristics* 18 (4), pp. 627-656, 2012.
- [2] F. H. Westerhoff, "Multi-Asset Market Dynamics", *Macroeconomic Dynamics*, 8/2011, pp. 596-616, 2011.
- [3] J.R. Thompson, J.R. Wilson and E. P. Fitts, "Analysis of market returns using multifractal time series and agent-based simulation", in *Proceedings of the Winter Simulation Conference (WSC '12)*. Winter Simulation Conference, Article 323, 2012.
- [4] C. D. Kirkpatrick and J. Dahlquist, *Technical Analysis: The Complete Resource for Financial Market Technicians*, Financial Times Press, 2006.
- [5] C. Lento, "A Combined Signal Approach to Technical Analysis on the S&P 500", *Journal of Business & Economics Research*, 6 (8), pp. 41-51, 2008.
- [6] O. Badawy and A. Almotwaly, "Combining neural network knowledge in a mobile collaborating multi-agent system", *Electrical, Electronic and Computer Engineering*, ICEEC '04, pp. 325, 328, 2004, DOI: 10.1109/ICEEC.2004.1374457.
- [7] P. R. Kaltwasser, "Uncertainty about fundamentals and herding behavior in the FOREX market", *Physica A: Statistical Mechanics and its Applications*, 389 (6), pp. 1215-1222, March 2010.
- [8] H. C. Aladag, U. Yolcu and E. Egrioglu, "A new time invariant fuzzy time series forecasting model based on particle swarm optimization", *Applied Soft Computing*, 12 (10), pp. 3291-3299, 2012.
- [9] P. Singh and B. Borah, "Forecasting stock index price based on M-factors fuzzy time series and particle swarm optimization", *International Journal of Approximate Reasoning*, 55 (3), pp. 812-833, 2014.
- [10] M. Aloud, E.P.K. Tsang and R. Olsen, "Modelling the FX Market Traders' Behaviour: An Agent-based Approach", in *Simulation in Computational Finance and Economics: Tools and Emerging Applications*, B. Alexandrova-Kabadjova, S. Martinez-Jaramillo, A. L. Garcia-Almanza and E. Tsang (eds.), IGI Global, 2012, pp. 202-228.
- [11] J. B. Glatfelter, A. Dupuis and R. Olsen, "Patterns in high-frequency FX data: Discovery of 12 empirical scaling laws", *Quantitative Finance*, 11 (4), pp. 599-614, 2011.
- [12] R.P. Barbosa and O. Belo, "Multi-Agent Forex Trading System", in *Agent and Multi-agent Technology for Internet and Enterprise Systems, Studies in Computational Intelligence*, vol. 289, 2010, pp. 91-118.
- [13] G. Batres-Estrada, *Deep Learning for Multivariate Financial Time Series*, Thesis of KTH Royal Institute of Technology, Stockholm, 2015.
- [14] E. Busseti, I. Osband and S. Wong, *Deep Learning for Time Series Modeling*, <http://cs229.stanford.edu/proj2012/BussetiOsbandWong-DeepLearningForTimeSeriesModeling.pdf>, 2012.
- [15] Y. Bengio, "Learning Deep Architectures for AI", *Foundations and Trends in Machine Learning*, 2 (1), 2009.
- [16] L. Takeuchi and L.Y. Ying, *Applying Deep Learning to Enhance Momentum Trading Strategies in Stocks* <http://cs229.stanford.edu/proj2013/TakeuchiLee-ApplyingDeepLearningToEnhanceMomentumTradingStrategiesInStocks.pdf>, 2013.
- [17] G. E. Hinton, S. Osindero and Y. W. Teh, "A fast learning algorithm for deep belief nets", *Neural computation*, 18(7), pp. 1527-1554.
- [18] T. Kuremoto, S. Kimura, K. Kobayashi and M. Obayashi, "Time Series Forecasting Using a Deep Belief Network with Restricted Boltzmann Machines" *Neurocomputing* 137 (2014), pp. 47-56, 2014.
- [19] J. Korczak, M. Hernes and M. Bac, "Fuzzy Logic as Agents' Knowledge Representation in A-Trader System", in E. Ziemba (ed.), *Information Technology for Management, Lecture Notes in Business Information Processing*, vol. 243, Springer International Publishing, 2016, pp. 109-124.

- [20] J. Korczak, M. Hernes and M. Bac, "Performance evaluation of decision-making agents' in the multi-agent system", in *Proceedings of Federated Conference Computer Science and Information Systems (FedCSIS)*, Warszawa, 2014, pp. 1171 – 1180. DOI: 10.15439/2014F188.
- [21] J. Korczak, M. Hernes and M. Bac, "Fundamental analysis in the multi-agent trading system", in *Proceedings of Federated Conference Computer Science and Information Systems (FedCSIS)*, Gdańsk, 2016, pp. 1171 – 1180. DOI: 10.15439/2014F188.
- [22] M. Hernes and N.T. Nguyen, "Deriving Consensus for Hierarchical Incomplete Ordered Partitions and Coverings", *Journal of Universal Computer Science* 13 (2), pp. 317-328, 2007.
- [23] M. Hernes and J. Sobieska-Karpińska, "Application of the consensus method in a multi-agent financial decision support system", *Information Systems and e-Business Management* 14 (1), Springer Berlin Heidelberg, 2016, DOI: 10.1007/s10257-015-0280-9.
- [24] P. D. McNelis, "Neural Networks in Finance: Gaining Predictive Edge in the Market", *Academic Press Advanced Finance Series*, Academic Press, Inc., Orlando, 2004.
- [25] V.V. Kondratenko and Y. Kuperin, *Using Recurrent Neural Networks To Forecasting of Forex*, arXiv:cond-mat/0304469 [cond-mat.disnn], 2003.
- [26] L. Di Persio and O. Honchar, "Artificial neural networks approach to the forecast of stock market price movements", *International Journal of Economics and Management Systems*, Volume 1, pp. 158-162, 2016.
- [27] L. Arnold, S. Rebecchi, S. Chevallier and H. Paugam-Moisy, *An Introduction to Deep Learning*. ESANN, 2011.
- [28] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 9, 2010 pp. 249– 256.
- [29] H. Lu, B. Li, J. Zhu, Y. Li, Y. Li, X. Xu, L. He, X. Li, J. Li and S. Serikawa, "Wound intensity correction and segmentation with convolutional neural networks", *Concurrency and Computation: Practice and Experience* 29, 2017.
- [30] Y.J. Cha, W. Choi and O. Büyükoztürk, "Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks", *Computer-Aided Civil and Infrastructure Engineering*, 32, pp. 361–378, 2017, doi:10.1111/mice.12263.
- [31] H. Larochelle and Y. Bengio, "Exploring strategies for training deep neural networks", *Journal of Machine Learning Research*, 1, pp. 1–40, 2009.
- [32] P. Hall, "How is deep learning different from multilayer perceptron?", <https://www.quora.com/How-is-deep-learning-different-from-multilayer-perceptron>, [access: 01.05.2017].
- [33] <https://github.com/h2oai> [access: 01.05.2017].
- [34] A. Candel, V. Parmar, E. LeDell, and A. Arora, "Deep learning with h2o, 2015, https://h2o-release.s3.amazonaws.com/h2o/rel-slater/9/docs-website/h2o-docs/booklets/DeepLearning_Vignette.pdf.