Resilient backpropagation ANN for predicting Bitcoin prices based on market depth data

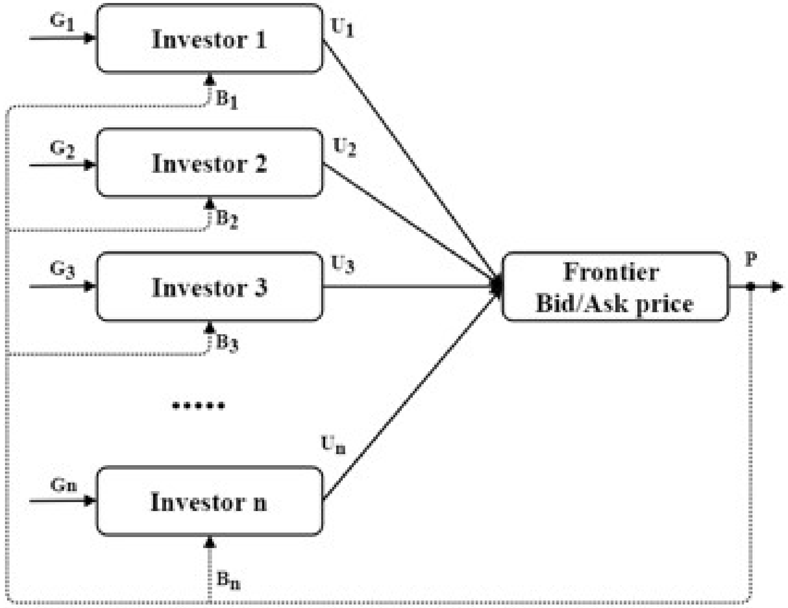
Milena Piryankova

Angel Marchev, Jr.

- Introduction
  - Problem definition
  - Artificial neural networks
- Methodology
  - Data
  - ANN architecture
- Results
  - Ask prices
  - Bid prices
- Conclusion

Problem definition

- Any financial market is a self-organized system, driven by real-time information (this paper mainly focuses on prices)
- Catallaxy is "the order brought about by the mutual adjustment of many individual economies in a market"
- Market depth is the total volume of orders. It helps to understand the idea of how much volume is possible to trade and how much the prices is likely to move.
- <u>Hypothesis: </u>Viable prediction could be made based on market depth data

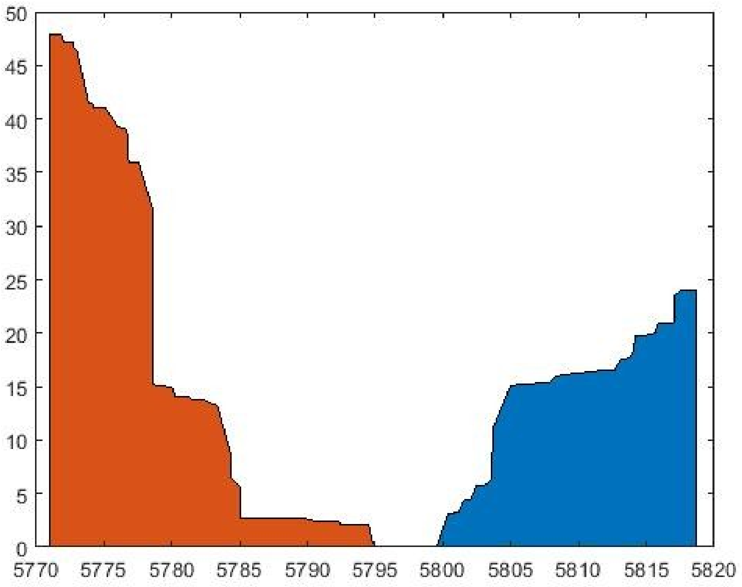


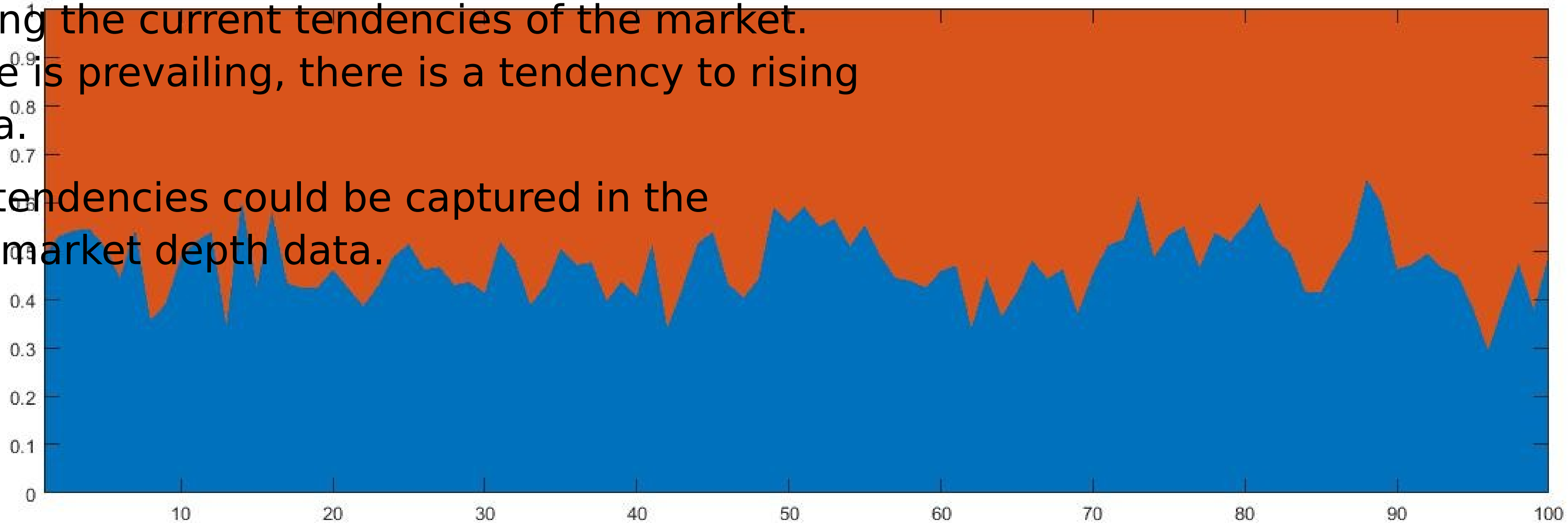
Where

- n – serial number of investor
- G – objectives of investors
- U – offers to the market by investors
- B – feedbacks to investors
- P – market price

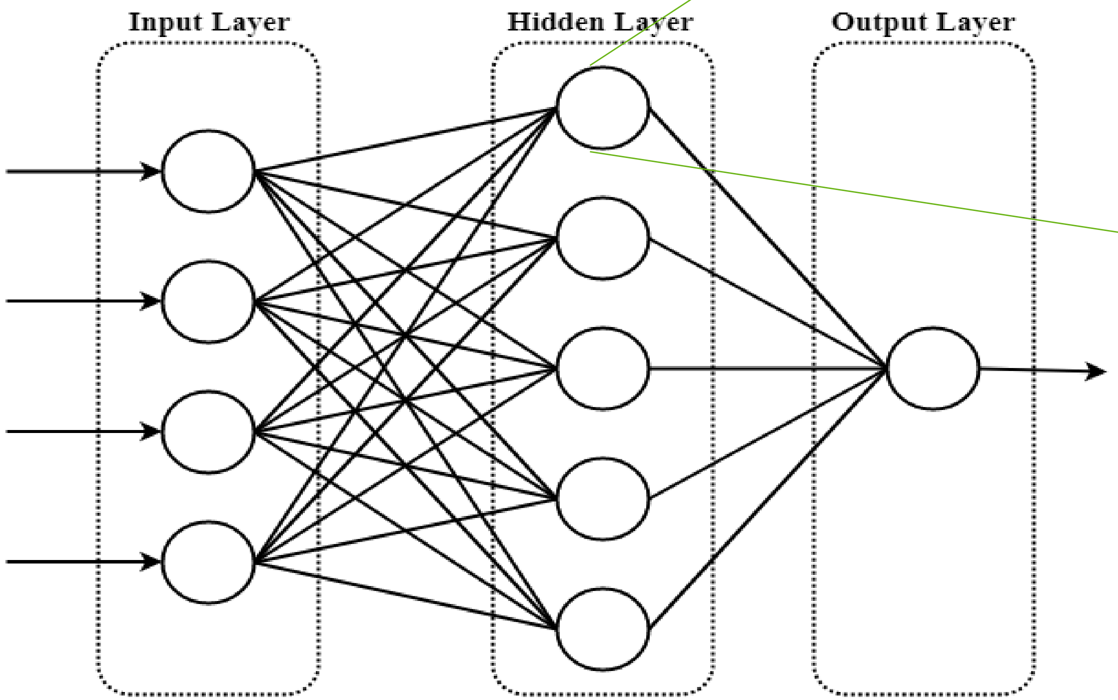
Demand and supply

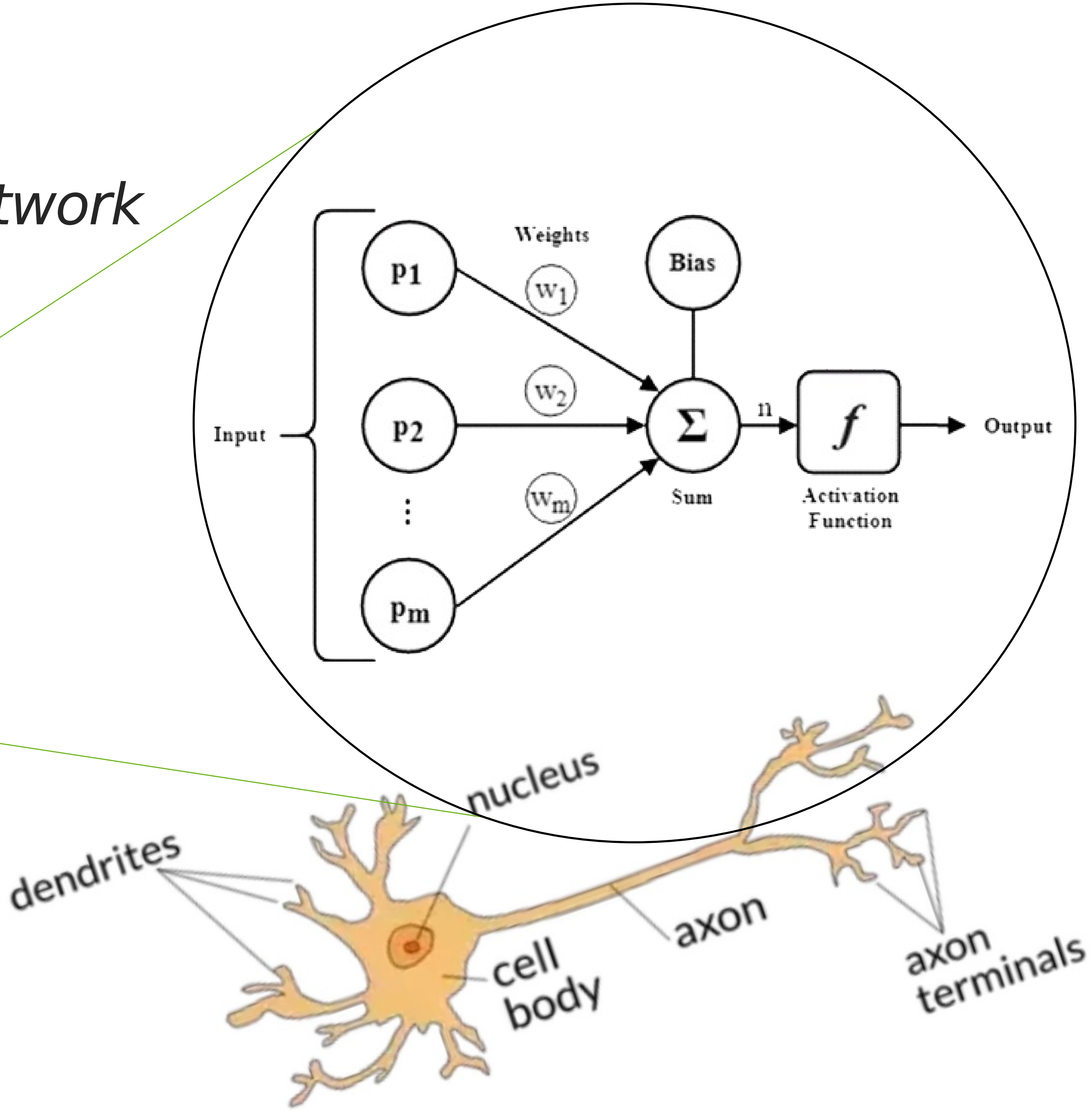
- The bid price represents the maximum price that a buyer is willing to pay
- The ask price represents the minimum price that a seller is willing to receive
- A transactions occurs when the seller and buyer agree on a same price.
- The ratio between the total outstanding volume of bid and ask offers is characterizing the current tendencies of the market. When the bid volume is prevailing, there is a tendency to rising prices and vice versa.
- These fundamental tendencies could be captured in the prediction based on market depth data.





Biological neuron, artificial neuron and artificial neural network





Artificial neural network as a computational model

- inspired by the network of neurons in biological nervous system;
- emulate the activity of the human brain as a combination of elementary computational elements into a large and complex system;
- consists of highly interconnected elements called neurons, each of which can solve simple mathematical function;
- functionality is mainly achieved by the connections among its neurons, utilizing their emergent properties;
- effective modelling of complex problems involving a large number of input variables;
- training is achieved by changing the weights of neuronal connections through algorithms (most used is backpropagation);
- once trained, other previously unknown signals can be predicted;
- major problem in training is overfitting - the model memorizes too specific patterns in the data it could lose the ability to generalize.

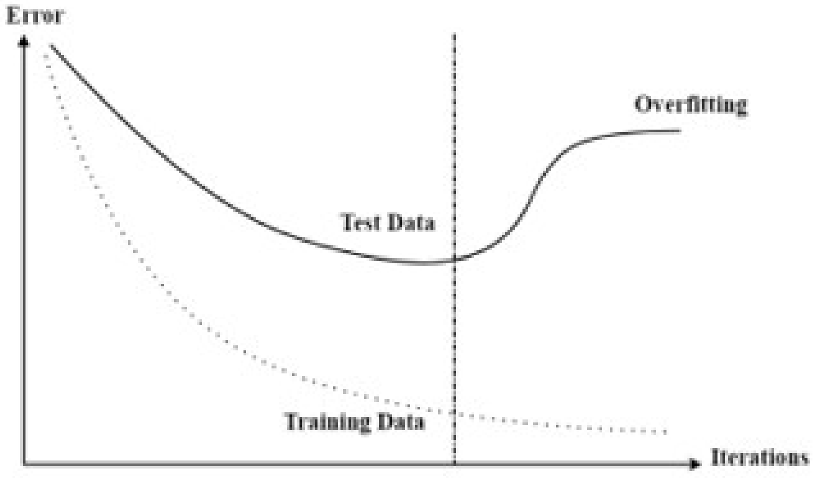


FIGURE 4. Overfitting ANN

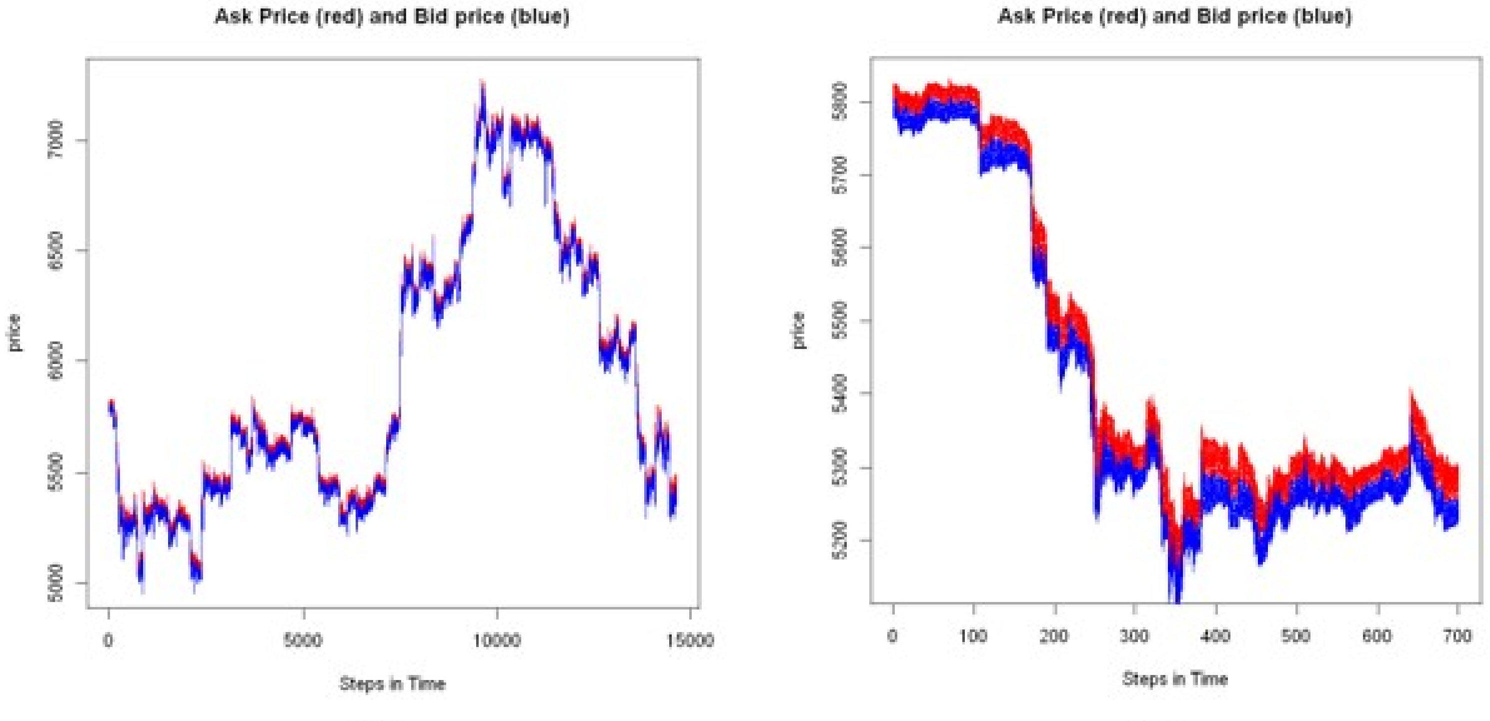
Data

datetime	ask_price_1	ask_price_2	ask_price_..	ask_price_50	bid_price_1	bid_price_2	bid_price_3	bid_price_..	bid_price_49	bid_price_50
21/06/2018 20:00	5800.01	5800.05	...	5822.31	5800.00	5799.99	5799.26	...	5777.76	5777.56
21/06/2018 20:05	5802.38	5802.43	...	5823.80	5802.37	5800.95	5800.58	...	5778.31	5778.17
21/06/2018 20:10	5804.75	5804.80	...	5825.28	5804.74	5801.90	5801.89	...	5778.86	5778.78
21/06/2018 20:15	5804.26	5804.29	...	5822.31	5799.12	5799.04	5799.00	...	5778.31	5777.56
21/06/2018 20:20	5798.38	5798.90	...	5823.42	5798.14	5798.11	5796.10	...	5775.92	5775.61
21/06/2018 20:25	5794.02	5794.05	...	5823.99	5793.31	5793.30	5793.02	...	5771.76	5771.61
21/06/2018 20:30	5791.38	5791.77	...	5817.90	5789.04	5789.01	5789.00	...	5767.68	5767.67
21/06/2018 20:35	5779.16	5779.18	...	5810.73	5779.15	5779.08	5778.35	...	5755.54	5755.00
21/06/2018 20:40	5782.39	5782.47	...	5814.40	5782.38	5780.00	5779.36	...	5755.54	5755.00
21/06/2018 20:45	5780.29	5780.32	...	5810.00	5780.28	5780.27	5780.00	...	5755.70	5755.54

TABLE 1. Data sample

- Bitcoin data
- 5-minute intervals;
- 14,640 observations
- Each observation represents an order book snapshot, representing 50 ask and 50 bid orders, where the last ask offer is the closest one to the first bid offer;
- First half observations used as training data, second half – only used in validation.

Data



(a) (b)

FIGURE 5. Data samples (a) full set (b) excerpt

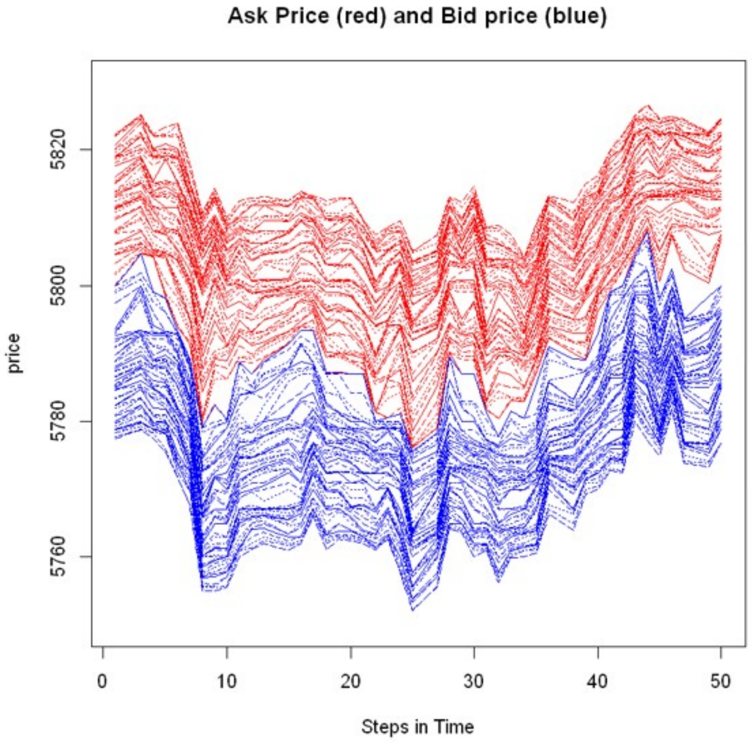


FIGURE 6. Zoom in

ANN architecture

- All connected ANN with backpropagation
- 100 input nodes (for each “current” ask and bid offers)
- Two hidden layers of 4 and 2 nodes
- Two output nodes – one each for bid and ask prices predictions one time ahead
- Bias node for each non-input layer

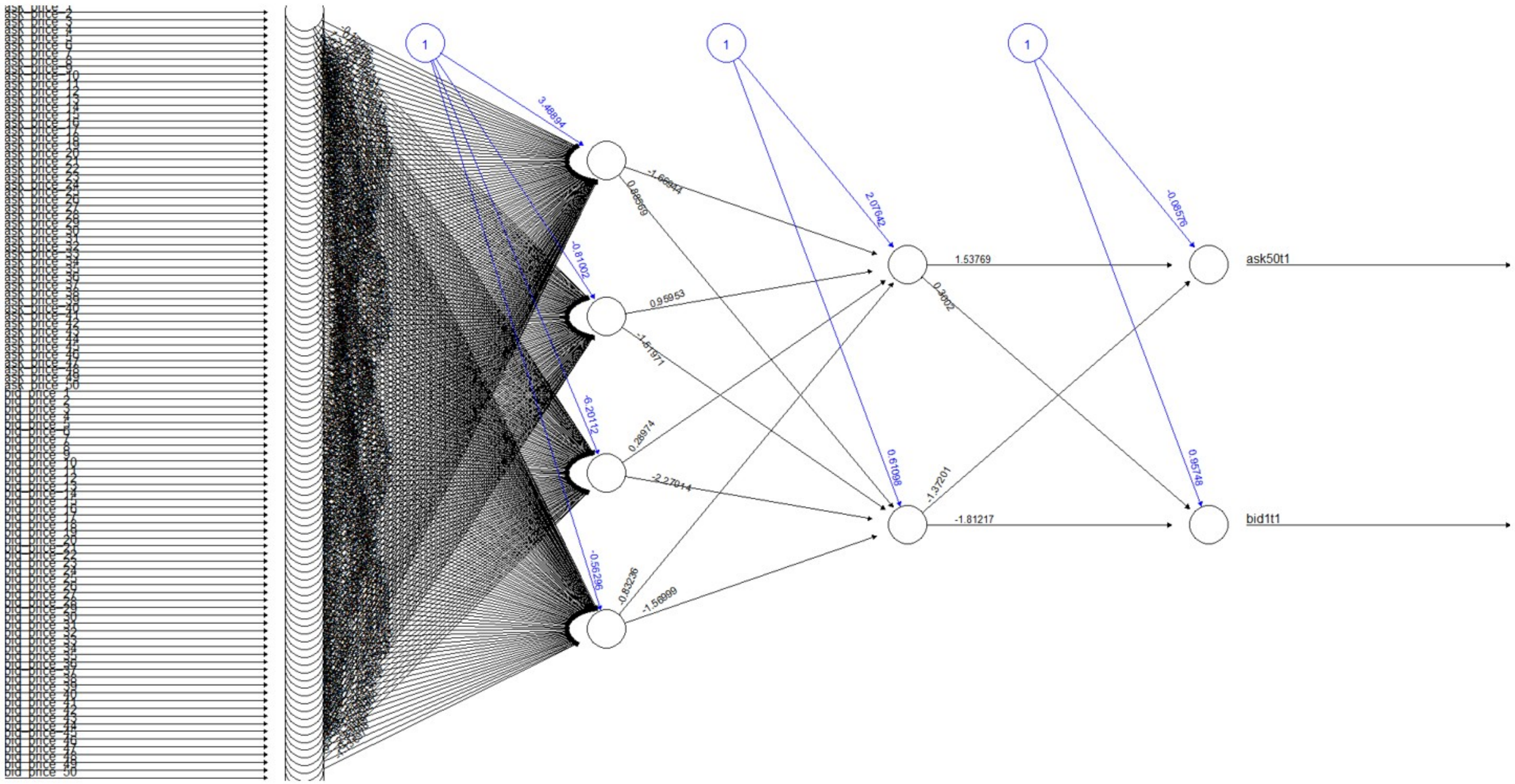


FIGURE 7. Architecture of the ANN used in this research

ANN analytical function

$$f\left(ask.price_{n_{t+1}}, bid.price_{1_{t+1}}\right) = f\left(ask.price_{1_t}\right) + f\left(ask.price_{2_t}\right) + \cdots + f\left(ask.price_{n_t}\right) + f\left(bid.price_{1_t}\right) + f\left(bid.price_{2_t}\right) + \cdots + f\left(bid.price_{n_t}\right) + b$$

where:

- n - number of the offer [1 to 50];
- $ask.price_{n_t}$- ask price which is the closest one to the bid price in the current moment;
- $ask.price_{n_{t+1}}$ - ask price which is the closest one to the bid price in the next moment;
- $bid.price_{1_t}$ - bid price which is the closest one to the ask price in the current moment;
- $bid.price_{1_{t+1}}$ - bid price which is the closest one to the ask price in the next moment;
- b – Bias

RPROP+ Resilient backpropagation with weight-backtracking

- A training algorithm specially designed to reduce the overfitting in backpropagation
- Igel and Husken,2003
- retracting a previous weight update for some of the current weights
- tracking the partial derivative of the corresponding weight
- change of sign indicates that the last weight update was too big
- if the sign stays the same, a regular weight update is executed.

For each ω_{ij}

if $\frac{\partial E}{\partial \omega_{\mathrm{ij}}}(t-1).\frac{\partial E}{\partial \omega_{\mathrm{ij}}}(t) > 0$, then

$$\Delta_{ij}(t) := min\left(\Delta_{ij}(t-1).\eta^{+}, \Delta_{max}\right)$$

$$\Delta\omega_{ij}(t) := -sign\left(\frac{\partial E}{\partial \omega_{\mathrm{ij}}}(t)\right).\Delta_{ij}(t)$$

$$\omega_{ij}(t+1) := \omega_{ij}(t) + \Delta\omega_{ij}(t)$$

elseif $\frac{\partial E}{\partial \omega_{\mathrm{ij}}}(t-1).\frac{\partial E}{\partial \omega_{\mathrm{ij}}}(t) < 0$, then

$$\Delta_{ij}(t) := max\left(\Delta_{ij}(t-1).\eta^{-}, \Delta_{min}\right)$$

$$\omega_{ij}(t+1) := \omega_{ij}(t) - \Delta\omega_{ij}(t-1)$$

$$\frac{\partial E}{\partial \omega_{\mathrm{ij}}}(t) = 0$$

elseif $\frac{\partial E}{\partial \omega_{\mathrm{ij}}}(t-1).\frac{\partial E}{\partial \omega_{\mathrm{ij}}}(t) = 0$, then

$$\Delta\omega_{ij}(t) := -sign\left(\frac{\partial E}{\partial \omega_{\mathrm{ij}}}(t)\right).\Delta_{ij}(t)$$

$$\omega_{ij}(t+1) := \omega_{ij}(t) + \Delta\omega_{ij}(t)$$

where

- ω_{ij} is weight between node i and node j
- E is measurement for error
- η is step size, depending on the partial derivative retaining the same sign for consecutive steps, where $0 < \eta^{-} < 1 < \eta^{+}$
- $\Delta_{max}, \Delta_{min}$ are bounding parameters for η
- *sign* is an operator which returns +1 if its argument is positive, −1 if the argument is negative, and 0 otherwise

Additional computations

- Normalization

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

where x is an original value,
x' is the normalized value.

- Denormalization

$$x'' = x' * (\max(x) - \min(x)) + \min(x)$$

where x is an original value, x' is the normalized value
and x'' is the denormalized value.

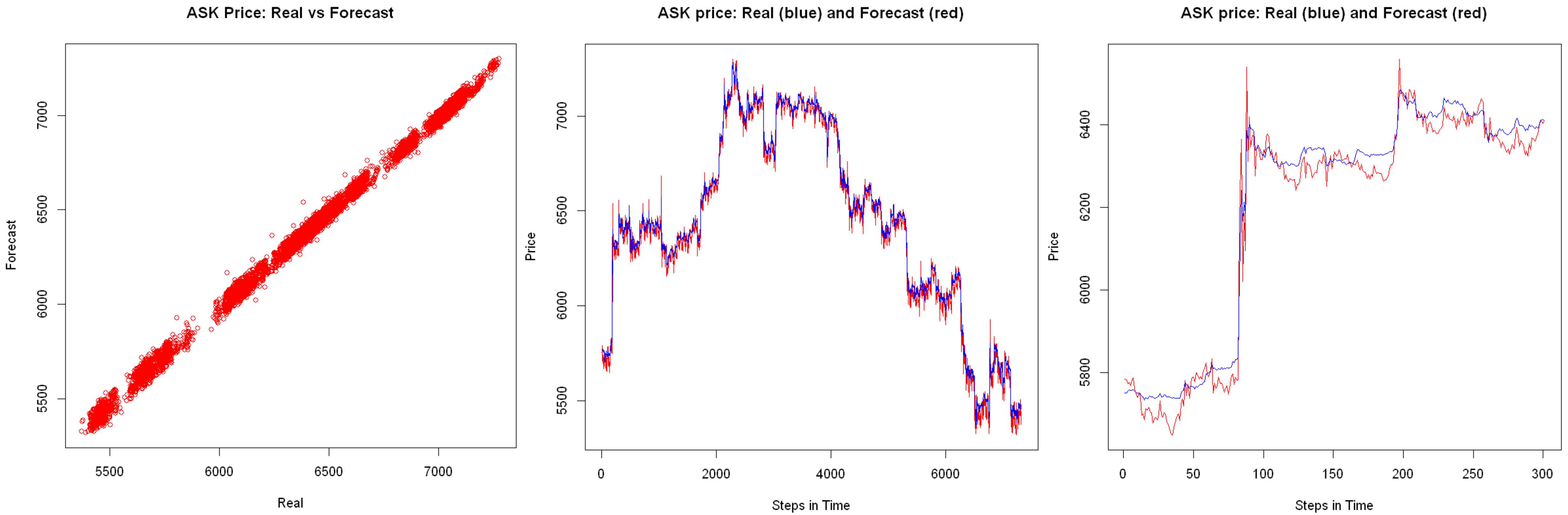
- Training evaluation – Sum of Squared Errors

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y_i})^2$$

- Validation – Mean Average Percentage Error

$$MAPE = \sum_{t=1}^{n} \left| \frac{Y_t - F_t}{Y_t} \right| \frac{100\%}{n}$$

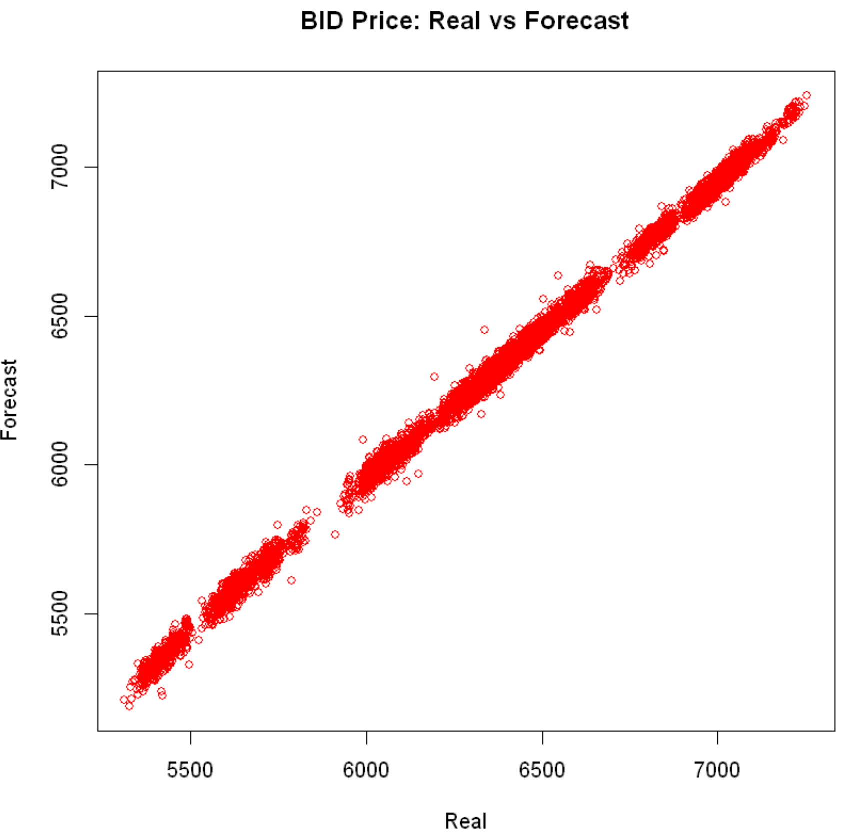
Results – Ask prices

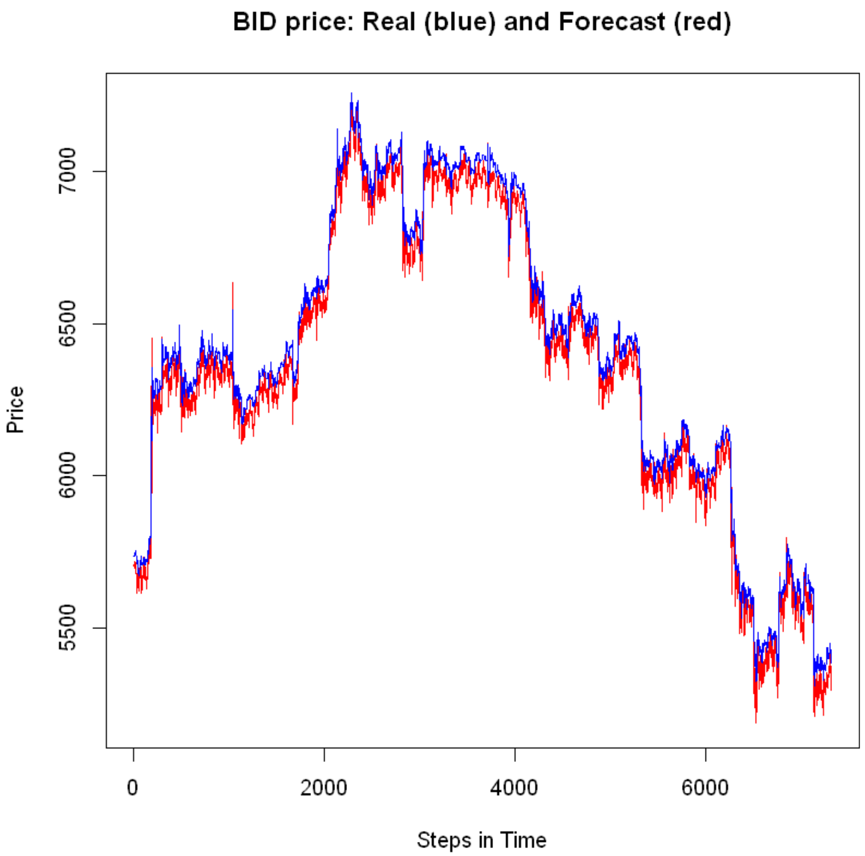


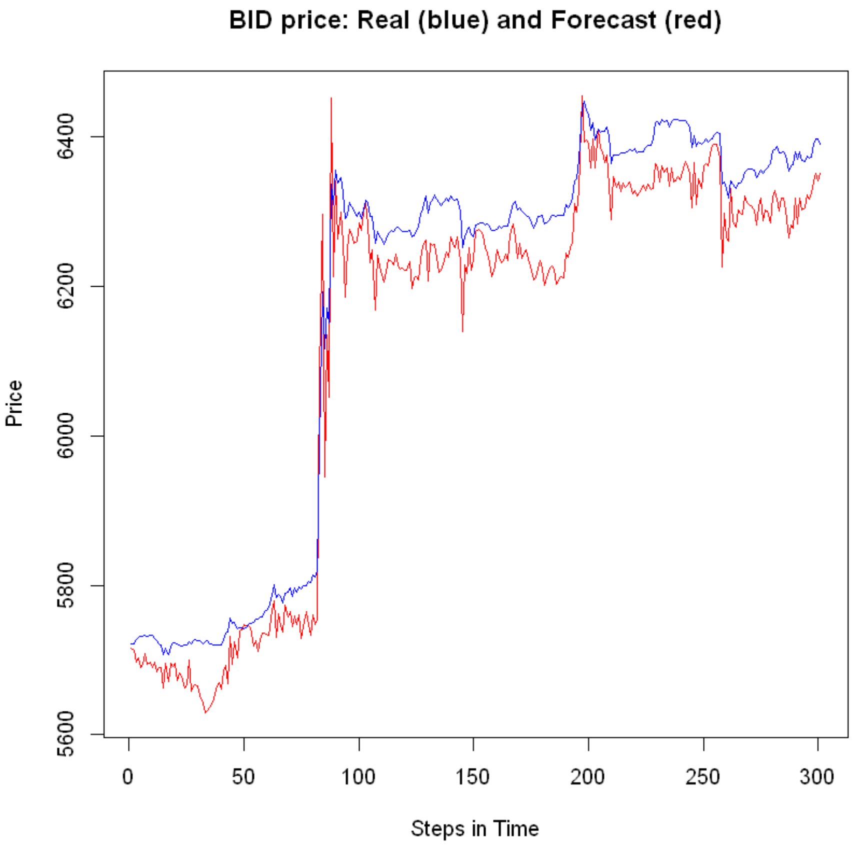
№ of observation	forecast_ask	real_ask	№ of observation	forecast_ask	real_ask
7321	5768.388	5756.50	14634	5376.621	5460.00
7322	5767.150	5757.75	14635	5401.927	5462.17
7323	5784.104	5759.23	14636	5413.330	5455.31
7324	5789.097	5759.57	14637	5406.282	5459.37
7325	5766.368	5760.00	14638	5393.238	5454.23
7326	5753.805	5760.79	14639	5398.716	5454.79

MAPEask
0.46%
(~24.83 €)

Results – Bid prices





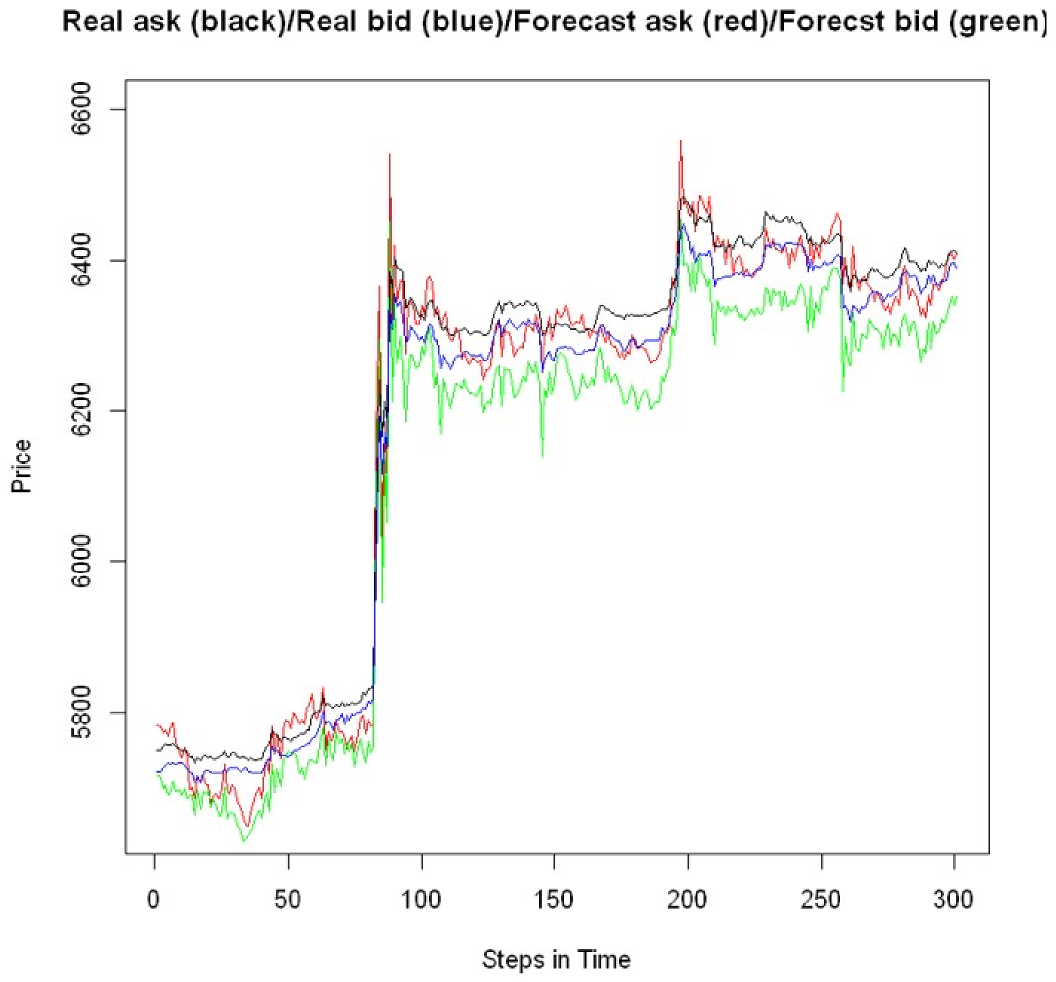


№ of observation	forecast_bid	real_bid	№ of observation	forecast_bid	real_bid
7321	5707.362	5735.16	14634	5309.542	5400.00
7322	5696.760	5736.95	14635	5322.176	5400.00
7323	5703.325	5736.98	14636	5320.662	5387.67
7324	5714.572	5736.99	14637	5296.957	5384.99
7325	5706.712	5741.36	14638	5327.599	5399.95
7326	5705.175	5743.92	14639	5323.248	5399.95

$MAPE_{bid}$ = 0.82% *(~44.27 €)*

Conclusions

- A viable prediction on bid and ask prices one time step ahead has been obtained
- Overall mean percentage error of less than 1%
- Although training/validation data split was implemented and despite using RPROP+ some overfitting visible at micro-level
- Relatively simple ANN architecture was the most suitable (after several experimented)



References

- Hayek, F. A. (1978), Law, Legislation, and Liberty, Vol. 2, pp. 108–09.
- Igel, C., M. Hüsken. (2003) Empirical evaluation of the improved Rprop learning algorithms. Neurocomputing, Volume 50, pp. 105-123, Elsevier
- Riedmiller, M. and Braun, H. (1993). ‘A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm’, IEEE Int. Conf. Neural Networks. 1. 586 - 591 vol.1. 10.1109/ICNN.1993.298623. Available from: https://ieeexplore.ieee.org/document/298623.
- Pagel, J. and Kirshtein, P. (2017). *Machine dreaming and consciousness*. 1st ed. Academic Press, pp.83-92., Available from: https://www.sciencedirect.com/book/9780128037201/machine-dreaming-and-consciousness
- Khan, A. and Augustine, P. (2019). Predictive Analytics in Cryptocurrency Using Neural networks: A Comparative Study. In: *International Journal of Recent Technology and Engineering (IJRTE)*. Blue Eyes Intelligence Engineering & Sciences Publication (BEIESP), pp.425-429. Available at: https://www.ijrte.org/wp-content/uploads/papers/v7i6/F2350037619.pdf
- Zhang, P. (2000). Neural Networks for Classification: A Survey. Systems, Man, and Cybernetics, Part C: Applications and Reviews, *IEEE Transactions on. 30*. pp. 451 - 462. 10.1109/5326.897072. Available from: https://ieeexplore.ieee.org/document/897072?arnumber=897072
- Stubbemann, J., Treiber, N., Kramer, O. (2015). Resilient Propagation for Multivariate Wind Power Prediction. ICPRAM 2015 - 4th International Conference on Pattern Recognition Applications and Methods, Proceedings., 10.5220/0005284403330337. Available from: https://www.researchgate.net/publication/268966338_Resilient_Propagation_for_Multivariate_Wind_Power_Prediction
- Kempf, A., Korn, O. (1999), “Market Depth and Order Size”, In *Journal of Financial Markets*. 2. 29-48. 10.1016/S1386-4181(98)00007-X. Available from: https://www.researchgate.net/publication/222446389_Market_Depth_and_Order_Size
- Silantyev, E. (2019), “Order flow analysis of cryptocurrency markets, Digital Finance”, 10.1007/s42521-019-00007-w., pp. 1–28 Available from: https://link.springer.com/article/10.1007/s42521-019-00007-w
- Donier, J., Bonart, J. (2015), “A Million Metaorder Analysis of Market Impact on the Bitcoin”, in *Market Microstructure and Liquidity, World Scientific Publishing Company,* 10.1142/S2382626615500082, pp.1-29., Available from: https://www.researchgate.net/publication/269636386_A_Million_Metaorder_Analysis_of_Market_Impact_on_the_Bitcoin
- Guo, T., Bifet, A., Antulov-Fantulin, N. (2018), “Bitcoin Volatility Forecasting with a Glimpse into Buy and Sell Orders”, *2018 IEEE International Conference on Data Mining (ICDM),* Available from: https://arxiv.org/abs/1802.04065
- Dixon, M., (2017), “Sequence classification of the limit order book using recurrent neural networks”, in *Journal of Computational Science*, pp. 277–286. Available at: https://arxiv.org/pdf/1707.05642.pdf
- Fischer, T.G., Krauss, C.; Deinert, A., (2019), “Statistical Arbitrage in Cryptocurrency Markets”, in *Journal Risk Financ. Manag.*, 10.3390/jrfm12010031, pp. 31-46 Available at: https://www.mdpi.com/1911-8074/12/1/31