JACOBS
UNIVERSITY

Exploring the effects of different selections of inputs on the accuracy of predictions of Bitcoin using Echo State Network

by

Sabin Bhandari

Bachelor Thesis in Computer Science

Prof. Herbert Jaeger
Name and title of the supervisor

Date of Submission: May 16, 2018

With my signature, I certify that this thesis has been written by me using only the indicated resources and materials. Where I have presented data and results, the data and results are complete, genuine, and have been obtained by me unless otherwise acknowledged; where my results derive from computer programs, these computer programs have been written by me unless otherwise acknowledged. I further confirm that this thesis has not been submitted, either in part or as a whole, for any other academic degree at this or another institution.

Signature Place, Date

Abstract

Right from its inception in 2008, Bitcoin has been influencing the financial market significantly. Its impact needs to be taken into account and for this, it is essential to understand the dynamics of Bitcoin. This is the intention behind this research.
Many other machine learning algorithms, as well as Recurrent Neural Networks (RNN), have been used in the field of financial price prediction. This guided research will explore the features of the fluctuation of Bitcoin price and then use the Echo State Network (ESN) to build the accuracy of the prediction. The novelty of the strategy lies in utilizing ESNs which are generally faster and significantly less demanding to prepare than other traditional RNNs. They are suitable for prediction of time series and an efficient method for learning about non-linear systems.
Bitcoin is one of the most popular cryptocurrencies, and its digital nature means that high-dimensional network and pricing data related to it are widely available. Thus, this research aims to analyze how well the ESN is suited for the prediction of Bitcoin price while using the weighted price and the volume of BTC as the input. We will also be observing the practical outcome of the Echo State Network.

Contents

1 Motivation of Research **1**

2 Introduction **1**
2.1 Bitcoin . . . 1
2.2 Price Prediction . . . 1
2.3 Prior Work . . . 2
2.4 Echo State Network . . . 3
2.5 Objective . . . 3

3 Mathematical Formula and Theory of ESN **4**
3.1 System equations . . . 4
3.2 Learning Equations . . . 5
3.3 Echo states . . . 5

4 Global Control parameters **6**
4.1 Size of the Reservoir . . . 6
4.2 Spectral Radius . . . 6
4.3 Input scaling . . . 6

5 Design Of Experiment **7**
5.1 Dataset Description . . . 7
5.2 Feature Selection . . . 7
5.2.1 Time differencing . . . 8
5.2.2 Fast Fourier Transform . . . 8
5.2.3 Energy . . . 9
5.2.4 Volume BTC . . . 9
5.3 Data Preprocessing . . . 10
5.3.1 Interpolation . . . 10
5.3.2 Exponential Smoothing . . . 11
5.3.3 Log scale and Standardization . . . 11
5.3.4 Profile of the price signal . . . 11
5.4 Network Setup . . . 12
5.5 Training Phase and K fold cross-validation . . . 13
5.6 Optimizing parameters . . . 13
5.7 Testing Phase . . . 13
5.8 Principle Component Analysis (PCA) . . . 14
5.9 Error Calculation . . . 14

6 Results **14**
6.1 Training Phase . . . 14
6.2 Testing Phase . . . 15
6.3 PCA . . . 16

7 Discussion **17**
7.1 Analysis of results . . . 17
7.2 Possible Features . . . 18
7.3 Future Work . . . 19

8 Conclusion **19**

1 Motivation of Research

Apart from making transactions easier, Bitcoins are impacting the real economy. The recent surge in the financial market because of Bitcoins can make one realize that while people who have invested in it can benefit greatly, people who are not as involved are also missing out on a great deal. One can be motivated to carry out the research not only for the possible benefits from the investment in cryptocurrencies but also as a way to remain informed about the extent of the impact that it continues to have on the financial market. The motive is also to pave the way for subsequent researchers to further develop on prediction methods that yield higher accuracy and fulfill the limitations of those methods to an extent. The evolution of cryptocurrencies is something that needs to be taken into consideration while looking at the future of financial services. In addition, being able to predict the pricing of a virtual currency that is as popular as Bitcoin would be certainly useful in the current scheme.

2 Introduction

2.1 Bitcoin

Bitcoin is an independent digitized form of payment that functions by peer-to-peer transactions [14]. There is no central authority monitoring this currency which means that no single body can make changes to the monetary policies and consequently, cause a meltdown [21]. It operates with an underlying support system of a distributed network of users and relies on methods of advanced cryptography to maintain stability and reliability.
Bitcoins can be described merely as chains of digital signatures that are stored in a "wallet" file. The chain of signatures contain all the background information on the Bitcoin that is required while transferring it from one user to another. The purpose of this characteristic of Bitcoin is to ensure legitimacy. A Bitcoin user's "wallet" will consist of a private and a public key, apart from the Bitcoins themselves. The public key is what other users require to transfer Bitcoins to you, and the private key is what enables you to send your Bitcoins to someone else. The network also preserves anonymity, and there are nodes that verify the transaction as having occurred at a specific time which resolves the issue of users trying to spend the Bitcoin twice. In short, Bitcoin is a form of virtual currency that enables fast and cheap transactions [19].
Bitcoin was released as an open-source software in 2009. It was invented by an unknown person or group of people under the name Satoshi Nakamoto [22]. As of now, Bitcoin is the highest valued cryptocurrency in the market.

2.2 Price Prediction

Bitcoin is a digital asset, and its market is a financial or stock market. Due to this, a large amount of data on Bitcoin relating to different aspects (such as OHLC (Open High Low Close) data, volume, hash rate, etc.) is collected over time. By analyzing regularities and patterns on the collected data would offer potential profits for market participants such as traders and investors. With analysis and research, the precision of investment decisions could be positively affected. There are many prediction methodologies applied in

the world of Bitcoin, as more and more people develop an interest in investing in cryptocurrencies. Thus, it is essential to leverage the prediction of Bitcoin by using different machine learning techniques [14].

2.3 Prior Work

The market price of Bitcoin fluctuates quite a lot, and due to this volatility, there are very few price prediction methods that are efficient. In 2016, Amjad and Shah [2] tested various prediction algorithms against the classical technique, ARIMA (Autoregressive Integrated Moving Average) to compare their efficiency and accuracy. Out of the four prediction algorithms, Random Forest seemed to yield the best accuracy (79%) while ARIMA seemed to perform significantly poorly on all metrics. The method worked as they chose a training period eight months before the testing period and the algorithms were also trained on more recent data. The algorithms incorporated a richer feature set allowing them to capture a more abundant amount of data as compared to the ARIMA model. This is not to say that there are no limitations to this approach. The most obvious ones are that they disregard the impact of trading decisions on the equilibrium of the market. Since this approach only considers a single Bitcoin, the limitation arises when the quantity is increased.
In another paper, Shah and Zhang [16] utilized Bayesian regression for "Latent Source Model" [6] in predicting the Bitcoin prices. In developing a high frequency trading strategy they used advanced statistical methods and employed parallel computing to process a huge amount of data. Thus, when run against real data trace, their strategy is able to nearly double the investment in less than a 60 day period. There are three important things to note from their research (1) Is their strategy scalable? (2) Is their computation scalable for large amount of investments? (3) Will their model be able to correctly predict the price of Bitcoin, owing to the fact that there are high volatility and fluctuation in a short span of time? The domain of scalability of strategy remains unexplored where more research should be done from their side. In order to make their model more accurate they had to use all the time series, however this introduces complexity overhead.
In another similar study by Madan, Saluja and Zhao [14] a binary classification algorithm was tested. In phase one, they used binomial general linear models (GLM), Support Vector Machine (SVM) and Random Forest to predict the sign of price change. They obtained 98.7% accuracy in this phase. Binomial GLM was found to be more effective than the other two algorithms. In second phase, binomial GLM and Random Forest were used where Random Forest outperformed GLM. It can be inferred from the results that Random Forest may have generated better accuracy (57.4%) because they used non-parametric decision trees. Hence, Random Forest method was able to separate the outliers and the separability of data. The most obvious limitation to this method was that it takes into account the price change only at a single point of time instead of changes over real time. The authors agreed that there were several improvements that they could incorporated to make this study better, i.e. examining patterns of subsets of the price data and trimming the training set of data to a set of pattern that is similar to the output pattern.
Lonno and Stenqvist [18] used the sentiment analytic software VADER to predict Bitcoin price change from tweets about Bitcoin. The prediction model evaluation showed that aggregating tweet sentiments over a 30 min period with four shifts forward, and a sentiment change threshold of 2.2%, yielded a 79% accuracy. The data that the predictions were based on were collected over the span of a month, and it can be argued that this amount

of data is not sufficient. Additionally, the Lexicon was an all-round Lexicon which meant that it was filtering the sentiments for the most common social media expression. The number of predictions was not sufficient enough to draw conclusions about the prediction model, despite the high accuracy that it yielded. Also, VADER is not explicitly designed to perform sentiment analysis of Bitcoin, and hence may not produce accurate sentiment scores for the tweets containing trading terms (such as rocket or moon indicating price growth) specific to Bitcoin.

2.4 Echo State Network

Recurrent Neural Networks (RNNs) represent the broad class of neural networks whose computational model is designed analogously to the biological working model of the brain. In RNNs, there is a presence of closed cycle connection topology with self-sustained dynamics. The abstract neurons are connected by abstract synaptic connections which enables the waves/echoes of activation to propagate through the network [12]. However, the supervised learning technique is difficult in RNNs as it is computationally expensive and difficult to optimize [1].
ESN, a special kind of RNNs, is a comparatively cheaper and faster method of supervised learning [12]. It consists of input vectors, weights (input, output and reservoir) and output vectors. A random reservoir which acts as a memory unit is created with random connections and weights. Then the output weights are trained through a recurrent process of applying a sigmoid function on the current state of the reservoir, along with the teacher and the input. The basic idea of ESNs is shared with Liquid State Machines (LSM), which were developed independently from and simultaneously with ESNs by Wolfgang Maass [13]. The learning rule for RNNs [15] such as LSMs, ESNs and Backpropagation Decorrelation are classified under Reservoir Computing [7].

2.5 Objective

The main objective of the research is to neither compete with the state of the art Bitcoin prediction methods nor to provide an alternate method of prediction. Rather, this research aims at analysing how well can ESN perform in the prediction task with different features from the data. The objectives of this research can be summarized in the following points:

- To evaluate how Echo State Network (ESN) is set up best for prediction of Bitcoin price using different features obtained from the data.
- To ascertain with what accuracy can the price of Bitcoin be predicted with carefully tuned ESN.

3 Mathematical Formula and Theory of ESN

3.1 System equations

For the practical implementation of the ESN, discrete-time neural networks with K inputs, N reservoirs, and L outputs are taken. In our case, K=7, N=100 and L=7. The mathematical formalism for the following ESN network is taken from [8] and [9] where,

$$\mathbf{u}(n) = (u_1(n), ..., u_K(n))^t$$

is the K-dimensional input signal.

$$\mathbf{x}(n) = (x_1(n), ..., x_N(n))^t$$

is the N-dimensional reservoir state.

$$\mathbf{y}(n) = (y_1(n), ..., y_L(n))^t$$

is the L-dimensional output signal.

W is the $N \times N$ reservoir weight matrix, $\mathbf{W}^{in}$ is the $N \times K$ input weight matrix, and $\mathbf{b}$ is an N-dimensional bias vector. f is a sigmoid function, where in this case it is tanh function. The state update equation for the activation of internal units is given by

$$\mathbf{x}(n+1) = f(\mathbf{W}\mathbf{x}(n) + \mathbf{W^{in}}\mathbf{u}(n+1) + \mathbf{b}). \tag{1}$$

The extended system state is obtained by the concatenation of the reservoir and input states at time n. It is given by

$$\mathbf{z}(n) = [\mathbf{x}(n); \mathbf{u}(n)]. \tag{2}$$

The output is obtained from the extended system state by

$$\mathbf{y}(n) = g(\mathbf{W}^{out}\mathbf{z}(n)). \tag{3}$$

Here, g is an output activation function (in this case an identity) and $\mathbf{W}^{out}$ is a $L \times (K+N)$-dimensional matrix of output weights.
The output units may optionally project back to internal units with connections whose weights are collected in a $N \times L$ backprojection weight matrix.

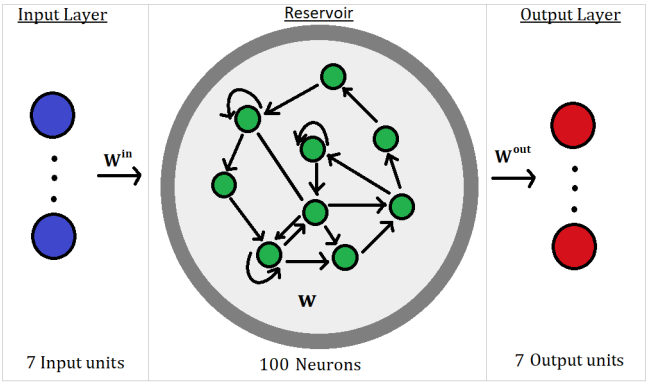


Figure 1: The basic schema of an ESN.

3.2 Learning Equations

In the state harvesting stage of the training, the ESN is driven by an input sequence $\mathbf{u}(1), ..., \mathbf{u}(n_{max})$, which yields a sequence $\mathbf{z}(1), ..., \mathbf{z}(n_{max})$ of extended system states. Even though there is sufficient data, I chose to work with 350522 (raw data points) for simplicity. The obtained extended system states are filed row-wise into a state collection matrix (SCM) $\mathbf{S}$ of size $(n_{max} \times (N + K))$. Usually, some initial portion of the collected states is discarded to accommodate for a washout of the arbitrary (random or zero) initial reservoir state needed at the time. Likewise, the desired outputs $\mathbf{d}$(n) are sorted row-wise into a teacher output collection matrix $\mathbf{D}$ of size $(n_{max} \times L)$ [7]. In this learning task, the main aim is to compute $\mathbf{W}^{out}$ by minimizing the mean squared error

$$\mathbf{MSE} = \frac{1}{n_{max}} \sum_{n=1}^{n_{max}} |\mathbf{d}(n) - \mathbf{W}^{out}\mathbf{x}(n)|^2. \tag{4}$$

The desired output weights $\mathbf{W}^{out}$ are the linear regression weights of the desired outputs $\mathbf{d}$(n) on the harvested extended states $\mathbf{z}$(n). Ridge regression (also known as Tikhonov regularization) is used as it improves the numerical stability and mitigates sensitivity to noise and overfitting [12]. The output weights $\mathbf{W}^{out}$ calculated as

$$\mathbf{W}^{out} = \left(\frac{\mathbf{S}^t\mathbf{S}}{n_{max}} + \alpha\mathbf{I} \right)^{-1} \frac{\mathbf{SD^t}}{n_{max}} \tag{5}$$

where $\frac{\mathbf{S}^t\mathbf{S}}{n_{max}}$ is the correlation matrix of the extended reservoir states with $\mathbf{S}$ being SCM , $\frac{\mathbf{SD}^t}{n_{max}}$ cross-correlation matrix of the states, $\mathbf{D}$ is the desired output matrix, $\mathbf{I}$ is the identity matrix and α is the regularization coefficient.

3.3 Echo states

Under certain conditions, the activation state $\mathbf{x}(n)$ of a recurrent neural network (RNN) is a function of the infinite input history presented to the network [9]. The activation state of the network is calculated on the basis of the echo function $\mathbf{E}$ as

$$\mathbf{x}(n) = \mathbf{E}(..., \mathbf{u}(n-1), \mathbf{u}(n)). \tag{6}$$

Proposition 1 *Assume a sigmoid network with unit output functions f_i = tanh. Let the weight matrix W have its largest singular value $\sigma max < 1$. Then the network has echo states for all admissible inputs. Let the weight matrix have a spectral radius $|\lambda max| > 1$, where the spectral radius denotes the eigenvalue of the weight matrix with the largest absolute value. Then this network has no echo states if $\mathbf{u}(\mathbf{n})$ = 0 is an admissible input sequence.*

From the above proposition taken from [9], when we scale the weight matrix $\mathbf{W}$ by a scaling factor of Wsf, this would respectively scale the spectral radius and singular values. It is clear that if we have a scaling factor greater than $|\lambda max|$, the network will not have echo states (zero being admissible input). The weight $\mathbf{W}$ should be scaled by a factor less than σmax only if zero input is used. For this research, the input is non zero, thus the scaling larger than 1 works best.

4 Global Control parameters

4.1 Size of the Reservoir

The size of the reservoir (here N) is an essential parameter of the model 1. The bigger the reservoir, the better is the obtainable performance. Appropriate regularization measures are taken against overfitting. The bigger the space of reservoir signals $\mathbf{x}(n)$, the easier it is to find a linear combination of the signals to approximate the target signal $\mathbf{d}(n)$ [11]. The size N of $\mathbf{W}$ should reflect not only length n_{max} of the training data but also the difficulty of the task. The value of N should not exceed an order of magnitude of $n_{max}/10$ to $n_{max}/2$. This is a simple precaution against overfitting [8]. If the task is more difficult, it requires a larger N. In our case, the size of the reservoir (N) is taken as 100 (as per personal communication with Prof. Herbert Jaeger). It can be also noted from [11] that the researchers often limit their reservoir sizes for convenience and compatibility of results.

4.2 Spectral Radius

The spectral radius of the reservoir weight matrix $\mathbf{W}$ codetermines (i) the effective time constant of the echo state network (larger spectral radius implies slower decay of impulse response) and (ii) the amount of nonlinear interaction of input components through time (larger spectral radius implies interactions). It is the maximal absolute eigenvalue of this matrix and it scales the width of the distribution of nonzero elements of $\mathbf{W}$ [7].
Typically a random sparse $\mathbf{W}$ (here sparse means to make most of elements in $\mathbf{W}$ equal to zero) is generated; its spectral radius $\rho(\mathbf{W})$ is computed; then $\mathbf{W}$ is divided by $\rho(\mathbf{W})$ to yield a matrix with a unit spectral radius; this is then conveniently scaled with the ultimate spectral radius to be determined in a tuning procedure [11].

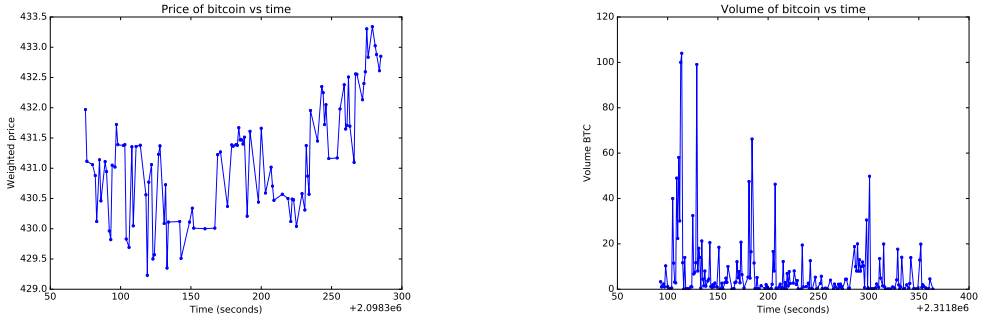
4.3 Input scaling

The various parameters were used to optimize the echo state networks: Wsf, $Winsf$ and bsf are the scaling factors used for the random input matrices (cross-validation was used as explained in 5.5). The input scaling codetermines the degree of nonlinearity of the reservoir dynamics. With very small effective input amplitudes the reservoir behaves almost like a linear medium. It is because the network will operate around the central part of the sigmoid. However, very large input amplitudes drive the neurons to the saturation of the sigmoid. This results in a binary switching dynamics [7]. Thus, the manual adjustment of the input scaling factor is required to find the scaling factor appropriate to the task [8].

5 Design Of Experiment

5.1 Dataset Description

The historical Bitcoin market data at 1-min intervals for select Bitcoin exchanges was obtained from the period of Jan 2012 to October 2017, with the minute to minute updates of OHLC (Open, High, Low, Close), Volume in BTC, indicated currency and weighted Bitcoin price. Timestamps are in Unix time. The prediction task is carried out on the weighted price of Bitcoin. The price of the Bitcoin is weighted as it is calculated by taking volume-weighted average of all the prices which are reported at each market (there are currently 400 markets of Bitcoin). There are a total of 3045858 data points in the weighted price data. 350522 out of the total data points were chosen in the prediction task for simplicity. They represent total data points from January 1st (12:01:00 AM), 2016 to December 31st (11:59:00 PM), 2016. The data obtained from [3] was contributed to https://www.kaggle.com by the user Zielak.
The main challenge in the prediction task was not with the prediction itself, but with the data preparation. As per the provider [3] and observation, there are certain anomalies in the data such as missing timestamps and jumps. These anomalies may be because of the exchange (or its API) being down or not existing at that time or from other technical errors in data reporting or gathering. Timestamps without any trades or activity have their data fields populated with duplicate data. Thus, this leads to ladder-shaped anomalies in the initial data. After removing such anomalies, unevenly spaced time series is obtained.



(a) Original weighted price **(b)** Original volume BTC

Figure 2: Raw input data points.

5.2 Feature Selection

Feature selection is an essential step in building an accurate predictor as it helps to select the attributes from/related to data that are most relevant for the prediction task [5]. Increasing the accuracy of the model is necessary. However, one might ponder about which features are essential for the model?. For this, different attributes of the signal must be tried out as an input to see which one works and which does not. In order to check if the feature is highly correlated to the main input signal, error is calculated. If the error increases on adding a feature, it is discarded. Also, adding unnecessary features would introduce noise in the model. For my research, I used standard feature selection techniques which are explained below.

5.2.1 Time differencing

Differencing is a data transformation technique to remove the trend from the data. *"The idea behind differencing is that, if the original data series does not have constant properties over time, then the change from one period to another might [4]."* That is why I used the first time differencing of the price signal as one of the features. For the data points (a, b, c, ...) new differenced points ((b-a), (c-b), ...) were calculated. After doing so, the trend in the data was gone. Later, on the reconstruction of the data (..., r, s) with predicted data points, ((t-s), (u-t), ...) with x = (t-s) and y = (u-t), the reconstructed points would be calculated as ((s+x), (s+x+y), ...) = (t, u, ..).

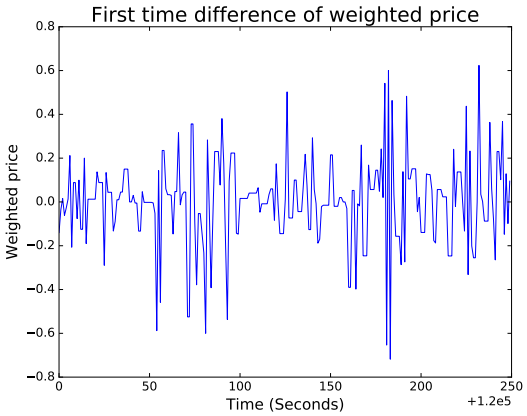


Figure 3: First-time difference of the price signal.

5.2.2 Fast Fourier Transform

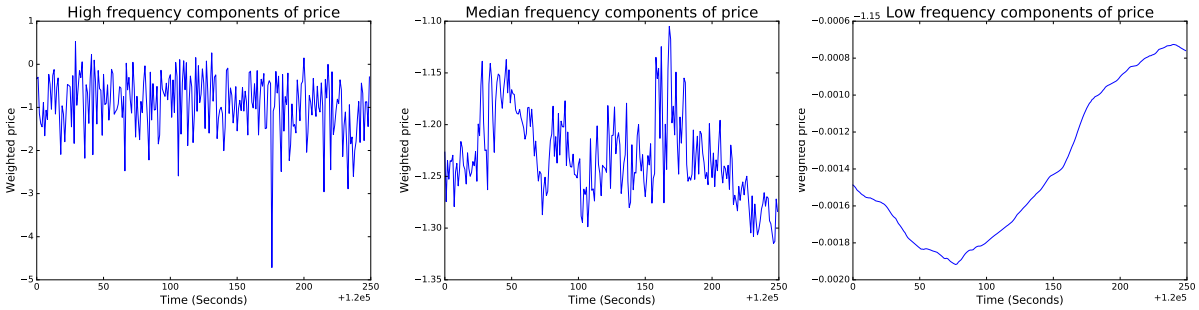
"A fast Fourier transform (FFT) is an algorithm that samples a signal over a period (or space) and divides it into its frequency components [24]." The FFT decomposes the signal into constituent functions of different frequencies. By taking a sampling window of 1024 (as it should be in the order of 2^n) and step size of 1, FFT is applied to the time series signal of the weighted price. Let $\mathbf{u}$(n) be the signal, then $\mathbf{u_{win}}$(1,...,win) is the input signal of window, win = 1024. Then, for each window, a maximum, a median and a minimum value are determined as

$$\mathbf{U_{max}} = max[\mathbf{u_{win}}] \tag{7}$$

$$\mathbf{U_{min}} = min[\mathbf{u_{win}}] \tag{8}$$

$$\mathbf{U_{med}} = (\frac{win+1}{2})^{th}. \tag{9}$$

Each of these represent a data point for a new signals (maximum, minimum and median). Then for the next data point, the window is slided by a single step. This is continued until the end of the signal. Finally, this gives three different features as shown in 4a, 4b and 4c denoting fast, medium and slow feature signals of the price signal.



(a) High-frequency signal **(b)** Median frequency signal **(c)** Low-frequency signal

Figure 4: High, median and low frequency signals.

5.2.3 Energy

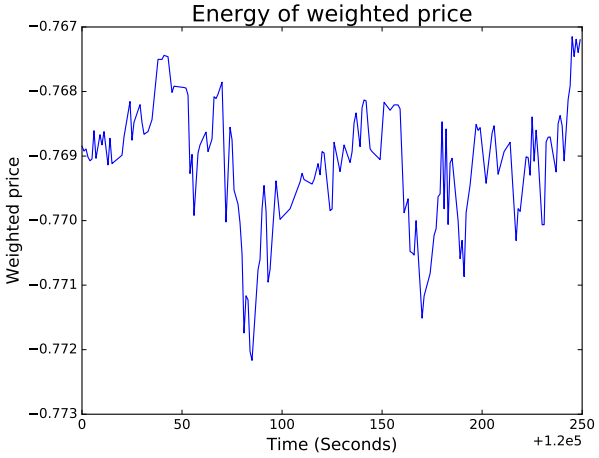


Figure 5: Energy of the Bitcoin price signal

The energy of the signal is calculated as

$$\mathbf{E} = \frac{1}{n_{max}} \sum_{i=1}^{n_{max}} \mathbf{u}(i)^2 \tag{10}$$

where n_{max} is the total data points, u(0,..., n_{max}) is the input data.

5.2.4 Volume BTC

The Volume of Bitcoin in BTC is the number of Bitcoins traded in the given period. Since the data set has the resolution of one second, the volume represents circulation of Bitcoins in that period. This is one of the indicators of the strength of the cryptocurrency. The volume BTC data points are used as a feature.

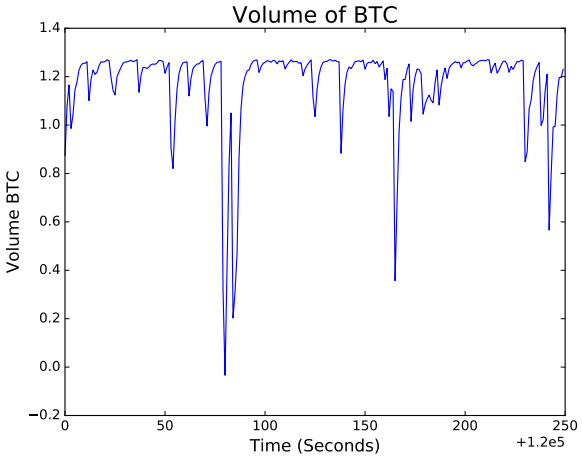


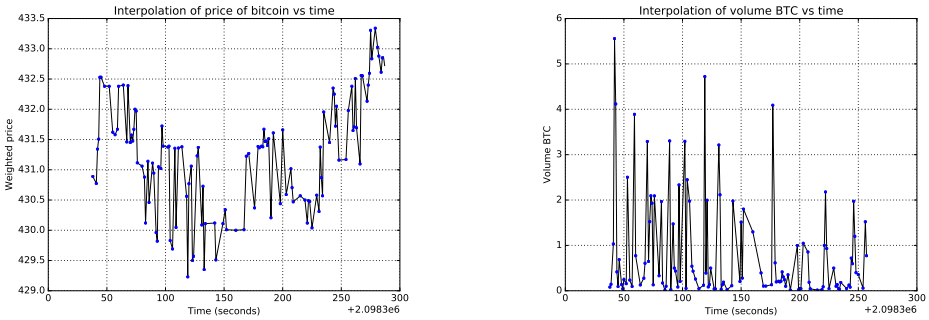
Figure 6: Volume BTC of bitcoin

5.3 Data Preprocessing

Data Preprocessing is an essential part of financial forecasting. It is not a good idea to feed a raw data into a model before preprocessing as it may lead to poor performance of the model. A raw data may be inconsistent or incomplete. So, to scale the data in standard setting or remove such anomalies, preprocessing is carried out.

5.3.1 Interpolation

The input for ESN should be evenly spaced in time. However, there are missing data points as explained in 5.1. Thus, in order to achieve evenly spaced time series data, linear interpolation is applied to the data points (both for weighted price and volume) as shown in 7a and 7b.



(a) Interpolated weighted price **(b)** Interpolated volume

Figure 7: Result of linear interpolation.

5.3.2 Exponential Smoothing

Exponential smoothing refers to using the exponentially weighted moving average (EWMA) to smooth data in signal processing [10]. It acts as low-pass filters to remove high-frequency noise in the data [23]. The initial raw data is represented as u_n at initial time $n = 0$. Then after applying the exponential smoothing algorithm, we will obtain output denoted by s_n. The exponential smoothing is given by

$$\begin{aligned} s_0 &= u_0 \\ s_n &= \alpha_s u_n + (1 - \alpha_s) s_{n-1},\ n > 0 \end{aligned} \tag{11}$$

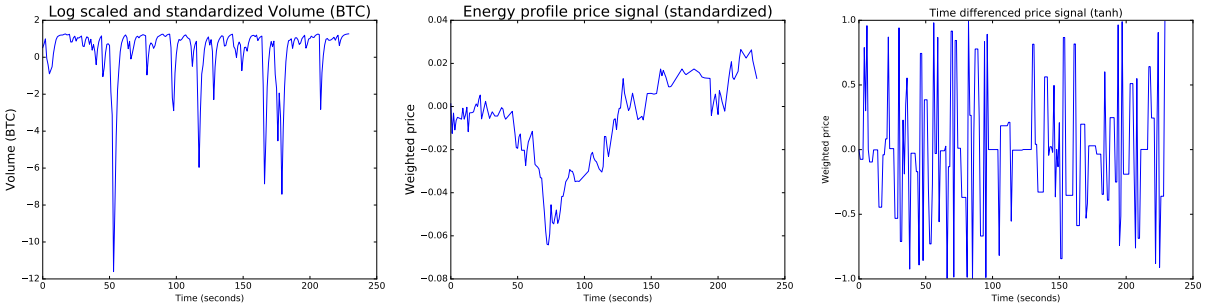
where α_s is the smoothing factor, and $0 < \alpha_s < 1$ [23].
The smoothing factor α_s is the speed at which the older responses are dampened (smoothed). The dampening is slow when α_s is near to 0, and dampening is quick when its value is near 1 [17]. The smoother is applied when preprocessing the volume BTC as well as generating the profile (after carefully tuning α_s).

5.3.3 Log scale and Standardization

Since, the input data points have peaks (which is not good for the ESN) log10 scale was applied to the input. After applying the log scale, the data was normalized to make it zero mean and unit variance. It was done by

$$u' = \frac{u - \bar{u}}{\sigma} \tag{12}$$

where u is the feature data points, and log scaled data, $\bar{u}$ is the mean of that data points, and σ is its standard deviation. This puts each learning task into a more standardized setting. The features have been standardized before being passed to the network as inputs.



(a) Volume signal. **(b)** Energy signal. **(c)** Time differenced signal.

Figure 8: Standardized features.

5.3.4 Profile of the price signal

The weighted price is chosen as the signal which the prediction model is to be applied. One year data of 2016 (350522) with one second time interval was selected as a raw input (refer to 2a). The raw data was interpolated to make it evenly spaced in time (as

shown in 7a), which resulted in 527038 data points. Finally, the data points were log scaled and standardized to build a profile of the price signal as shown in 9.

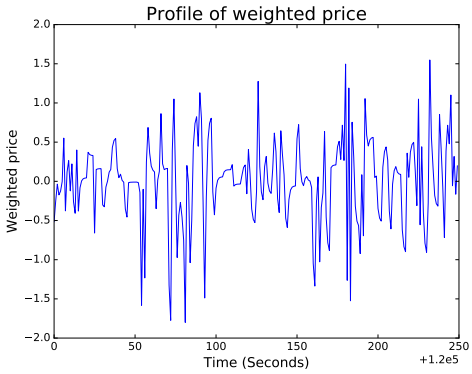


Figure 9: The preprocessed weighted price.

5.4 Network Setup

The Network is set up with 7 input units, 100 neurons, and 7 output units. The input is the result of the standardized data points, which is fed into the network containing 100 neurons. The random seed is fixed to eliminate the random fluctuation of performance. $\mathbf{W}^{in}$, $\mathbf{W}$ and $\mathbf{b}$ are initialized with randomly sampled values in the interval [-0.5, 0.5], so that mean value of the weights would be zero. As explained in [8], the bias ($\mathbf{b}$) is added such that it will *"immediately enable the training to set the trained output to the correct mean value."* The spectral radius (here $\rho(\mathbf{W})$) is calculated as

$$\rho(\mathbf{W}) = \ max(\ abs(\ eig(\ \mathbf{W}\ ))). \tag{13}$$

The reservoir weight is scaled by the inverse of $\rho(\mathbf{W})$ to achieve the spectral radius of 1. This would ensure the echo state property of the network. The weights are scaled respectively by input weight scaling factor ($Winsf$), reservoir weight scaling factor (Wsf) and bias scaling factor (bsf) [11].
After careful inspection of Graph 10, the washout of 10 training data points was determined, i.e., the point after which there is no residue of the initial state. Then, the signal is cut off from the initial to the washout to remove initial transient and noise. The plot of activation states was done.

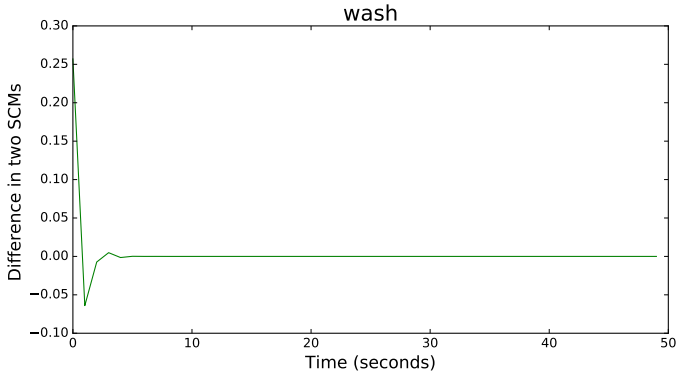


Figure 10: The difference between two SCMs, one initialized with zeros and another with ones.

5.5 Training Phase and K fold cross-validation

Out of the total 527038 data points obtained after the preprocessing step, 400000 data points are used in cross-validation step. Remaining 127038 data points is kept untouched for testing.

Figure 11: 10 fold cross-validation

In K-fold cross-validation, the full data set is split into K equal length partitions, where K-1 partitions are selected as the training set, and the remaining partition is selected as the validation set. In our case, K = 10, i.e 10-fold cross-validation scheme is applied. The model is trained under the training set and the teacher data set. The teacher data set is shifted by one from the training dataset. The result of training the data set in this 10-fold cross-validation is shown in figure 12 and 14.

Finally, after training, the trained model is used to predict time shifted signals on the validation set. In this case, $\mathbf{NRMSE}$ calculation is carried out for each fold, and the resultant mean $\mathbf{NRMSE}$ is calculated.

5.6 Optimizing parameters

The parameters such as scaling factors for the input weights, the regularization coefficient and the number of reservoir states are used as optimizing parameters to obtain lower $\mathbf{NRMSE}$, and higher accuracy in the model. The standard strategy to increase optimization is to use the larger number of the reservoir. However, this would be computationally expensive. In the implementation, 100 reservoir states are used. The parameters were tuned manually (rather than using grid search). Only one parameter was selected and changed until an optimal value was found before changing another one. And then, it was repeated for other parameters until the performance of the predictor was satisfactory.

5.7 Testing Phase

After data sets for the 7-dimensional input signals for the testing were obtained, the weight matrices from training, i.e., $\mathbf{W}^{in}$, $\mathbf{W}$ and $\mathbf{W}^{out}$ (optimized) were used in the state update equation (refer to 2) with the input data points to obtain Data Collection Matrix. Finally, the output $\mathbf{y}$ was calculated by using $\mathbf{W}^{out}$ and the desired output $\mathbf{d}$ (refer to 2). Then, the $\mathbf{NRMSE}$ was calculated to check how well the network has learned/performed.

5.8 Principle Component Analysis (PCA)

PCA is used as a dimension reduction method as well as a feature selection tool to obtain components according to decreasing order of their coefficients. PCA is applied in the input seven-dimensional feature matrix to get the k principal components. These components are ranked in order of decreasing importance as according to their explained variance where each variable contributes to each element with varying degree. Instead of using the original features, new features selected by choosing the principal components having the largest variance criteria are used as an input to the reservoir [20]. In my implementation, I would be comparing the results with and without using PCA.

5.9 Error Calculation

The error calculation is performed with respect to the given desired output weights $\mathbf{d}(n)$, where the output signal $\mathbf{y}(n)$ is compared to the original signal $\mathbf{d}(n)$, using a Normalized Root-Mean-Square-Error ($\mathbf{NRMSE}$) as

$$\mathbf{NRMSE} = \sqrt{\frac{\sum_{i=1}^{N}((\mathbf{y}(i) - \mathbf{d}(i))^2}{N\hat{\sigma^2}(d)}} \tag{14}$$

where $\mathbf{N}$ denotes the number of the output data points, $\hat{\sigma}^2$ denotes the variance of the points in $\mathbf{y}$ and $\mathbf{d}$ [25]. The training error calculation is essential to optimize the model. If the result of $\mathbf{NRMSE}$ in training is close to 1 then the signals are unrelated to each other (failure of training), while the value close to 0 means that training procedure worked well.

6 Results

6.1 Training Phase

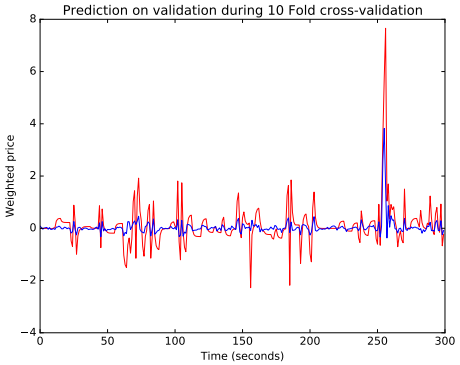


Figure 12: Validation fold and prediction of the 7th fold.

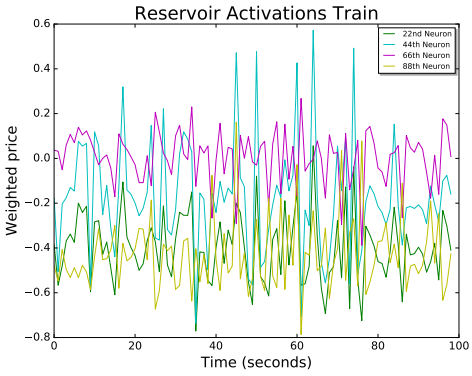


Figure 13: Some reservoir activations of the best fold.

Figure 12 shows the plot between the validation data and its prediction at fold seven where the $\mathbf{NRMSE}$ is 0.824. The range of the optimized parameters during cross-validation where it gave best results are (1) $Wsf \in [0.19, 0.23]$, (2) $Winsf \in [0.58, 0.65]$, (3) $bsf \in [0.28, 0.34]$ and (4) $\alpha \in [0.09, 0.011]$. The final cross-validation parameter values for the optimal result is shown in Table 1. During this phase, the error was computed on each validation set. The maximum error obtained was 0.920 while the minimum error was 0.824. This resulted in mean training $\mathbf{NRMSE}$ error of 0.873. Some reservoir activation (22nd, 44th, 66th and 88th) are shown in Figure 13, which reveals the states inside the reservoir.

Parameter	Value
Wsf	0.2
$Winsf$	0.6
bsf	0.3
α	0.01

Table 1: Optimal parameters for 10 fold cross-validation for seven features.

6.2 Testing Phase

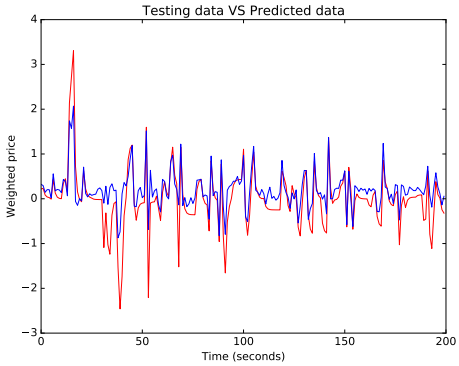


Figure 14: Result after testing the data.

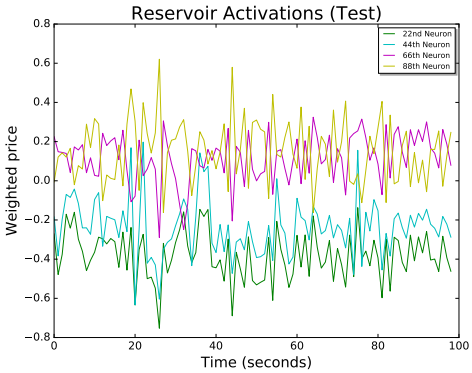
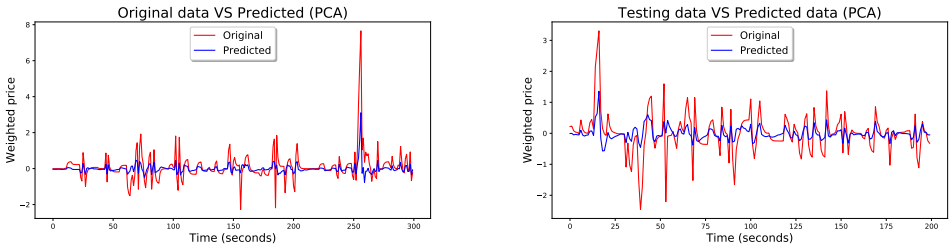


Figure 15: Some reservoir activation.

Figure 14 shows the plot between the test data (that was kept untouched) and its prediction. Out of seven output units, the output of the first feature which is the profile of the weighted price is only considered for the prediction task. The output weight of the minimum error was taken for final testing. The mean $\mathbf{NRMSE}$ of testing is found to be 0.758. Some reservoir activation as shown in Figure 15.

6.3 PCA

PCA was also applied to the feature matrix, and the explained variance for four principal components are 0.406, 0.293, 0.208 and 0.069. The mean $\mathbf{NRMSE}$ during training was 0.88 while the mean $\mathbf{NRMSE}$ during testing was 0.853. Figure 16a shows the plot between best fold validation and its prediction while Figure 16b shows the testing data and its prediction. Some activation during testing is shown in Figure 17.



(a) Validation data and its prediction.

(b) Testing data and its prediction.

Figure 16: Original data and its prediction.

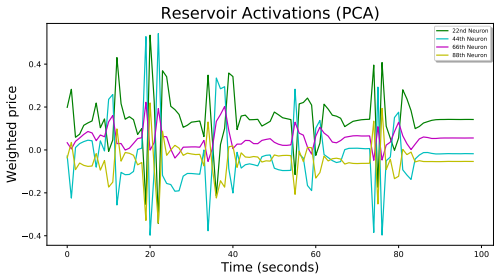


Figure 17: Some reservoir activation.

The optimal values obtained after running cross-validation on the four features obtained from PCA is given in Table 2.

Parameter	Value
Wsf	0.5
$Winsf$	0.64
bsf	0.4
α	0.014

Table 2: Optimal parameters for 10 fold cross-validation for four principle components.

7 Discussion

The primary goal of the project is not to compete with state of the art, but to evaluate how the ESN can perform in the task of prediction. Since the data points were not equally spaced in time, most of the time and effort was dedicated to the data preprocessing part. After studying different methods of prediction in 2.3, two of the methods [14] and [2], mainly focused on predicting the direction of the price change of the Bitcoin where the former approach considered only a single Bitcoin while the latter considered the price change at only at a single point of time. [16] had employed parallel computing method to reduce the complexity. In this implementation, the price of the Bitcoin at the next step is calculated where the prediction itself will give the direction of change of the price. The model was easily set-up and read outs from the output assisted in tuning the global control parameters. Even though the $\mathbf{NRMSE}$ error is high (0.758), several improvements could be done in order to increase the performance of the model. Adding multiple inputs and tuning parameters were easily done. This prediction procedure is less complicated than other counterparts, as it is easy to train ESNs and solutions are calculated in a single step.

7.1 Analysis of results

First of all, to get an idea of how good the prediction model performed, we need to compare it with a baseline model. For the baseline model, the prediction is just the replication of the data from the previous time step. By doing so, $\mathbf{NRMSE}$ of 1.032 was obtained. This is significantly larger than the error achieved in testing phase (0.758). This shows

that adding more features that are relevant (more to be discussed below) to the prediction task improved the performance of the model.
As compared to using only four features from the PCA, it was seen that the model performed slightly better when all seven features were used. There are various factors that we can attribute to this behaviour. The PCA considers measurements from all of the original variables in the projection and compresses them into a lower dimensional space. It only looks at the linear relationships and overlooks the possible multivariate nature of the data structure [20]. The activations of the seven feature procedure are more saturated than that of the activations of the PCA features as shown in Figures 17 and 15 respectively.
The performance of the ESN depends on how well the inputs are preprocessed. If the input signal contains peaks or sudden jumps, this would adversely affect the inner mechanism of the ESN. Thus, the prediction cannot follow the original data properly because there are peaks created by the outliers in the data. This is one of the setbacks encountered in the research. When the ESN trained with a signal in a specific range encounters the sudden peaks/spikes, then this would excite the neuron to the ceiling or floor of the sigmoid. This encounter would adversely affect the input history carried by the previous neurons (maybe results in losing the memory of the previous activation). The signal would instead act as a discrete signal with gaps in the places where it has the peaks, and hinder the performance of the model as a whole.
The preprocessing is essential as it scales down the outliers in the input signal and reveals the hidden features in signal. The linear interpolation used in the preprocessing step is not effective as it only connects the two unevenly placed data by a single line. Due to this, there is a loss of potentially important information between these data points.
It was observed that when the value of Wsf was high, there was fast oscillation of the neurons. When $Winsf$ was large, there was binary switching of the value of neuron which made the prediction deterministic. It is because of high influence of the input in the reservoir states. Thus, it is essential to plot the activation as it helps to visually inspect the dynamics in the reservoir.

7.2 Possible Features

The prediction of the price of Bitcoin depends on many aspects. For the research, seven features were used, however, several other factors need to be taken into consideration. Due to the lack of other potentially useful features such as the sentiments of the market, transaction to trade ratio, hash rate etc., better performance could not be achieved. This was mostly not possible as the data for the specific period with the specific resolution could not be achieved.
Feature analysis should be carried out to know which features are relevant to the prediction task. There is no easy way to learn if the feature is relevant or not, however, error calculation on adding such features would be useful. Having more features does not necessarily mean that it would increase the efficiency of the model. Selecting the features that contribute new information is essential, or else it would only add unnecessary noise to the network. The research falls short in analyzing and shortlisting possibly useful features from a wide selection of features for the main prediction task.

7.3 Future Work

The following improvements can be made in the existing research:

- There are three different features obtained after FFT (high, median and low signals). Three ESNs could be used where each network could be trained with high, median and low signals to obtain fast, medium and slow networks. Finally, the output of these networks could be combined to get a prediction.
- The interpolation scheme could be improved by using Markov's chain and subsampling scheme. However, special care should be taken while choosing an interpolation scheme. During the previous project, spline interpolation was used. However, ESN learnt to understand the interpolation scheme within some limits rather than Bitcoin dynamics. This would insert certain regularities into the data.
- To improve the accuracy of the prediction, more input units (features) can be added with special care in analyzing right features. The input can be a ramp signal that denotes the day of the week when the price of Bitcoin rises (on weekends).
- The training procedure can be refined to use raw data points.
- Outlier detection can also be integrated to get better results.

Finally, the reconstruction of the data should also be done. There is wide scope for improvement where the quality and accuracy of the prediction is concerned [2].

8 Conclusion

To make predictions about the financial market is quite a challenge. The data itself makes the prediction task tricky. Although there is an abundance of information in general, it is challenging to find data based on which one can conclude. Lack of useful data is very problematic, however, if we have a significant amount of data, it would, in turn, increase the computational complexity. Thus, it was expected that the results would reflect these shortcomings, especially when the data in question came with its irregularities. The volatility of cryptocurrencies is another factor that obstructed the accuracy. However, the need for the prediction methods related to Bitcoins and cryptocurrencies, in general, need to be taken into serious consideration. With the type of storm that cryptocurrencies are creating in the financial market, one must be equipped with techniques to remain informed about the type of changes that these virtual currencies can bring about in the economy and consequently, in one's lives.

References

[1] Amilde, R. "A Novel Method for Training an Echo State Network with Feedback-Error Learning." *Advances in Artificial Intelligence* (2013), p. 9. URL: http://dx.doi.org/10.1155/2013/891501.

[2] Amjad, M. and Shah, D. "Trading Bitcoin and Online Time Series Prediction." In: *Proceedings of the Time Series Workshop at NIPS 2016.* Ed. by Oren Anava et al. Vol. 55. Proceedings of Machine Learning Research. Barcelona, Spain: PMLR, 2017, pp. 1–15. URL: http://proceedings.mlr.press/v55/amjad16.html.

[3] *Bitcoin Historical Data.* Kaggle data set, [Online; accessed December 6, 2017]. URL: https://www.kaggle.com/mczielinski/bitcoin-historical-data.

[4] Boateng, N. *Time Series Analysis Methods*. [Online; accessed April 6, 2018]. URL: https://bit.ly/2rsGSzs.

[5] Jason Brownlee. *An Introduction to Feature Selection.* [Online; accessed May 1, 2018]. 2014. URL: https://machinelearningmastery.com/an-introduction-to-feature-selection/.

[6] G.H. Chen, S Nikolov, and D Shah. "A latent source model for nonparametric time series classification" (Jan. 2013).

[7] Jaeger, H. "Echo state network." *Scholarpedia* 2.9 (2007). revision #183563, p. 2330. DOI: 10.4249/scholarpedia.2330. URL: http://www.scholarpedia.org/article/Echo_state_network.

[8] Jaeger, H. *A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach.* Tech. rep. GMD Report 152. German National Research Center for Information Technology, 2002, p. 48.

[9] Jaeger, H. *Short term memory in echo state networks.* Tech. rep. GMD Report 152. German National Research Center for Information Technology, 2001, p. 60.

[10] Jinka, P. *Exponential Smoothing for Time Series Forecasting.* [Online; accessed May 4, 2018]. 2017. URL: https://www.vividcortex.com/blog/exponential-smoothing-for-time-series-forecasting.

[11] Lukoševičius, M. "A Practical Guide to Applying Echo State Networks." In: *Neural Networks: Tricks of the Trade*. Ed. by Orr G. Montavon G. and K Müller. 2012, pp. 659 –686.

[12] Lukoševičius, M., and Jaeger, H. "Reservoir computing approaches to recurrent neural network training." *Computer Science Review* 3.3 (2013), pp. 127 –149. ISSN: 1574-0137. DOI: {https://doi.org/10.1016/j.cosrev.2009.03.005}. URL: {http://www.sciencedirect.com/science/article/pii/S1574013709000173}.

[13] Maass, W., Natschlaeger, T., and Markram, H. "Real-time computing without stable states: A new framework for neural computation based on perturbations." *Neural Computation* 14(11) (2002). Online, pp. 2531–2560. URL: https://www.ncbi.nlm.nih.gov/pubmed/12433288.

[14] Madan, I. and Saluja, S. *Automated Bitcoin Trading via Machine Learning Algorithms.* Tech. rep. Stanford University, 2014, p. 5.

[15] Schiller, U. and Steil, J. “Analyzing the weight dynamics of recurrent learning algorithms.” *Neurocomputing* 63.Supplement C (2005). New Aspects in Neurocomputing: 11th European Symposium on Artificial Neural Networks, pp. 5 –23. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2004.04.006. URL: http://www.sciencedirect.com/science/article/pii/S0925231204003145.

[16] Shah, D. and Zhang, K. “Bayesian regression and Bitcoin.” *CoRR* abs/1410.1231 (2014). arXiv: 1410.1231. URL: http://arxiv.org/abs/1410.1231.

[17] *Single Exponential Smoothing.* NIST/SEMATECH e-Handbook of Statistical Methods, [Online; accessed April 28, 2018]. 2013. URL: https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc431.htm.

[18] Stenqvist, E. and Lönnö, Z. *Predicting Bitcoin price fluctuation with Twitter sentiment analysis.* Tech. rep. Online. 2017. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-209191.

[19] Turpin, J. “Bitcoin: The Economic Case for a Global, Virtual Currency Operating in an Unexplored Legal Framework.” *Indiana Journal of Global Legal Studies* 21.9 (2014). ISSN: 1. DOI: 10.2979/indjglolegstu.21.1.335. URL: https://www.repository.law.indiana.edu/ijgls/vol21/iss1/13.

[20] *Using principal component analysis (PCA) for feature selection.* Cross Validated, user: chl, [Online; accessed May 9, 2017]. 2017. URL: https://bit.ly/2HZW29w.

[21] *What is Bitcoin?* We Launch ICO, [Online; accessed December 15, 2017]. 2015. URL: https://www.welaunchico.com/HTML/bitcoin.html.

[22] Wikipedia contributors. *Bitcoin — Wikipedia, The Free Encyclopedia.* [Online; accessed 9-May-2018]. 2018. URL: https://en.wikipedia.org/w/index.php?title=Bitcoin&oldid=840375341.

[23] Wikipedia contributors. *Exponential smoothing — Wikipedia, The Free Encyclopedia.* [Online; accessed May 4, 2018]. 2018. URL: https://en.wikipedia.org/w/index.php?title=Exponential_smoothing&oldid=839619599.

[24] Wikipedia contributors. *Fast Fourier transform — Wikipedia, The Free Encyclopedia.* [Online; accessed May 1,2018]. 2018. URL: https://en.wikipedia.org/w/index.php?title=Fast_Fourier_transform&oldid=838005116.

[25] Wikipedia contributors. *Feature_scaling — Wikipedia, The Free Encyclopedia.* [Online; accessed December 15, 2017]. 2004. URL: https://en.wikipedia.org/wiki/Feature_scaling.