Application of Machine Learning and Deep Learning on Identifying Bubbles

FRE-GY 7773 – Machine Learning in Financial Engineering

Anthony Ning, Xiaohan Yue, Zhiyuan Wei

December 23, 2022

ABSTRACT

Cryptocurrencies are getting popular and the trading amount is rising in the last 10 years. Cryptocurrencies have now become an important composition in the market, and they are likely to become a new trend of trading currency and asset in the future. However, cryptocurrencies are notorious for their volatility, and there were bubbles in the past few years. Thus, we are interested in studying the bubbles in the cryptocurrency's market.

We decided to use LPPLS (Log Periodic Power Law Singularity) model, SVM (Support Vector Machine) model and LSTM (Long Short-Term Memory) model to predict the historical bubbles. Lastly, we will compare and evaluate the performance of each model.

With the paper, we try to apply different models to predict bubbles. We hope to make contribution to the field with different approaches, and further provide inspiration and novelty to market participants and future researchers.

Keywords: Cryptocurrency, Bubble, LPPLS, SVM, LSTM

Contents

List of Figures

List of Tables

1 Introduction

As methods of making payments evolve, the era of cash has now gradually evolved into the era of digital currencies and even cryptocurrencies. The rise of cryptocurrencies has now become a vital composition of the market. However, the cryptocurrency market is infamous for its volatility, and one can experience a huge loss due to it. Such a volatile market further intrigues us on the topic of bubbles. In fact, some bubbles happened in the cryptocurrency market in the history. In this project, we decided to study the cryptocurrency market's bubbles, and we will incorporate machine learning techniques and other approaches to try to detect and identify bubbles.

'Bubbles' are normally defined as a situation where the asset price exceeds its fundamental value[1]. The topic of detecting anomalies between market prices and real values and further identifying bubbles has reached interest to many recently. As a matter of fact, it has been a challenge for one to indicate or calculate the fair value of cryptocurrencies. Thus, in the project, we will first identify bubble periods based on the events and dates recorded on articles, news and relative websites.

In this project, we will focus on Bitcoin and choose it to represent the cryptocurrency market because of its large market capitalization and trading volume. Next, we will be using statistical, machine learning and deep learning models to try to identify and predict bubbles. Last but not least, we will compare the results of each model, evaluate and make a conclusion.

The fact that bubbles had indeed happened makes many people study in the area of bubbles. Although there are papers and articles contributing to the development of bubble-detection, this project tries to implement different models to predict bubbles and make a comparison. Not only we want to use different approaches to detect bubbles in the cryptocurrency market, but we also want to make further suggestion and contribution to the field of bubble-detection. With this project, we hope that we can bring an out-of-scope perspective to the topic of bubble-detection and to the subject of the cryptocurrency market.

Due to cryptocurrency's futuristic properties and extreme price volatility, the ability to predict bubbles has always been vital not only for investors to understand the cryptocurrencies' price dynamics, but also for policymakers and financial regulators who are responsible for wellbeing of the financial system. This paper also gives researchers who are interested in cryptocurrency a novel way to leverage new algorithms to understand the bubbles in cryptocurrencies.

[1] https://www.tandfonline.com/doi/full/10.1080/02664763.2020.1823947 [Fatma and Kamil, 2021]

2 Data and Methodology

2.1 Data

2.1.1 Cryptocurrency data

First of all, for data selection, 'Bitcoin' has a significant trading volume in the cryptocurrency market, so we choose it to represent the whole cryptocurrency market. We collect the daily price of Bitcoin from Bloomberg, ranging from 07/19/2010 to 12/18/2022, which has 3,804 records of data in total.

As for the definition of bubbles, we choose Wikipedia[2] and relative news and articles as our basis. Based on the sources and the dates recorded on them, we define periods of historical bubbles. You can refer to the following graph for historical bubble periods in the cryptocurrency market.

Figure 1: Historical bubble periods

As you can see, there are four bubbles in the history. Based on this information, we also produce a series of binary bubble data indicating whether there is bubble on each day, which will be used in some of our models.

2.1.2 Macroeconomic data

In SVM and LPPLS model, we include macroeconomic-variables and financial variable as features.

[2] https://en.wikipedia.org/wiki/Cryptocurrency_bubble [Cryptocurrency bubble, Wikipedia]

The macroeconomic-variables include GDP growth rate, short-term interest rate (IRS), long-term interest rate (IRL), unemployment rate (UNEMP), inflation rate (INF) and balance of payments (BLNC).

GDP growth rate is a measure of economic health of a country. The higher the GDP growth rate, the healthier the economy. We hypothesize that higher GDP growth rate makes people optimistic to future and therefore easily cause cryptocurrency bubbles. We calculate GDP growth rate as the logarithm of the ratio of present GDP to GDP a quarter ago.

Short-term and long-term interest rates are also measures of economic health and can affect cryptocurrency bubbles. The short-term interest rate is 3-month treasury bill interest rate. The long-term interest rate is 10-year treasury bill interest rate. We calculate the 10-year treasury bill interest rate by adding the 3-month treasury bill interest rate and the difference between the 10-year treasury bill interest rate and 3-month treasury bill interest rate.

Unemployment rate indicates health of the economy as well. It is calculated as the percentage of the unemployed in total labor force.

Inflation rate should also affect the price of cryptocurrency because intuitively, if cryptocurrency doesn't follow inflation, then there will be arbitrage opportunities. We use growth rate of CPI as the inflation rate.

Balance of payments measures the transactions between home country and other parts of the world. It also implies economic prosperity. It is calculated as current account surplus or value of export minus value of import.

The macroeconomic data are obtained from the Federal Reserve Bank of St. Louis.

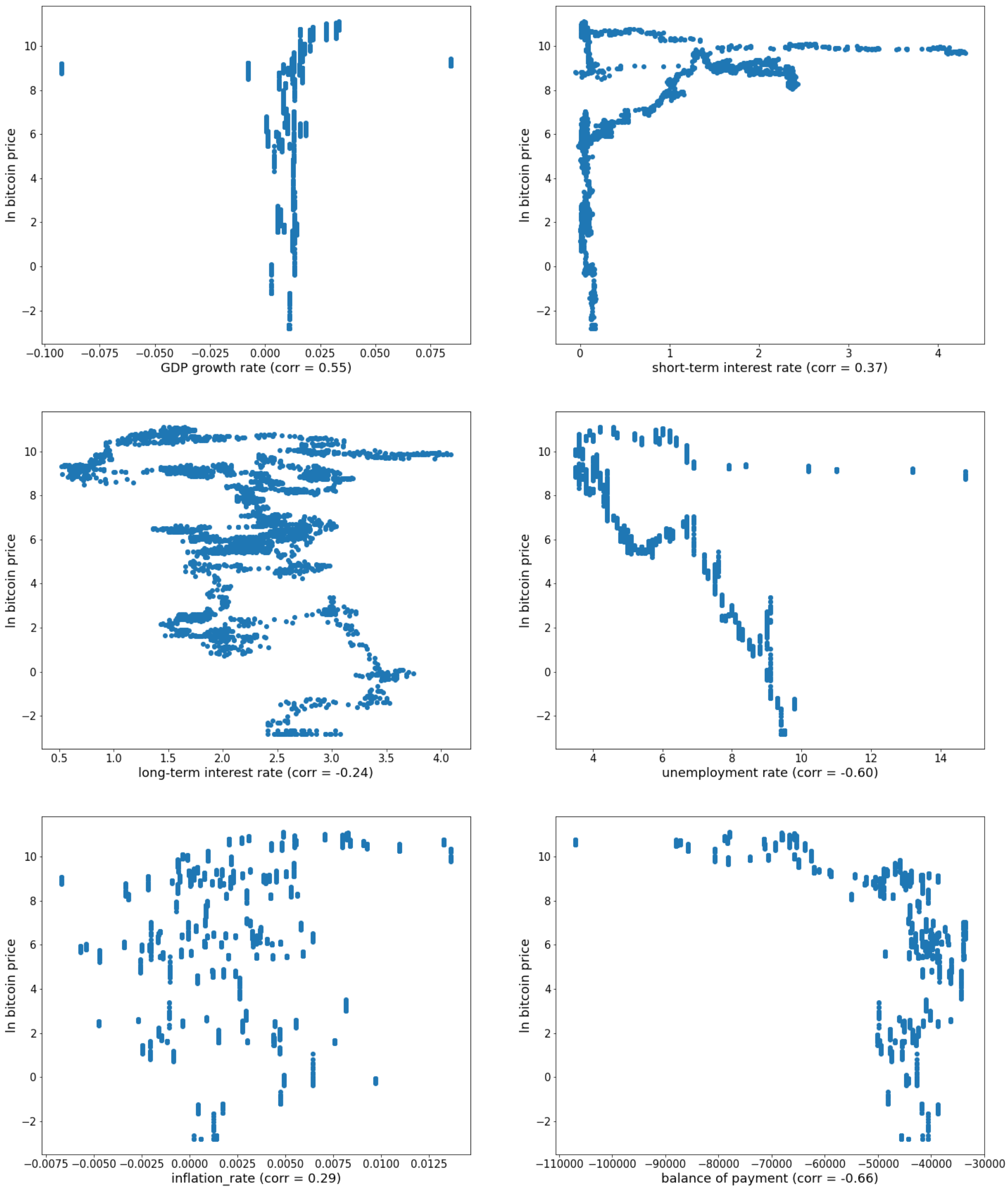

Figure 2: correlation plots and coefficients between macroeconomic variables and Bitcoin price

We calculate the Spearman correlation coefficients of the macroeconomic factors and plot the correlation graphs. The reason why we use Spearman correlation coefficient is that it can better capture nonlinear relationships.

1. The correlation coefficient between GDP growth rate and Bitcoin price is 0.55. This is

in line with our hypothesis.

2. Correlation coefficient of short-term interest rate and bitcoin price is also positive (0.37). This shows that higher short-term interest rate means people being more optimistic about future and in turn drives Bitcoin price.

3. However, the figure for long-term interest rate is negative. This may be because people view 10-year treasury bill as a way of investment and invest more in it when interest rate is high, in turn decrease investment activities in bitcoin.

4. Unemployment rate fits our hypothesis that higher unemployment rate means less active economy and so people don't invest in bitcoin and drive down Bitcoin price.

5. Inflation rate is positively correlated with Bitcoin price with correlation coefficient 0.29 and complies with our anticipation.

6. Balance of payment is negatively correlated with Bitcoin price. This may be because more trading activities (more current account deficit or higher absolute value of current account surplus) mean a more active economy and so drives up Bitcoin price.

2.1.3 Financial data

The financial variable we select is S&P 500 index. It can represent level of the financial market prosperity. The data is obtained from Bloomberg.

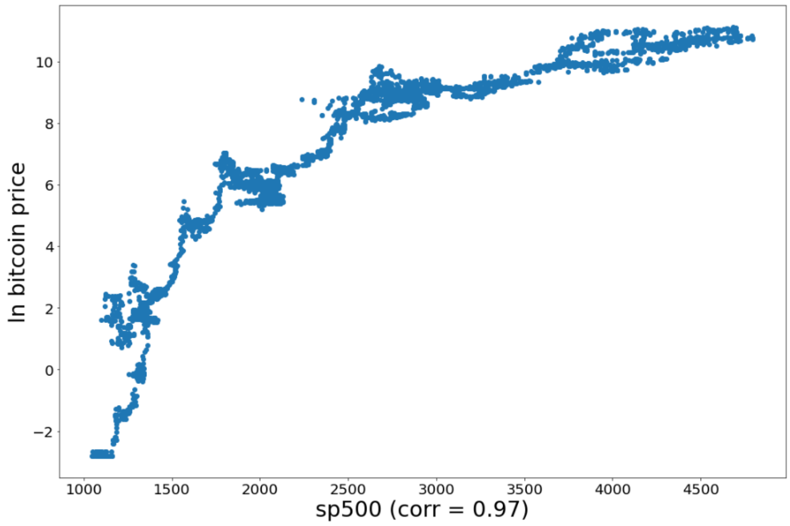

Figure 3: correlation between SP500 and Bitcoin price

S&P 500 is highly and positively correlated with Bitcoin price (0.97). Then to avoid spurious correlation, we calculate the correlation coefficient of the return of the two and find it's 0.11, showing that there is still positive correlation. Return is calculated as the logarithm of the ratio of present value to the most recent previous value.

2.1.4 Bitcoin return data

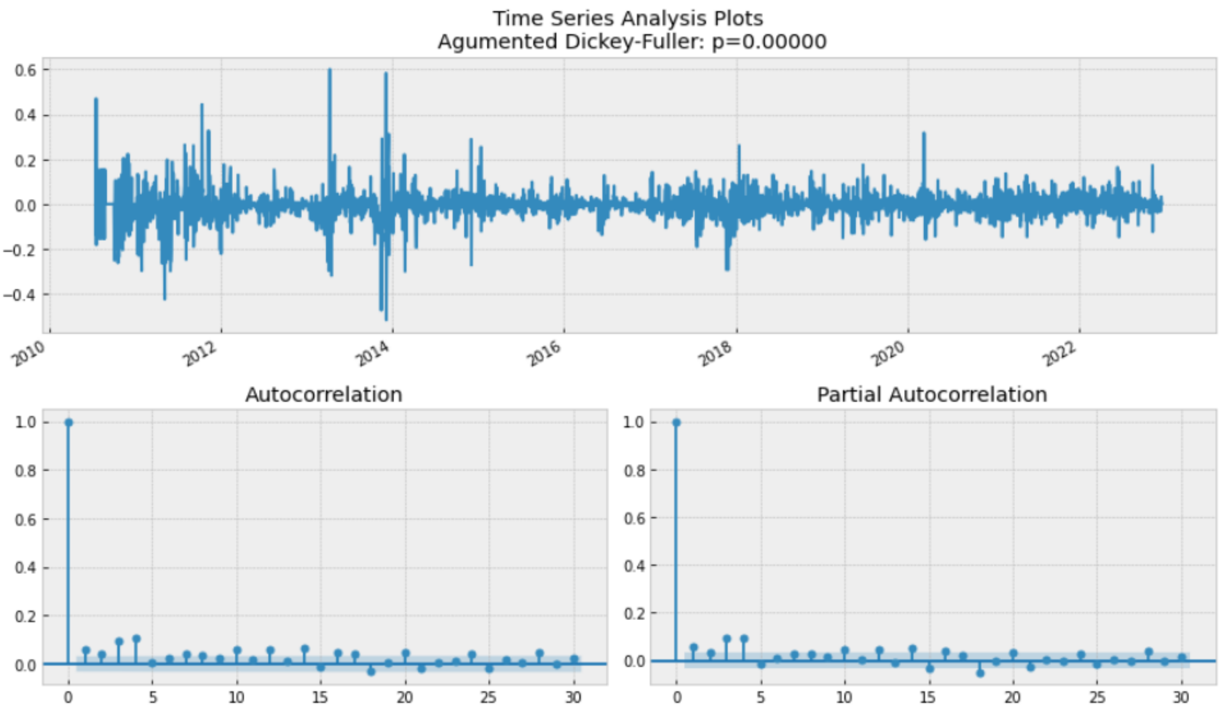

Figure 4: ACF and PACF plot of Bitcoin price return

We also calculate the return of Bitcoin price as logarithm of the ratio of present Bitcoin price to Bitcoin price on the previous day. the plot the autocorrelation and partial autocorrelation of Bitcoin price return (calculated as logarithm of the ratio of Bitcoin price and Bitcoin price on the previous day). We see that recent Bitcoin price return significantly affects current price return. In order not to lose generality, we will include the past 5 business day's return in the features.

2.2 Methodology

In this study, we will implement LPPLS (Log Periodic Power Law Singularity) model, SVM (Support Vector Machine) model and LSTM (Long Short-Term Memory) model to predict bubbles in the Bitcoin market and analyze the results. LPPLS is a statistical model which only uses Bitcoin price data. SVM and LSTM use macroeconomic and financial variables as features and binary bubble data as target.

2.2.1 LPPLS (Log Periodic Power Law Singularity) Model

a. LPPLS Model and Optimization

The LPPLS model combines two bubble characteristics, which are the transient super-exponential growth and the accelerating log-periodic volatility fluctuations. Under the condition that no crash has yet occurred, it leads to the simple mathematical formulation of the LPPLS for the expected value of a log-price[3]:

$$LPPLS\left(t\right)\equiv E\left[\ln p\left(t\right)\right]=A+B(t_c-t)^m+C(t_c-t)^m cos[\omega ln(t_c-t)-\varphi]$$

We have a linear part and cosines part. In the formula, A represents the expected value of the underlying asset at the critical time t_c, which represents the theoretical termination time of a financial bubble. If B is smaller than zero, it ensures that the price is growing super-exponentially as time approaches to t_c, and on the other hand, if B is bigger than zero, it ensures that the price is decreasing super-exponentially as time approaches to t_c. When the 'm' term is bigger than zero, it ensures that the price remains finite at t_c, whereas the expected log price diverges when the 'm' term is smaller than one. As for the cosines part, logarithm under cosines helps to model variability. ω helps to model the number of waves, and C models their sizes.

Several ways have been used to improve the LPPLS model's performance and accuracy. The reformed LPPLS formula of the model that we are going to use here is presented as follows, which we substitute C and ϕ parameters of the cosines part from the original formula with $C_1\cos\phi$ and $C_2\sin\phi$.

$$\begin{aligned}LPPLS\left(t\right)&\equiv E\left[\ln p\left(t\right)\right]\\&=A+B\left(t_c-t\right)^m+C_1\left(t_c-t\right)^m\cos\left[\omega\ln\left(t_c-t\right)\right]+C_2\left(t_c-t\right)^m\sin\left[\omega\ln\left(t_c-t\right)\right]\end{aligned}$$

The LPPLS model uses L2 norm of cost function to fit the data, which is presented as the following:

$$F\left(t_c,m,\omega,A,B,C_1,C_2\right)=$$
$$\sum_{i=1}^{N}[lnp(t_i)-A-B(t_c-t)^m-C_1(t_c-t)^m cos[\omega ln(t_c-t)]-C_2(t_c-t)^m sin[\omega ln(t_c-t)]]^2$$

[3] https://arxiv.org/abs/1905.09647 [Shu and Zhu, 2020]

Setting apart the 4 linear parameters [A, B, C_1, C_2], the remaining 3 nonlinear parameters t_c, m, ω yields the following cost function:

$$\begin{aligned} F_1\left(t_c,\ m, \omega\right) &= min_{[A,B,C1,C2]} F(t_c, m, \omega, A, B, C_1, C_2) \\ &= F(t_c, m, \omega, \hat{A}, \hat{B}, \hat{C}_1, \hat{C}_2) \end{aligned}$$

The estimated linear parameters, $\widehat{A}, \hat{B}, \widehat{C_1}, \widehat{C_2}$, can be obtained by solving the following optimization problem:

$$\left[\widehat{A}, \hat{B}, \widehat{C_1}, \widehat{C_2}\right] = arg\ min_{[A,B,C1,C2]} F(t_c, m, \omega, A, B, C_1, C_2)$$

As for the non-linear parameters, $[t_c, m, \omega]$, they can be obtained by solving the following optimization problem:

$$[t_c, m, \omega] = arg\ min_{[t_c,m,\omega]} F_1(t_c, m, \omega)$$

Lastly, the LPPLS model uses the Ordinary Least Squares method to optimize and estimate the linear and non-linear parameters.

b. LPPLS Confidence Indicator

It is used to measure the confidence levels of positive bubbles and negative bubbles happening. Positive bubbles refer to an accelerating ascending price trend while negative bubbles mean an accelerating descending price trend.

There are five steps to determine confidence indicators. First, LPPLS model will shrink the window from the start time t_1 to the 'fictitious' present t_2 with specified time steps we determine, and this will create a group of time series of price. Second, we determine the search space of fitting parameters. Third, we calibrate the model for each fitting window. Four, we filtrate the calibration results. Last, we calculate the LPPLS confidence indicator from dividing the number of time windows satisfying the filter conditions by the total number of the fitting windows.

Here, by setting different thresholds, we can further filter the confidence indicators and reduce the frequency of noises.

2.2.2 SVM (Support Vector Machine) Model

SVM is an unsupervised machine learning method, commonly used in binary classification problems. Compared with logistic regression, it introduces the concept of kernel function, which has a better classification effect on non-linear relations; at the same time, due to the introduction of the dual problem, it makes the complexity of calculation change from the size of dimension to the number of samples, avoiding dimensional explosion.

In SVM, we are given a sample set with m samples and n features: $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_m, y_m)\}, y_i \in \{+1, -1\}$ where $\mathbf{x}_i$ is the n-dimensional feature for sample i and y_i is the target for sample i. We plot the samples in the n-dimensional space and are asked to find a hyperplane to classify the samples and maximize the margin.

In the n-dimentional space, hyperplane can be represented as $\boldsymbol{w}^T\boldsymbol{x} + b = 0$ where $\boldsymbol{w} = (w_1; w_2; ...; w_n)$ is the normal vector and b is the displacement item. The Euclid distance from point $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ to hyperplane $\boldsymbol{w}^T\boldsymbol{x}+b = 0$ is $\frac{|\boldsymbol{w}^T\boldsymbol{x}+b|}{||\boldsymbol{w}||}$ where $||\boldsymbol{w}|| = \sqrt{w_1^2 + w_2^2 + ... + w_n^2}$.

Then we can set up the problem we want to solve:

$$\begin{gathered}\max_{\boldsymbol{w},b} \min_{\boldsymbol{x}_i, i=1,2,...,m} \frac{2|\boldsymbol{w}^T\boldsymbol{x}_i + b|}{||\boldsymbol{w}||} \\ s.t. \boldsymbol{w}^T\boldsymbol{x}_i + b > 0, y_i > 0 \\ \boldsymbol{w}^T\boldsymbol{x}_i + b < 0, y_i < 0\end{gathered}$$

where $\boldsymbol{x}_i = (x_{1i}; x_{2i}; ...; x_{ni})$ which is the vector of features of the ith sample and n is the number of features. $\boldsymbol{w} = (w_1; w_2; ...; w_n)$ is the normal vector we need to optimize and b is the displacement item.

This problem is equivalent to

$$\begin{gathered}\max_{\boldsymbol{w},b} \min_{\boldsymbol{x}_i, i=1,2,...,m} \frac{2y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)}{||\boldsymbol{w}||} \\ s.t. y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) > 0\end{gathered}$$

We further write it as:

$$\begin{gathered}\max_{\boldsymbol{w},b} \frac{2}{||\boldsymbol{w}||} \min_{\boldsymbol{x}_i, i=1,2,...,m} y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \\ s.t. \exists r > 0, \min_{\boldsymbol{x}_i, y_i, i=1,2,...,m} y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) = r\end{gathered}$$

By scaling w and b we get:

$$\max_{\boldsymbol{w},b} \frac{2}{||\boldsymbol{w}||}$$
$$s.t. \min_{\boldsymbol{x}_i, y_i, i=1,2,...,m} y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) = 1$$

By changing the maximization problem to minimization problem and changing the form of the constraint we get:

$$\min_{\boldsymbol{w},b} \frac{||\boldsymbol{w}||^2}{2}$$
$$s.t. y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$

We include Lagrangian multiplier $\alpha_i \geq 0$ and the Lagrangian function can be written as

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{||\boldsymbol{w}||^2}{2} + \sum_{i=1}^{m} \alpha_i(1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)),$$

where $\boldsymbol{\alpha} = (\alpha_1; \alpha_2; ...; \alpha_m)$ and $\alpha_i \geq 0$,

We have

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i = 0,$$
$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{m} \alpha_i y_i = 0,$$

which is

$$\boldsymbol{w}^* = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i$$
$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

Put the above two results to the Lagrangian function we get:

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}(\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i)^T (\sum_{j=1}^{m} \alpha_j y_j \boldsymbol{x}_j) - \sum_{i=1}^{m} \alpha_i y_i (\sum_{j=1}^{m} \alpha_j y_j \boldsymbol{x}_j)^T \boldsymbol{x}_i + \sum_{i=1}^{m} \alpha_i \\ &= \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \boldsymbol{x}_i^T \alpha_j y_j \boldsymbol{x}_j - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \boldsymbol{x}_j^T \boldsymbol{x}_i + \sum_{i=1}^{m} \alpha_i \\ &= -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j + \sum_{i=1}^{m} \alpha_i \end{aligned}$$

Therefore we get the dual problem:

$$\begin{gathered} \max_{\boldsymbol{\alpha}} -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j + \sum_{i=1}^{m} \alpha_i, \\ s.t. \sum_{i=1}^{m} \alpha_i y_i = 0 \\ \alpha_i \geq 0, i = 1, 2, ..., m \end{gathered}$$

Then we can use KKT condition to solve the α_i's and by using $\boldsymbol{w}^* = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i$ we can get $\boldsymbol{w}^*$:

$$\begin{cases} \alpha_i \geq 0; \\ 1 - y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \leq 0; \\ \alpha_i(1 - y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b)) = 0 \end{cases} \tag{1}$$

For any sample (x_i, y_i), we have either $\alpha_i = 0$ or $1 - y_i(w^T x_i + b) = 0$. If $\alpha_i = 0$, then the sample will not show up in the dual problem and will not affect $f(x)$ (it is not a supporting vector). If $\alpha_i > 0$, then we have $y_i f(x_i) = 1$, then the sample point is on the boundary (it is a supporting vector).

Then we know

$$
\begin{aligned}
\exists(\boldsymbol{x}_k, y_k) s.t. 1 - y_k(\boldsymbol{w}^T\boldsymbol{x}_k + b) &= 0 \\
y_k(\boldsymbol{w}^T\boldsymbol{x}_k + b) &= 1 \\
y_k^2(\boldsymbol{w}^T\boldsymbol{x}_k + b) &= y_k \\
\boldsymbol{w}^T\boldsymbol{x}_k + b &= y_k \\
b^* &= y_k - \boldsymbol{w}^T\boldsymbol{x}_k \\
b^* &= y_k - \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i^T \boldsymbol{x}_k \\
b^* &= y_k - \sum_{i \in S} \alpha_i y_i \boldsymbol{x}_i^T \boldsymbol{x}_k
\end{aligned} \tag{2}
$$

where $S = \{i | \alpha_i > 0, i = 1, 2, ..., m\}$, i.e., the set of subscripts of supporting vectors. Therefore b^* can also be solved.

Till now we introduced hard margin SVM, we assume that the sample points are linearly separable. However, in reality, there may not be a hyperplane that can correctly classify the original sample points.

To solve this problem, we can map the original space to a higher dimensional space to make the sample linearly separable in the space. One can prove that if the original space has finite dimensions, then there must exists a higher dimensional space to make the sample linearly separable.

We use $\phi(\boldsymbol{x})$ to represent the eigenvector after mapping $\boldsymbol{x}$. Then the hyperplane in the new space can be represented as:

$$
f(x) = \boldsymbol{w}^T\phi(\boldsymbol{x}) + b \tag{3}
$$

Similar to the situation in hard margin SVM, we have Lagrangian problem:

$$
\begin{gathered}
\max_{\boldsymbol{w}, b} \frac{||\boldsymbol{w}||^2}{2} \\
s.t. y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b) \geq 1
\end{gathered}
$$

And the dual problem:

$$\max_{\boldsymbol{w},b} -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j) + \sum_{i=1}^{m}\alpha_i,$$

$$s.t. \sum_{i=1}^{m}\alpha_i y_i = 0$$

$$\alpha_i \geq 0, i = 1, 2, ..., m$$

After solving the problem we get:

$$\begin{aligned} f(\boldsymbol{x}) &= \boldsymbol{w}^{*T}\phi(\boldsymbol{x}) + b \\ &= \sum_{i=1}^{m}\alpha_i y_i \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}) + b \\ &= \sum_{i=1}^{m}\alpha_i y_i \kappa(\boldsymbol{x}, \boldsymbol{x}_i) + b \end{aligned} \tag{4}$$

$\kappa(.,.)$ is the kernel function.

There are many kinds of Kernel functions to choose, like linear, poly, RBF, sigmoid. we use RBF kernel in this project.

However, sometimes it tends to be hard to find a suitable kernel function to make sample points completely linearly separable. Even if we find one, it is hard to determine whether it is an over fitted result.

Therefore, we can allow for some errors on some sample points and therefore we include soft margin.

We allow for some samples to violate the constraint:

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$

At the same time, we want to minimize the number of samples that violate the constraint, therefore the optimization goal can be:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} max(0, 1 - y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)) \tag{5}$$

where $C > 0$ is a constant.

We include slack variables $\xi_i = max(0, 1 - y_i \left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)) \geq 0$, then we can rewrite (5) as:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \xi_i$$
$$s.t. y_i \left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1 - \xi_i$$
$$\xi_i \geq 0, i = 1, 2, ..., m$$

Again, using Lagrangian multiplier, we get

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu}) = \frac{||\boldsymbol{w}||^2}{2} + C\sum_{i=1}^{m} \xi_i + \sum_{i=1}^{m} \alpha_i(1 - \xi_i - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)) - \sum_{i=1}^{m} \mu_i\xi_i, \tag{6}$$

where $\alpha_i \geq 0$ and $\mu_i \geq 0$ are Lagrangian multipliers.

We have

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x_i} = 0,$$
$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{m} \alpha_i y_i = 0,$$
$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \mu_i = 0,$$

which is

$$\boldsymbol{w}^* = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x_i}, \tag{7}$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0, \tag{8}$$

$$C = \alpha_i + \mu_i. \tag{9}$$

Put these to (6) and we find the dual problem:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}^T\boldsymbol{x_j} - \sum_{i=1}^{m}\alpha_i,$$
$$s.t. \sum_{i=1}^{m}\alpha_i y_i = 0$$
$$0 \le \alpha_i \le C, i = 1, 2, ..., m$$

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}^T\boldsymbol{x_j} - \sum_{i=1}^{m}\alpha_i,$$
$$s.t. \sum_{i=1}^{m}\alpha_i y_i = 0$$
$$0 \le \alpha_i \le C, i = 1, 2, ..., m$$

Then we can again use KKT condition to solve the α_i's and by using $\boldsymbol{w}^* = \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i$ we can get $\boldsymbol{w}^*$. b^* can also be solved in a similar way like hard-margin SVM.

2.2.3 LSTM (Long Short-Term Memory) Model

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. It can process not only single data points, but also entire sequences of data. Different from standard feedforward neural networks, LSTM has feedback connections. LSTM has two states— cell states and hidden states, and it also has three gates. You can refer to the graph below for the illustration of the model.

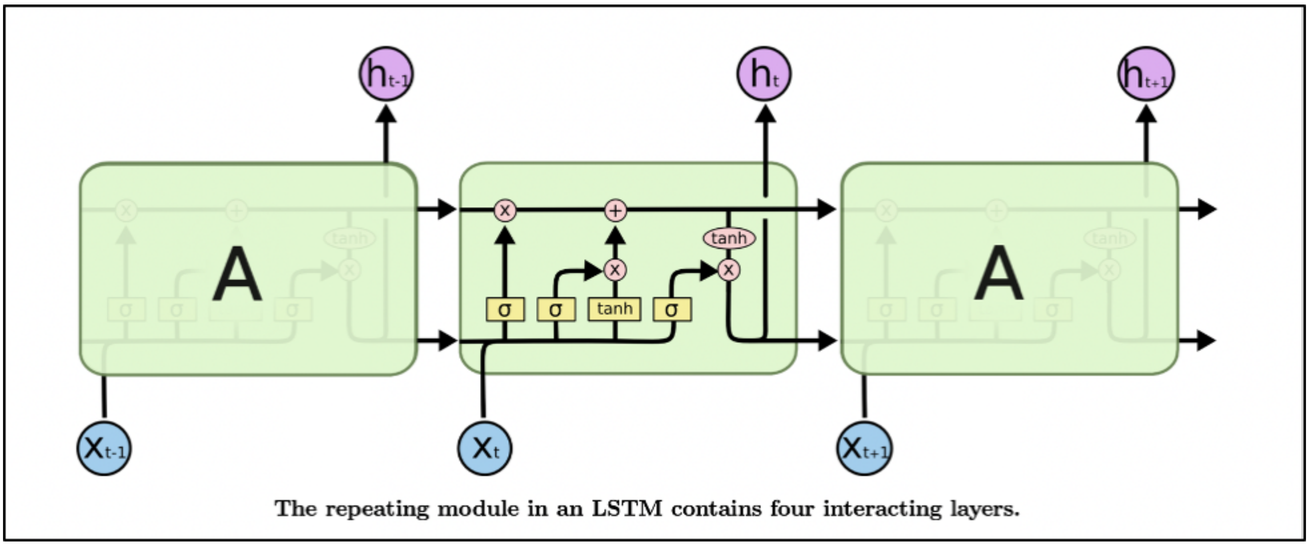

Figure 5: The repeating module in an LSTM contains four interacting layers

The three gates are introduced as the following.

Input Gate: The input gate receives the new information and the prior predictions as inputs and provides a vector of information that represents the possibilities. The input information is regulated using an activation function, and the logistic sigmoid function is commonly used here. The values in the vector will be between 0 and 1. The highest number is more likely to be predicted next.

Forget Gate: This gate removes the information that is no longer useful in the cell state. The two inputs, which is the new information at a particular time and the previous prediction, are fed to the gate and multiplied by weight matrices. Then, the result is passed through an activation function that gives 0 when the information should be forgotten or 1 when the information must be retained for future use.

Output Gate (selection gate): The output gate makes a selection based on the new information and the previous predictions. The final prediction will be a result of multiplying the result of the output gate and the normalized possibilities that are provided by the cell and the input gate.

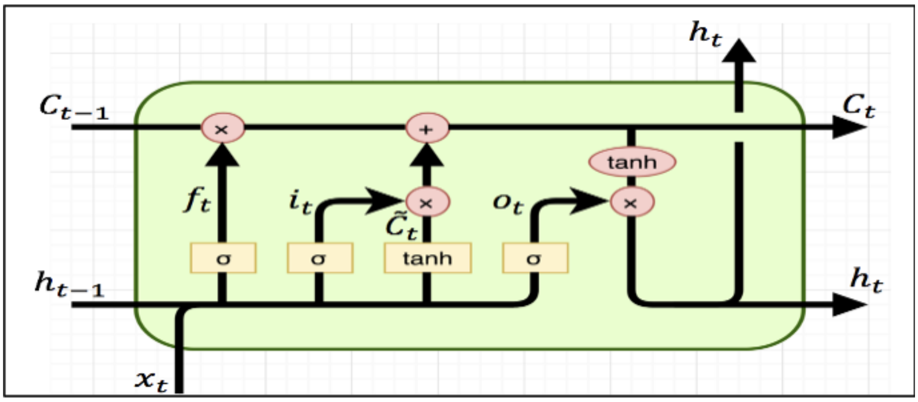

Figure 6: The detailed module in an LSTM

LSTM defines 3 basic requirements of a recurrent neural network:

First, the system be able to store information for an arbitrary duration.

Second the system be resistant to noise (i.e. fluctuations of the inputs that are random or irrelevant to predicting a correct output)

Third, the system parameters be trainable (in reasonable time).

The formula used in LSTM is presented as follows:

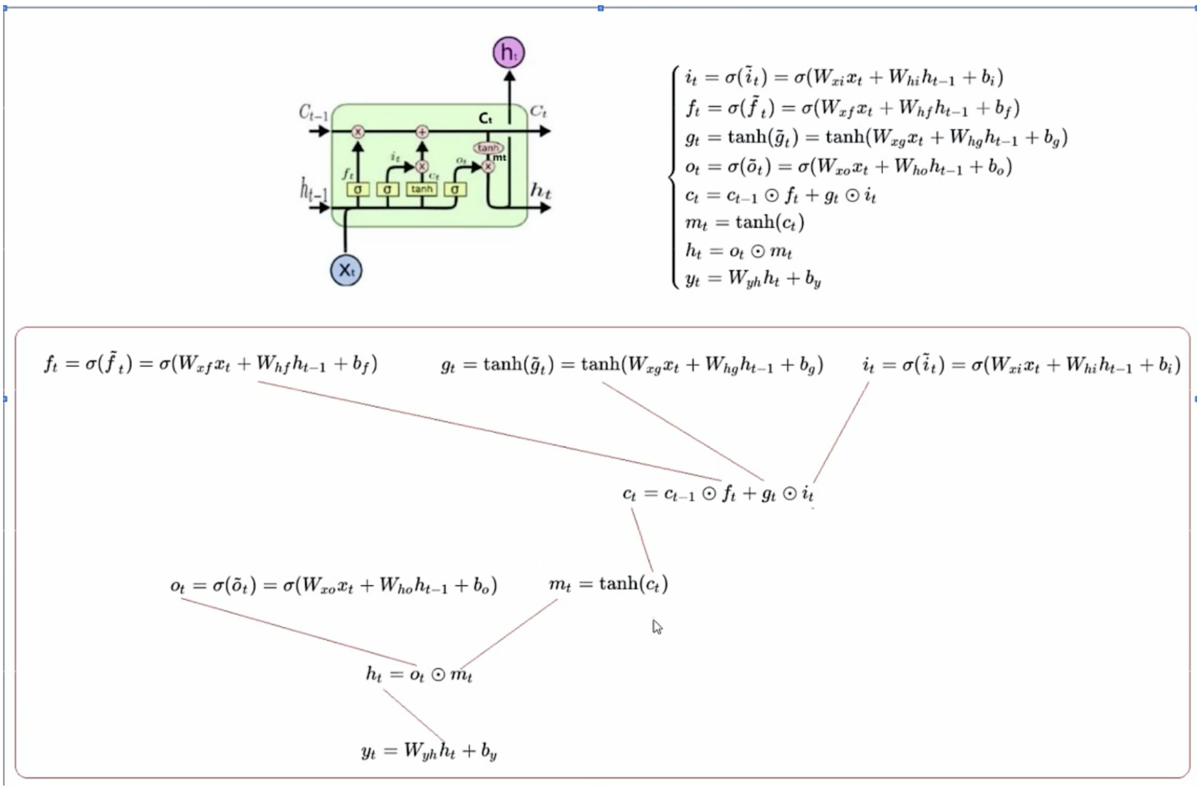

Figure 7: The mathematical formula for detailed module

3 Empirical Analysis

3.1 LPPLS (Log Periodic Power Law Singularity) Model

We feed the historical log price of Bitcoin to the model, and the output of the model is the expected log price of Bitcoin, which you can refer to the graph below.

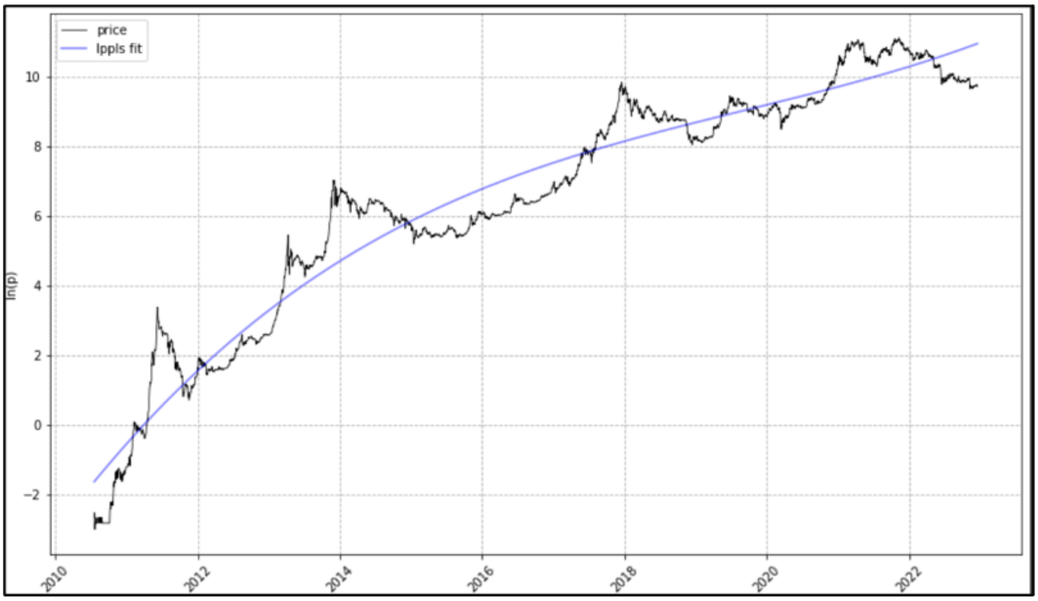

Figure 8: The expected log price from LPPLS model

The LPPLS model defines bubbles when the actual log price exceeds the expected log price produced by the model. We tried to color the bubble periods based on the definition, and the following graph presents the results of bubble periods defined by the LPPLS model.

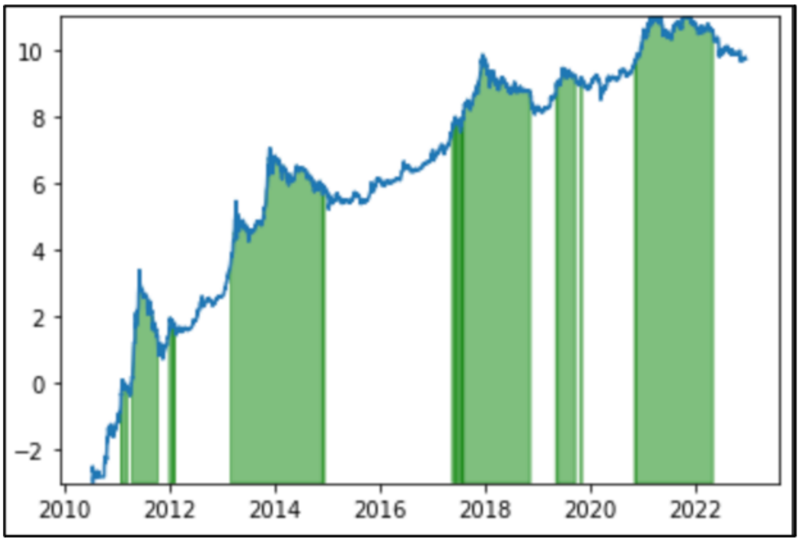

Figure 9: Bubble periods defined by the LPPLS model

The LPPLS model has 74.05% accuracy of bubble-detection, which means it matches 74.05% of the true data defined by Wikipedia and relative new sources. As we can tell and see from the graph above, LPPLS model captures the four bubble periods in the cryptocurrency market. However, it also produces and captures some noise.

Confidence indicators produced by the LPPLS model also helps us to identify bubbles. We set the smallest window size as 30, window size as 120, and the step size as 5 to produce a group of time series data. We then use the data to produce confidence indicators as you can see from the following graph.

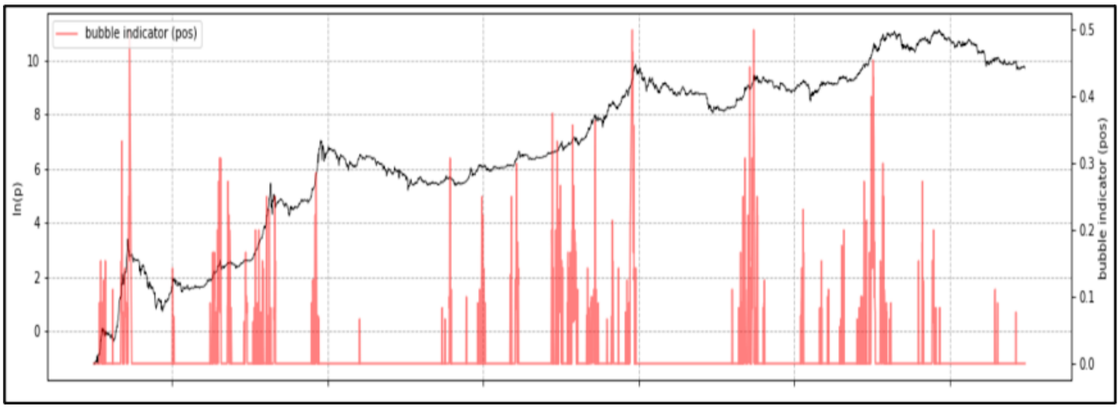

Figure 10: Confidence indicators for positive bubbles

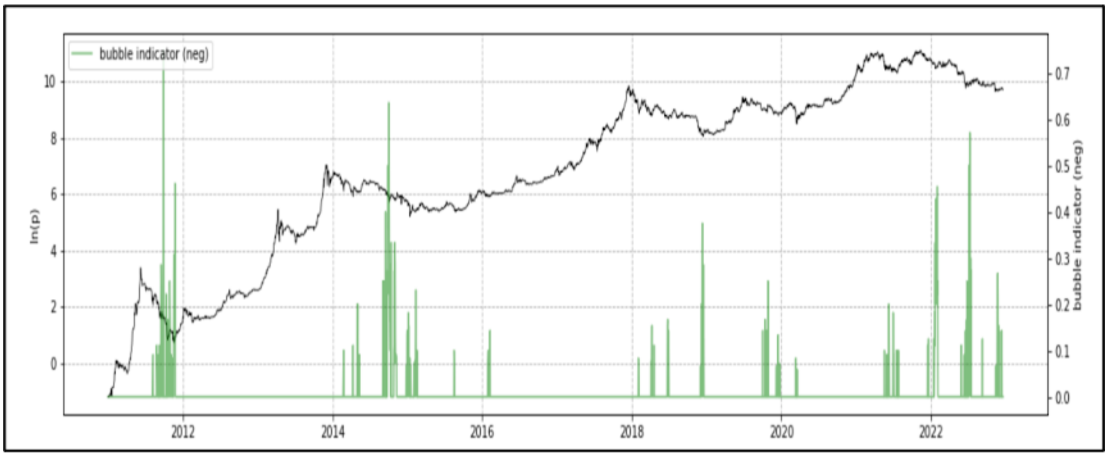

Figure 11: Confidence indicators for negative bubbles

The confidence indicators for positive and negative bubbles can assist us in identifying bubbles, and we can further set a threshold to help filter out the noise. In other words, we keep the indicators only if their confidence level is bigger than the threshold. We thus set different thresholds here and compare the results.

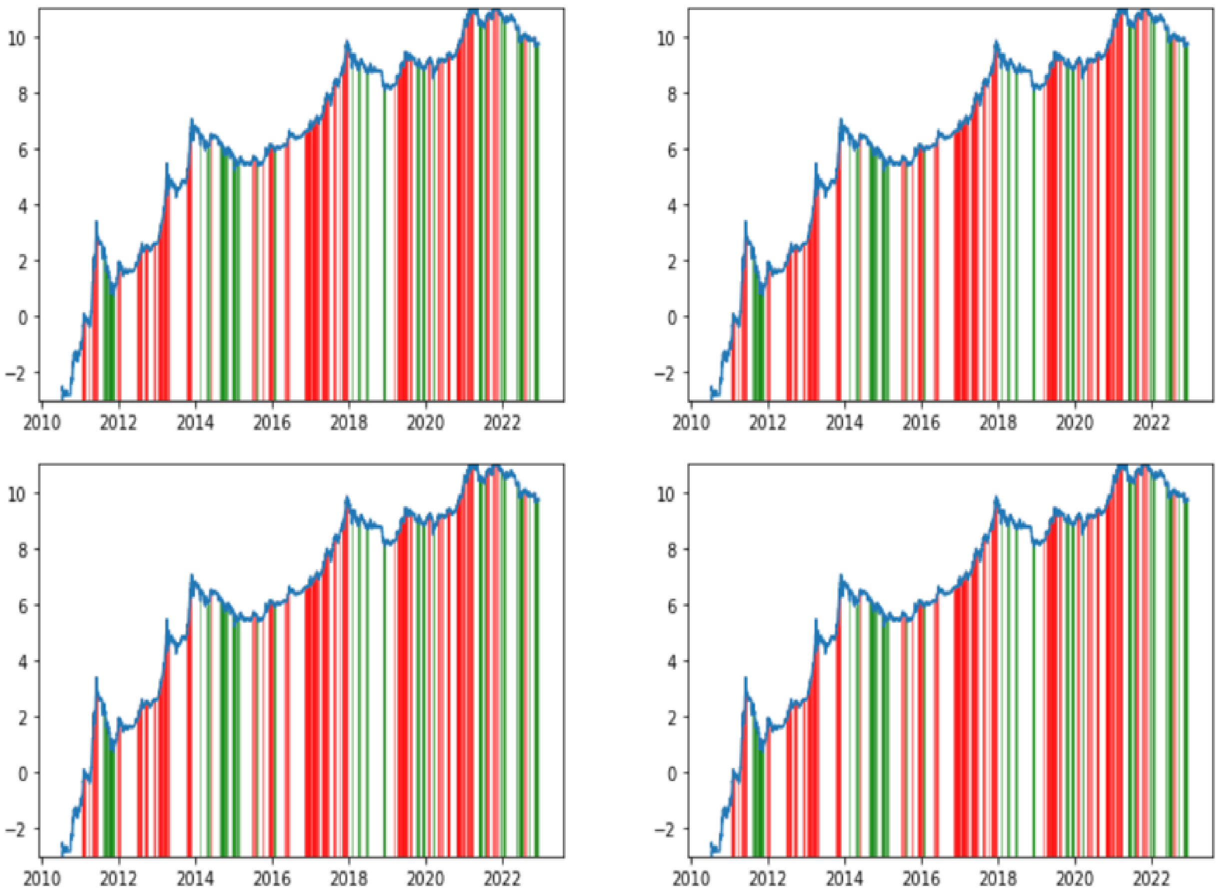

Figure 12: Comparison of different thresholds

As we can tell from the graphs above, it is important to set a threshold to get rid of noise. To conclude, the LPPLS model can identify bubbles based on the expected log price produced from itself and the confidence indicators, but it also captures noises. Thus, we can get rid of these noises by data pre-processing and setting thresholds for confidence indicators.

3.2 SVM Model

The target we use is binary bubble variable where 1 means there is bubble on that day and 0 means there is no bubble on that day. We first need to preprocess the data. The bubble data is daily while the macroeconomic data are mostly monthly and quarterly. Therefore the macroeconomic data are converted into daily by filling each day with the most recent data up to that day. After these preprocessing, all the data are daily and after removing the initial vacant values from calculating returns. The dataset starts in 2010-07-27 and ends in 2022-12-18. Then for the SVM model, we choose RBF kernel for the SVM model and we will get the training and test accuracy for each value of C = 0.01, 0.1, 1, 10, 100, 1000. We put the first 80% of data as training set (2010-07-27 to 2020-11-18) and the last 20% of data as test set (2020-11-19 to 2022-12-18). We standardized the features before fitting the model. After fitting the model we calculate the accuracy of the model. Accuracy is defined as number of correct predictions divided by the number of total sample, indicated by %:

Table 1: training accuracy and test accuracy given different values of hyperparameter C

C	**train**	**test**
0.01	75.18	100
0.1	85.78	100
1	94.40	100
10	98.95	99.47
33	99.84	97.63
55	99.93	95
78	100	92.63
100	100	91.97
1000	100	92.11

SVM model performs very well for both training set and test set, also for any value of C. When C is greater than 1, the training and test accuracy are all over 90%. Accuracy of training set gradually increase as C increases. This is understandable because the punishment to wrong classification gradually increase so there are fewer and fewer wrong predictions for training set. Accuracy of test set gradually decrease as C increases because of overfitting. The high test accuracy may be because the dates we set as test set are all bubbles and as the combination of features are all on the bubble side of the fitted hyperplane, they are all predicted accurately.

3.3 LSTM Model

The training accuracy of the LSTM Model on this problem is 57.35% and testing accuracy is 24.62%. We can see that LSTM Model performs not well on this problem. From our perspective, the main reason should be focused on the imbalanced data and feature engineering. For one thing, we use the daily frequency data and bubble is rarely happened in the whole history (it happened just 4-5 times in the past.) and most of the training data are 0, which represents non-bubble periods, in the target value, while in the test data, most of them are 1, which indicates bubble periods. This causes the imbalanced data.

For another, the way to select features is also an important problem. In this case, we use a lot of macro-economic factors, and test their relation with Pearson correlation coefficient, but it is a measure of linear correlation between two sets of data. It is the ratio between the covariance of two variables and the product of their standard deviations. Thus, it is essentially a normalized measurement of the covariance, such that the result always has a value between 1 and 1. As with covariance itself, the measure can only reflect a linear correlation of variables, and ignores many other types of relationships or correlations. To conclude, for this model, we have a lot of work remain to be done to improve the performance and the accuracy.

4 Conclusions and Future Work

In this paper, we define the bubbles based on the events and dates based on relative websites and news sources. Further, we use models to make attempts to identify and even predict bubbles. Based on our research, there are few things needed to be addressed.

First and foremost, the general definition used to bubbles is defined as a situation where the asset price exceeds its fundamental value. Thus, for future research, we can devote ourselves to determining the fundamental value of cryptocurrencies in order to have a more accurate definition to bubbles and 'true' bubble periods.

Next, the LPPLS model, the SVM model and the LSTM model are all powerful tools in the field of machine learning. In our study in using the models to identify bubbles, we find out that the LPPLS model indeed has the ability to help identify bubbles. With its output of the expected log price and its definition to bubbles as well as the confidence indicators, they are all powerful in identifying bubbles. However, the model is prone to capture noises. Thus, by manipulating data pre-processing and setting thresholds for confidence indicators, we can reduce and filtrate the noises.

As for the SVM model, it is abnormal for the SVM model to have a such high accuracy in identifying and predicting bubbles. Three problems need to be addressed. First, we split the data into a train dataset and a test dataset with a proportion of 80-20. However, the 20%

test dataset comprises of mostly 1, which represents bubbles. That is, the way we split causes to produce imbalanced train and test datasets. We think that the problem can be addressed by using time-series splitting method to balance two datasets and thus the model can be evaluated more accurately and properly. Further, the features can be selected and evaluated more rigorously. Most of the features we use in the SVM model and the LSTM model are macro-economic factors. However, the correlation and relationship of them and the Bitcoin prices needs to be further verified. Third, the way we manipulate data needs to be addressed as well. The Bitcoin data we use here is daily-basis while many macro-economic features are monthly- or quarterly-basis. This kind of imbalanced data needs to be handled properly in order to produce an accurate result.

Last but not least, the LSTM model has a low accuracy in identifying and predicting bubbles. Likewise, the most critical issue is the imbalanced datasets. We split the dataset into train and test datasets with the proportion of 80-20, just like what we did in the SVM model. As far as we are concerned, the LSTM model captures and remembers the old memories, which makes the model prone to make a prediction of 0, whereas the test dataset is mostly comprised of 1. Thus, it is important to have a proper way to handle and split the imbalanced dataset. Another thing is the design of the model itself. In our opinion, the LSTM model is more suitable for predicting sequential continuous variables. Hence, we can refine our approaches and try to predict continuous variables in the future, or we can try to implement different deep learning models that is more suitable for predicting binary labels.

To sum up, we here implement different models to try to identify and predict bubbles. Nonetheless, further improvement can still be made. We need to determine the fair value of cryptocurrencies, and the improvement also includes data pre-processing, imbalanced data manipulation and feature engineering. Aside from that, other approaches and models can be used as well. The approaches include trying to predict continuous variables and use them to identify or predict bubbles as well as trying to implement different statistical, machine learning and deep learning models.

5 References

https://andreybabynin.medium.com/log-periodic-power-law-singularity-model-and-tesla-stock-b775a8b44d26 [Andrey Babynin, 2020] Log-periodic power law singularity model and Tesla stock, September 2020.

https://github.com/Boulder-Investment-Technologies/lppls [LPPLS, Github]

https://www.tandfonline.com/doi/full/10.1080/02664763.2020.1823947 [Fatma and Kamil, 2021] A two-step machine learning approach to predict SP 500 bubbles, Journal of Applied Statistics, Vol. 48, 2021.

https://en.wikipedia.org/wiki/Cryptocurrency$_bubble[Cryptocurrencybubble, Wikipedia]$

https://www.sciencedirect.com/science/article/pii/S0378437120302077 [Shu and Zhu, 2020] Real-time prediction of Bitcoin bubble crashes, Physica A, Vol. 548, March 2020.

[Gilles and LeRoy, 1992] Asset Price Bubbles, The New Palgrave Dictionary of Money and Finance, 1992.

[Zhang et al. 2016] LPPLS Bubble Indicators over Two Centuries of the SP 500 Index, SSRN Electronic Journal, 2016.

https://www.sciencedirect.com/science/article/pii/ S0167268120301505 [Enoksen et al. 2020] Understanding risk of bubbles in cryptocurrencies, Journal of Economic Behavior and Organization, Vol. 176, 129-144, June 2020.

https://doi.org/10.48550/arXiv.2002.06405 [Bashchenko and Marchal, 2020] Deep Learning for Asset Bubbles Detection, arXiv, Feburary 2020.

https://www.mdpi.com/2079- 9292/11/15/2349 [Ammer and Aldhyani, 2022] Deep Learning Algorithm to Predict Cryptocurrency Fluctuation Prices: Increasing Investment Awareness, Electronics, Vol. 11, July 2022.

[Sornette, Johansen and Bouchaud, 1996] Stock market crashes, Precursors and Replicas, Journal de Physique I, Vol. 6, January 1996.

https://www.sciencedirect.com/science/article/abs/pii/S0378437120304611 [Shu and Zhu, 2020] Detection of Chinese Stock Market Bubbles with LPPLS Confidence Indicator, Physica A, Vol. 557, November 2020.

https://www.sciencedirect.com/book/9780128144824/computational-learning-approaches-to-data-analytics-in-biomedical-applications [Khalid and Gayla, 2019] Computational Learning Approaches to Data Analytics in Biomedical Applications, 2019.