Digital Currency Parity Rates Forecasting with Neural Networks

E. Eray Akkaya[1], A. Cihan Keles[2,3], Gokhan Sahin[3]

[1] Department of Mechatronics Engineering, Istanbul Gelisim University, Avcılar, Istanbul, Turkey
(E-mail: eeakkaya@gmail.com)

[2] Department of Physics, Yeditepe University, Atasehir, Istanbul, Turkey

[3] Department of Information Systems and Technologies, Yeditepe University, Atasehir, Istanbul, Turkey

Abstract.: In this decade digital currency, especially Bitcoin which is the first and most popular one become most used investing method by people and small companies. Its dynamics change rapidly therefore one can buy Bitcoins with less price and then sell later with higher price in a short period of time. However, predicting this period is very though. In this study presents the prediction Bitcoin/Euro exchange rates by using artificial neural networks. Neural networks can generalize from experience that needs high experience and devise trading rules that account for changing market conditions. Using neural network one can use the predictive information alone or with other available analytical tools to fit the trading style, risk property and capitalization. The neural network will help minimize above factors by simply giving an estimated exchange rate for a future day. Both Bitcoin / Euro exchange data used in the time interval from 2011 to 2016. (by the end of the 2016 December)

Keywords: Neural Networks, Digital Currency, Bitcoin, Time Series Analysis

1 Introduction

Bitcoin is becoming a popular payment method around the world. Nowadays, a lot of companies start to accept bitcoin when its reliability is increasing. Also, everyone especially European people like bitcoin in around the world. A lot of exchange websites and atm machines are available for exchange. Bitcoin was first invented by Satoshi Nakamoto who is anonymous name in 2009 and Bitcoins firstly made by mining blockchains using algorithm. Then mined bitcoin was started to exchange by the bitcoin miners in United States and spread later 2011 bitcoin can sell and buy in euro. Today, 1 Bitcoin is approximately equals to 1006 Euro when this research is made. However, parity rises and decreases nonlinearly because of its block halving and dynamics change rapidly therefore one can buy Bitcoins with less price and then sell later

Received: 14 April 2017 / Accepted: 12 December 2017
© 2017 CMSIM
ISSN 2241-0503

with higher price in a short period of time. However, predicting this period is very though. In this study presents the prediction of Bitcoin/ Euro exchange rates by using artificial neural networks. Neural networks can generalize from experience that needs high experience and devise trading rules that account for changing market conditions.

Neural Network systems are kind of computer systems which make learn and generate new data and predict some events from this data. These systems imitate human brain neurons behavior.

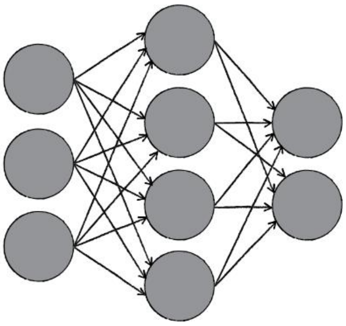

Fig.1. An example of Neural Network System

As can be seen in Fig.1. left side is input variables, it can be changed in every prediction works. For example, one can predict next month's weather conditions. He/she has for example temperature , humidity and wind power values. They are called input variables. Middle of the figure is hidden neurons. This can be changed according to simulation method which is apllied. Important part is output data

2 Used Neural Network Model

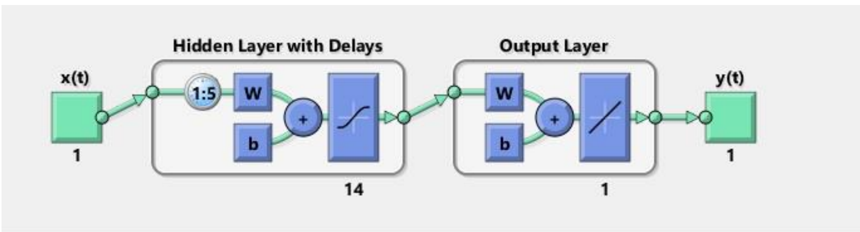

Fig.2. Used model for predicting Euro/BTC parity values.

In this study, we used MATLAB® software program and its default program which is Neural Network Start program toolbox. This toolbox easily make predictions by very friendly user interference program.
In Fig.2. shows the diagram of the neural network model which is used for this Bitcoin study. 1:5 sign show us number of delay and 14 means number of hidden errors. As mention in introduction neural network simulations use hidden layers for output. You cannot increase number or decrease by yourself. Optimum hidden of layer is important for prediction. Our study, we observed Euro versus Bitcoin parity. So, we took times series Euro/BTC data as input. Also for nonlinear time series prediction in neural network, you must use targets. Therefore, we used United Stated Dollars versus Bitcoin parity data is used as target value. Same size, same date but dollar parity is used. After then, we retrained twenty times to make correlation is plausible. You can see error histogram in figure
Neural Network uses Time series equation below:

$$y(t) = f(x(t-1), \dots \dots, x(t-d) \tag{1}$$

y(t) series is our predicted result, *x(t)* input time series data and *d* is delay number.
You can see the script which is toolbox used when you make selections.

```
% y(t+1) once x(t) is available, but before the actual y(t+1) occurs.
% The network can be made to return its output a timestep early by removing one delay
% so that its minimal tap delay is now 0 instead of 1. The new network returns the
% same outputs as the original network, but outputs are shifted left one timestep.
nets = removedelay(net);
[xs,xis,ais,ts] = preparets(nets,inputSeries,targetSeries);
ys = nets(xs,xis,ais);
earlyPredictPerformance = perform(net,tc,yc)
% Solve an Input-Output Time-Series Problem with a Time Delay Neural Network
% Script generated by NTSTOOL.
% Created Wed Apr 05 14:53:14 AST 2017
%
% This script assumes these variables are defined:
%
%   euro2kasim - input time series.
%   usd2kasim - target time series.

inputSeries = tonndata(euro2kasim,false,false);
targetSeries = tonndata(usd2kasim,false,false);
```

```
% Create a Time Delay Network
inputDelays = 0:4;
hiddenLayerSize = 14;
net = timedelaynet(inputDelays,hiddenLayerSize);

% Prepare the Data for Training and Simulation
% The function PREPARETS prepares timeseries data for a particular network,
% shifting time by the minimum amount to fill input states and layer states.
% Using PREPARETS allows you to keep your original time series data unchanged, while
% easily customizing it for networks with differing numbers of delays, with
% open loop or closed loop feedback modes.
[inputs,inputStates,layerStates,targets] = preparets(net,inputSeries,targetSeries);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;


% Train the Network
[net,tr] = train(net,inputs,targets,inputStates,layerStates);

% Test the Network
outputs = net(inputs,inputStates,layerStates);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, plotresponse(targets,outputs)
%figure, ploterrcorr(errors)
%figure, plotinerrcorr(inputs,errors)

% Early Prediction Network
% For some applications it helps to get the prediction a timestep early.
% The original network returns predicted y(t+1) at the same time it is given x(t+1).
% For some applications such as decision making, it would help to have predicted
```

Levenberg-Marquardt calculation is utilized for second-order preparing speed without computing the Hessian network.

Hessian code approximated as

$$H = J^T J \tag{2}$$

$$g = J^T e \tag{3}$$

Where J is the Jacobian matrix that contains first order derivatives of the system errors as indicated by weights and inclinations network errors is described with *e*

So, the Levenberg-Marquardt algorithm takes this approximation like Newton:

$$x_{k+1} = x_k - \left[J^T J + \mu I\right]^{-1} J^T e \tag{4}$$

This algorithm typically requires more memory however, less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

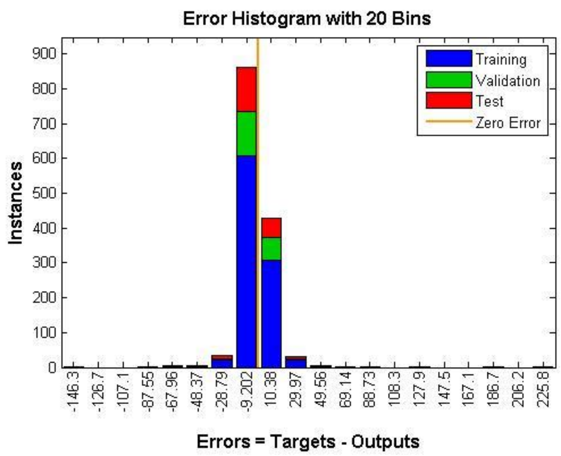

Fig. 3. Error Histogram of Neural Network Prediction Simulation with 20 retrains

In Fig.3. Error histogram provides us validation of our simulation. We can say that our simulation mechanism can be acceptable for further study.

3 Prediction of Parity values and comparison

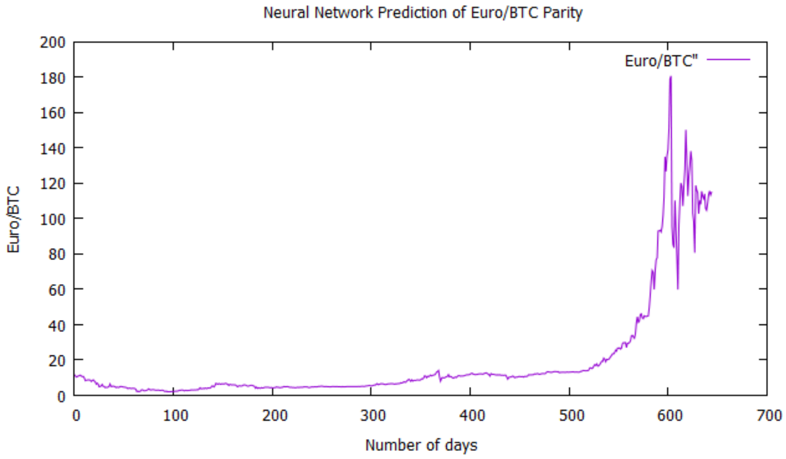

Fig. 4. Prediction Result of Neural Network Simulation

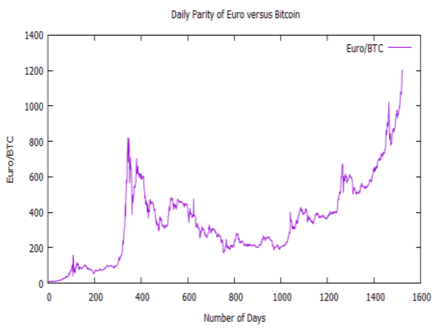

Fig. 5. Raw data Euro/BTC

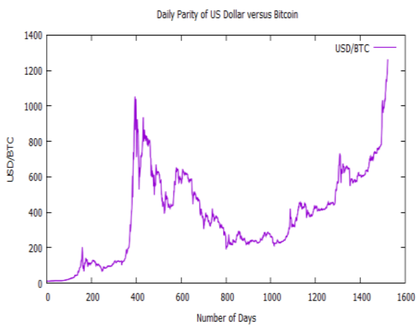

Fig. 6. Raw data USD/BTC

Fig.4. shows us the prediction result of Euro/BTC parity values by using neural network toolbox of MATLAB. It predicts seven hundred future days. Fig.5 and Fig.6. are in order raw data of Euro/BTC and USD/BTC. Fig.7 is DFA analysis graph by using raw data from Euro/BTC parity values and Fig.8. again, DFA analysis graph by using predicted result from neural network system. As you

can see on figures their slopes are nearly same which are , 1.14 in historical data and 1.48 in predicted results. However, prediction values do not be expected in bitcoin financial time series flow. Recently as mentioned before, 1 Bitcoin is approximately 1000 Euros and 20 Euros prediction is very though. It will be determined in conclusion part.

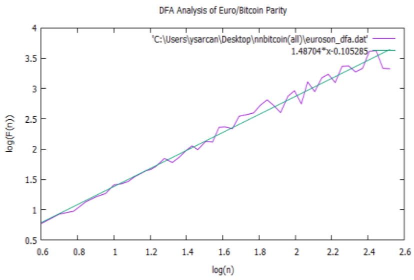

Fig. 7. DFA Analysis of Euro/BTC parity in history

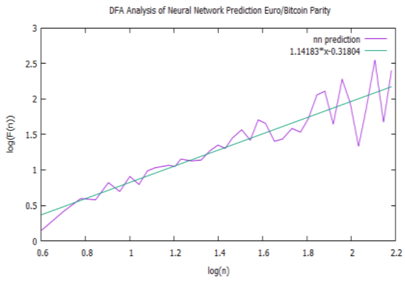

Fig. 8. DFA Analysis Euro/BTC from Neural Network Prediction Analysis

4 Conclusion

Bitcoin parity values in average daily time series show chaos in previous works by using nonlinear techniques and positive maximal Lyapunov exponents were calculated. In this work we try artificial neural network predictions to be sure its chaotic dynamics. The results look consistent with detrended fluctuation analysis. Dfa analysis slope for neural network prediction data is 1.14 which can be seen in Fig.8 and raw data of Euro versus bitcoin's dfa slope is 1.48 which can be seen in Fig.7 that we can say nearly same However, when you look at Fig.4, the first 500 days of the 20 Euro for 1 bitcoin, it seems not make sense, even if it is a block halving event. But we can say that if people around the world will stop buying and start to rapidly sell, there is a possibility of falling. We can say this for 20 euro in this sense, but the value after 500 days can actually be applied or profitable. However, previous works, finding maximal positive Lyapunov exponents are shown that bitcoin parity values are chaotic and Detrended fluctuation analysis' slope values similarity also supported chaos in Bitcoin

References

1.Bitcoin and Digital Currencies Paperback – July 1, 2013
by James Cox (Author)

2. Neural Network Toolbox for use with Matlab Howard Demuth Mark Beale
The Math Works 2002

3. Nonlinear Dynamical Systems Analysis for the Behavioral Sciences Using Real Data

4. Nonlinear Time series Analysis Holger Kantz Thomas Schreiber

5. Bitcoin: A Peer-to-Peer Electronic Cash System Satoshi Nakamoto satoshin@gmx.com www.bitcoin.org

6. Yapay Sinir Aglari, Prof.Dr. Ercan Öztemel, Papatya Publisher, 2003 ISBN: 9789756797396

7. H.Ahmet Yildirim, Avadis S. Hacinliyan, Ergun Eray Akkaya, "Chaos in Digital Currency Markets", CHAOS2014, Lisbon Portugal.