CryptoComparator

A data visualization tool for trend analysis and similarity relations between cryptocurrencies

Matteo Emanuele
`emanuele.1912588@studenti.uniroma1.it`

Pietro Manganelli Conforti
`manganelliconforti.1754825@studenti.uniroma1.it`

Pietro Nardelli
`nardelli.1546380@studenti.uniroma1.it`

March 2021

1 Introduction

Since the birth in 2009 of the first cryptocurrency, Bitcoin, numerous private cryptocurrencies have been introduced. Bitcoin is by far the most successful one, but many other gained their space in the market, drawing a lot of media attention as a whole new financial phenomenon identified with "cryptocurrencies phenomena", which in 2020 reached a total market value higher than 950 billions of dollars. Cryptocurrencies such as Bitcoin, Ethereum or Litecoin have been attracting the attention of information technology professionals, economists, investors, banks, government, and even the police. This technological novelty has increased over the last years due to its innovative features, simplicity, transparency, high market price and popularity. Cryptocurrencies challenge the current financial systems and conventional forms of currency. The underlying purpose behind the cryptocurrency movement is related to the decentralization of power; they are not controlled by a central bank or government.

Since the behavior of a single cryptocurrency is exactly the same of a general coin, but with more volatility, investing and predicting the trends may be complex. The usage of a tool for analysis of small groups of cryptocurrencies and/or couples to compare them efficiently can be of extremely help for interested users.

In the following report we will discuss our tool for cryptocurrencies comparison, that we named *"cryptoComparator"*. Its general goal is to allow the user to analyze trends of different features of a coin and give the possibility also for cross analysis togheter with other cryptocurrencies in order to be able to recognize patterns that can be exploited for general decision making in the finance field.

In the following report we will go through the dataset used and how we treated it in order to gain the best out of it. Then we will dive into the visualizations implemented, a core element in this work. We will discuss, subsequently, the interactivity part of our data viz, and ultimately we will briefly talk about the positioning of our work with respect to other reaserch works that represent the state of the art for the visual analytics.

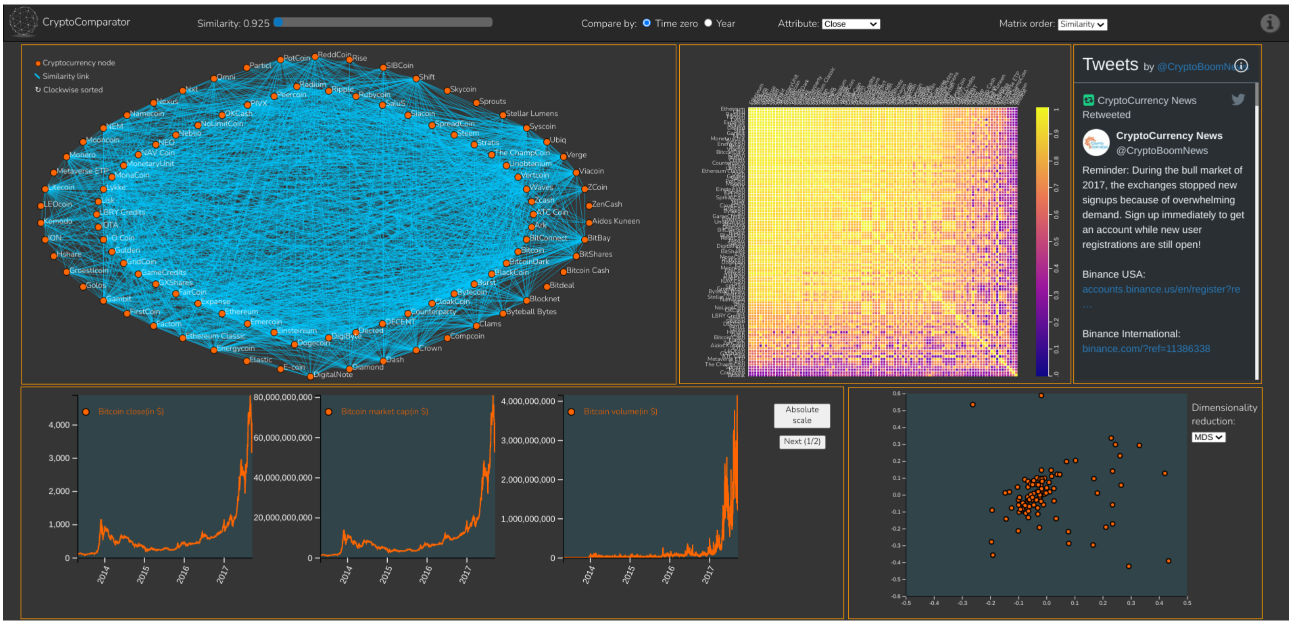


Figure 1: Homepage of CryptoComparator.

2 Dataset

2.1 Overview

We worked on a collection of data of a set of cryptocurrency, a hundred to be specific.
The dataset present itself as a collection of 100 sub-dataset, one for each coin.
Every crypto presents a varying number of points, which depends on its day of birth. Bitcoin,for instance, since it is the more long-lived cryptocurrency of all, has the highest number of points, ranging from april 2013 up to the end of september of 2017, which is a common day among all the sub-datasets, since it is the day the data were collected. So in general, a specific set of points related to a single coin, will range from the day finance data were present online, up to the last week of september(22th-24th).
Inside a sub-dataset, we can notice how a sample of data are representing financial information related to a single day of the market.
We have 6 different attributes: *Date, Open value, Low value, High value, Close value, Market Cap, Volume.* We will explain each one of them in order:(*note: each of these elements are measured in dollars*):

- *Date:* it shows the date related to each sample(i.e. "Sep,22,2017").
- *Open Value:* it records the value of a single unit of the cryptocurrency at which opened the market in that specific day.
- *Low value:* it records the minimum value reached by the cryptocurrency in a specific day.
- *High value:* it records the maximum value reached by the cryptocurrency in a specific day.
- *Close value:* it shows the value of the cryptocurrency when the market closed in a specific day. In finance is commonly used for creating the "line charts" for showing the time serie of the coin price.
- *Market Cap:* it shows the amount of money spent into that cryptocurrency until that day.
- *Volume:* its the amount of coins exchanged in that day. It symbolize how actively people trade a specific cryptocurrency in a day.

2.2 Preprocessing

The dataset were unable to be used for how they were presented. Some preprocessing has been necessary in order to use the information we had at maximum potential.
For starters, all the elements except "Date" presented missing values. Since this value were labelled with a

symbol "-" , the first preprocessing we did was to replace such values with 0. This was needed since we did not wanted to create "holes" in our time series removing them entirely. so we adopted the strategy to replace the missing value with a standard value, zero, in order to keep the time serie behavior for each attribute.
Another huge part of the preprocessing was done on the "date" attribute, since the way it was formatted was useless. We approached such preprocessing in two different ways for python and javascript. In python we rewrote the whole date for each day as year-month-day, without anything in the middle. To make an example, the day "22 september 2017" has been converted into "20170922". This is a common format for date domains in python, since this smart format allows to keep the time order also for dates.
For javascript, we formatted the date in another way, since it was required to be in a specific way for the d3.js library. Such format transformed(i.e.) the "22 september 2017" in "Fri Sep 22 2017 00:00:00 GMT+0200 (Ora legale dell'Europa centrale)". Such form for a date can be obtained simply with the new Date() function of javascript.
One last preprocessing was done on the "Market Cap" and "Volume" attributes. This was needed since the values in these two features were of the order of millions/billions, but unfortunately they were formatted in the wrong way, using commas to identify different ciphers. Our preprocessing transformed a sample value from 60,152,300,000 into 60152300000.

2.3 Data manipulation

In order to compare cryptocurrencies between each other we have computed a similarity value for each pair of cryptocurrencies in our dataset. There are different techniques in statistics that can be used to determine how one dataset is associated with another. One of the most commonly used technique is the so-called Pearson correlation coefficient r. Given two signals x, y (i.e. time series) the Pearson correlation coefficient is:

$$r(x,y) = \frac{Cov(x,y)}{\sigma_x \sigma_y} = \frac{1}{N}\sum_{i=1}^{N}\frac{(x_i-\mu_x)(y_i-\mu_y)}{\sigma_x \sigma_y}$$

where N is the number of samples, μ_x and σ_x are the mean and standard deviation of x, respectively, and μ_y and σ_y are the mean and standard deviation of y. The measure of the correlation, always lies between 1 and -1. A correlation coefficient of +1 signifies perfect correlation, a value of -1 shows that the data are negatively correlated (anticorrelation: if one variable is increasing, the other is decreasing) and finally, a correlation coefficient of 0 represents no correlation [4]. To obtain the final similarity value $s_{x,y}$ we decided to normalize $r_{x,y}$ between 0 and 1. In this way, our final similarity encoding for two time series x, y is:

- $s_{x,y} = 1$: they are similar so they have the same trend (correlation);
- $s_{x,y} = 0.5$: they are not correlated (independent);
- $s_{x,y} = 0$: they have an opposite trend (anticorrelation).

For each attribute of a cryptocurrency dataset we decided to compute two different type of similarities based on how two time series x, y are manipulated (e.g. $x^{BTC}_{marketCap}$, $y^{ETH}_{marketCap}$):

- Time zero: we first horizontally shifted one of the two such that they both start from the same time t_0. Than, in order to use Pearson correlation, the longest one is truncated so that they have the same length.
- Year: we compute the Pearson correlation coefficient only on two time series that have both samples for the whole selected year. In this way, the truncation is made from the start of the selected year to the end of it (e.g. $x^{BTC}_{marketCap,2017}$, $y^{ETH}_{marketCap,2017}$) so that we can use a straightforward approach while formulate a similarity that focuses on a particular span of time. We decided to use only the years between 2015 and 2017 because these are the ones in which the cryptocurrencies of our dataset were the most active.

3 Visualization

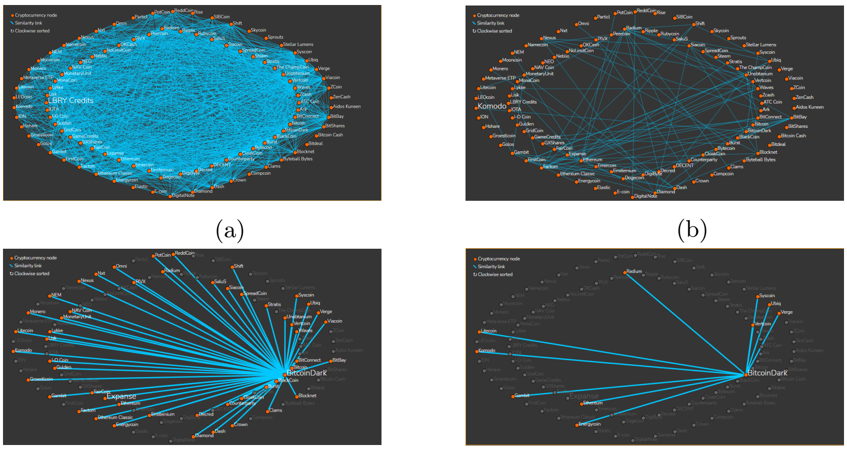
3.1 Elliptic Network Graph

The main visualization element, where the user can visualize and choose among all the various cryptocurrencies of interest, is the network graph. In this network are visualized all the currencies as nodes, disposed in an alphabetical order, and all the similarities between them as links, according to a specific controllable threshold 2(a)(b). When a node is clicked then is highlighted the relative subgraph containing all the links which target

the pressed node and all the source nodes of these links, while shadowing other elements 2(c)(d). This graph visualization, as explained in the next section, is not necessary to interact with the rest of the data although is recommended.
The representation aims to give a clean interactive visualization of the dataset on a network, basing on the classical approaches as the circular network graph or the usual scattered one. These last ones present both the problem to display some overlapping information, especially in a basic not ordered graph where links, nodes and node texts can collide easily if the number of information to display is high. If our representation was smaller, for example counting a smaller number of nodes (so a comparison between less cryptocurrencies, es [7]), those last two representations would have been a valid solution. But in our case the visualization represents a large dataset, with the double purpose of:

- being visually useful during a node search;
- distinctly expose the whole graph and sub-graphs generated on click, without the problems explained above.

In particular without a disposition schema the user undergoes into an inefficient search problem when looking for a specific node of interest, so a disposal for nodes is needed. An early solution we used was the circular shaped network sorted alphabetically, where all the nodes were at a fixed distance along the perimeter of a circle. In that way choosing one value among the others was much easier than looking for it among all of them while scattered. Unfortunately, this solution still wasn't sufficiently clear, with many nodes to visualize (one hundred in our case) and links (which can be thousands). To overcome this we arrived to the solution of disposing them along a double concentric elliptical network graph, still maintaining the alphabetical order and stretching the visualization on the horizontal axis. This led to an easier way to search for a node while preserving the readability of the overall visualization. The solution also follows the natural orientation of a text, so it doesn't need to rotate them radially as for the circular shape case and allows a bigger font size, so to an improved visibility.
Moreover, visibility is also enforced by the introduction of a spike at the top and the bottom of the function of the ellipse, and therefore nodes are placed on a path which provides a better displacement between them in the critical point of the representation (at the top and at the bottom).
To improve the user experience and to exclude any problem in the interaction with the graph, the implementation keeps the possibility to grab and move nodes around (pressing a node or on the corresponding text) in any condition. Naturally links follow the moved node, giving the possibility to reshape the graph into arbitrary shapes.
Lastly, a small legend describe the basic meaning of nodes and links, while for the full explanation hovering with the mouse on the "i" icon (top right corner) shows the information above the graph cell.

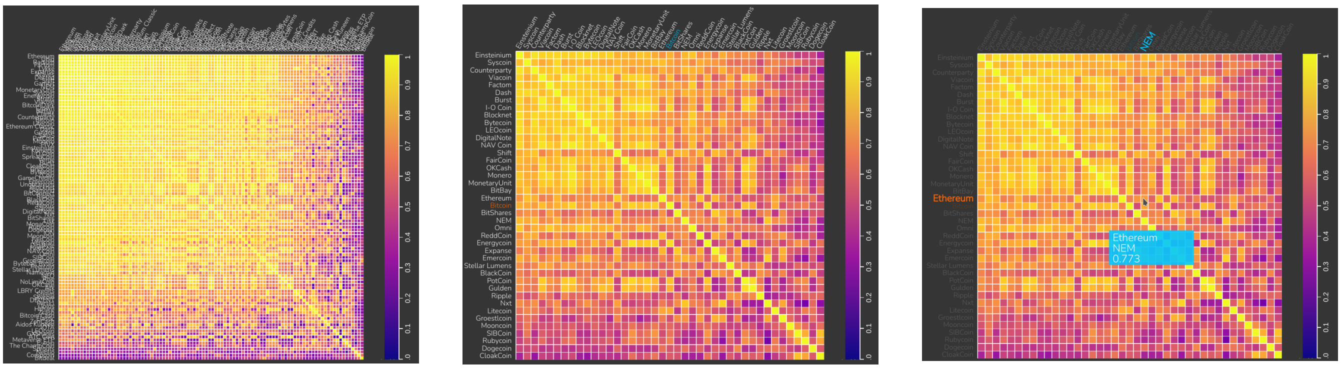


(a) (b) (c) (d)

Figure 2: Similarity graph visualization: (a) shows the full graph, with starting similarity value and slider threshold; (b) shows the reduced graph where the threshold value for similarity obscured all the links under the chosen threshold; (c) An example of subgraph; (d) A reduced subgraph.

3.2 Similarity Heatmap

As we said in section 2.3, we computed the similarity of each pair of cryptocurrencies starting from the correlation coefficient and normalizing it between 0 and 1. The most popular way to describe correlation is by a correlation matrix. The correlation matrix is a symmetric matrix with all diagonal elements equal to $+1$. In our case, we provide a similarity matrix that has the same structure of the matrix previously described with the difference that values vary between 0 and 1. In order to make the differences between similarity values easier to perceive (reading tabular data may be difficult) we built a color coded similarity matrix (heatmap).
The heatmap is a data matrix that visualizes values in the cells by the use of a color gradient. This gives a good overview of the largest and smallest values in the matrix, revealing any patterns and displaying whether any variables are similar to each other [10]. The data contained within a cell is based on the relationship between the two variables in the connecting row and column. The default similarity heatmap has dimension 100×100, showing to the user an overview of the similarities for all the cryptocurrencies in that particular selection (as we can see in figure 3(a)). Based on some interactions, as we will speak in section 4, the matrix will be reduced (having a lower bound of 1×1 dimension). The reduced matrix is shown in figure 3(b).
A good color scheme is an essential factor for correct interpretation of the heatmap. The choice vary between a diverging and sequential color schemes. Diverging palettes fix colors in both lower and higher end of the data and in the middle; they are better suited for data that range to both, negative as well as positive, directions (e.g. difference from some reference). Instead, sequential palettes fix the lowest and the highest value; they are more appropriate for non-negative data (e.g. percentage between 0 and 100) [10]. For our purpose, we decided to use a sequential multi-hue scheme called "plasma" that might be more suitable to clearly see some differences in similarity values. This color-scheme can be fully displayed at any time thanks to the legend that is required alongside the Heatmap in order for it to be successfully read. A grey color is additionally used when no similarity can be computed (e.g. at least one cryptocurrency was inactive in that particular selected year).



(a) (b) (c)

Figure 3: Similarity heatmap visualization: (a) shows the matrix with dimension 100×100; (b) shows the reduced matrix where elements are the nodes of the subgraph with respect to the cryptocurrency clicked by the user and the threshold value for similarity; (c) An example of mouseovering on a cell.

3.3 Line Charts

Line charts are one of the classic approach for visual inspection for time series. If the time serie is only one, a single line chart can give you all the information required to grasp every behavior in the trend over time. If the comparison and the cross analysis may be the focus of a user, to actually have both of the time series represented on a shared domain between the two is usually the key to victory. Two common approaches are either representing the lines on the same chart with different colors, or representing the two lines in two different charts one above the other in order to be able to compare specific sub-domain windows. In our work we leant towards the first approach, the "multiple line chart", since we believe that the combination of interactivity and the double lines was a key combination for exploratory analysis the user may be interested to.
A picture of the line charts is reported in figure 4.

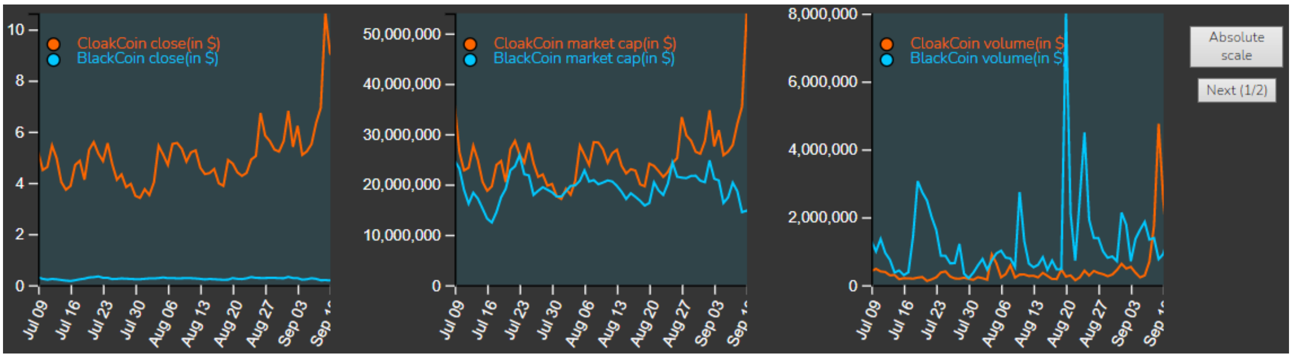


Figure 4: Line charts created by selecting two different cryptocurrencies.

As we can see at screen there are three charts, and there is a button indicated with "1/2" that allow the user to switch from these three charts to other three charts. The totality of the six charts allow the user to see the time series of all the attributes associated to a cryptocurrency.
Each chart is highly interactive: a brush for zooming was implemented. When it zoomed, the axis translate dynamically from the previous state to the new one, computing the new domain and codomain for the axis on-line.The domain recomputed is quite obvious, what is more interesting is that also the codomain change with respect the maximum value in the time window selected,in order to allow the user to explore also small values on the trends. A minimum time window width was introduced to stop the user to zoom indefinately until reaching a point where no data were available: a time window of 2 days was set as the minimum explorable time window. Whenever one of the charts is zoomed, all the other charts respond togheter, modifying themself at the same time, both in domain and codomain, in order to keep track of the time window for each attribute. The visual exploration that the user seek may be dependant on wanting to analyze data on an absolute scale or on a relative scale: this led us to implement the possibility to switch from one scale to another through a button in order to give the free choice to the user.
Last note about the charts is that they are built to work exactly in the same way also with the representation of only one line except of two: this give to the user to possibility to explore also a single cryptocurrency alone, and not only in relation or togheter with another one.

3.4 Dimensionality Reduction Scatterplot

Dimensionality reduction techniques where a mandatory element for the project. We decided to implement two different dimensionality reduction approaches: Multidimensional Scaling and Principal Component Analysis.
The results where then plotted in two dimensions, generating a scatterplot, probably the most famous and common way to visualize dataset with techniques-reduced dimensions. A picture of the scatterplot can be seen in figure 5

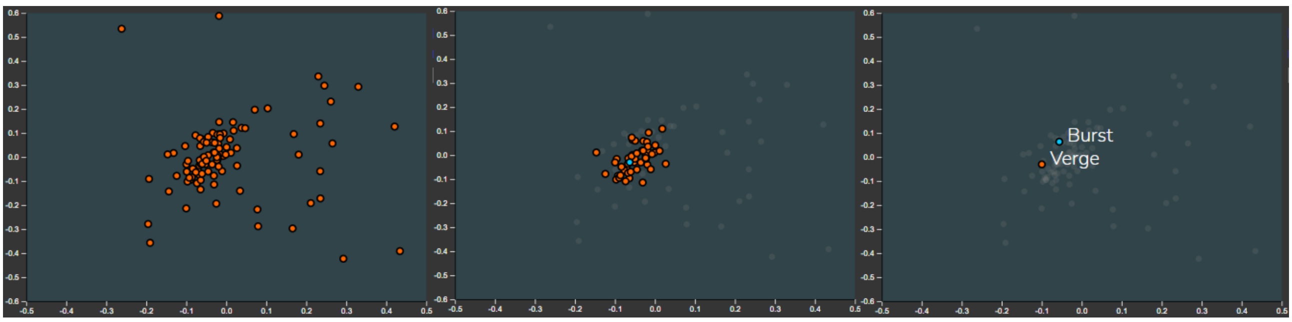


Figure 5: MDS scatterplot. The last two images show the result of a sequence of user interactions.

As we can see we have three modalities of usage for our scatterplot. The left most shows how we can have a representation of the totality of the nodes and how they got plotted on a bidimensional space. This version of the graph allows the user to search manually for outliers, since each circle on the screen is responsive to the user mouse cursor: it increase in size when hovered over, and the name is shown. Additionally,the single colored

circle is possible to be clicked to see the line charts also for the single cryptocurrency selected.
In the middle scatterplot we can see a subgroup of cryptocurrencies selected. In this case, the selected ones are still interactive, presenting all the functionalities previously mentioned, while the obscured ones are not interactive. As we can see there is a light blue circle: this is different since it is the center of the sub-graph generated clicking a coin on another visualization,the double elliptic graph.
The last scatterplot in the picture shows a couple of cryptocurrencies. Since the user may be interested in analyzing the position in the bidimensional space of a coin with respect to the other, such visualization can be handy to the user. In this modality, the name are explicitely written, and they do not require the user to hover the mouse cursor over the circle.
Last element of interaction for the user is the dropdown menu: from this we can select which dimensionality reduction techinque to use. Since we said we have used both PCA and MDS, we have left to the user the possibility to choose which one of the two he/she wants to see plotted. This gives free choice to the user, maximizing the possibility to fullfill his/her needs.

3.5 Twitter window

The cryptocurrencies world is a fast changing one, indicated as an high-risk speculative product: unexpected changes in market sentiment can lead to sharp and sudden moves in price, making it an high-risk with high variance investment. Therefore is unstable, also due the fact that is unregulated, in fact crypto-market manipulations can happen at any time, market is always open and full anonymity is exploited by illegal activities . Then, for completeness and for a continuous update we decided to implement a twitter window, tuned to the last news about this world.

4 Interactions

4.1 Menu

The menu on the top of the page contains a slider, a drop down menu and two radio buttons. The first one, the slider, determines one of the main graph variable value which is the similarity threshold. Each link for construction yields a similarity value between the source and the target nodes, which is used to hide/recover it when asked by the user through the slider. This value is also connected with: the scatterplot, reducing/increasing the highlighted points in it; and with the heatmap size, being a measure of the subgraph dimension.
The radio button allows to choose between the possible similarities, which are 24: 6 for each attribute in relative time and 6 for each yearly measurement of the similarity(2015-2016-2017). After the selection, all the links threshold values are updated and consequently is the graph.
Being the similarities very different, each one of them have to relate to a different starting value from the slider (and so to a threshold value which determines if the link will be added to the graph or not). This in order to have, when possible, a maximum of visible links, to not overflow the visualization.
The matrix is by default ordered basing on similarity values in descending order. But thanks to a dropdown list the user can select also the ordering by name in order to make the search of a particular cryptocurrency easier.

4.2 Elliptic Network Graph

Graph changes produced by user interaction can be described on 3 levels: intra-graph, from the graph to the other representation and from the menu(explained above).

4.2.1 intra-interactions:

The intra-graph interactions are three, the first, as explained in the previous, is when a node is grabbed and moved around from the node circle or node text. This moves also all the links visible at the moment, and is available also for shadowed nodes (in gray).
Another small interaction is respect to the node text size, which is increased when the node is pressed or if the mouse moves over it, giving a better readability to the text and to the one which is being currently selected.

The last interaction is the main one, the selection of a node. This expresses the user interest on only that particular node and its connections (so similarities). In the page, this reflects into the highlighting of the subgraph: the node selected name size increases, each link which targets the node becomes thicker and each node connected to one of these links stays colored. All the other elements are obscured, and everything returns to normality if the same node is pressed again, returning to the full graph(but always in respect to the slider value).
When the user clicks on a node, the matrix will be recomputed basing on the threshold value and the selected node. It will show a matrix based on all the cryptocurrencies that have similarities greater than the threshold value with respect to the selected node. The selected node name will be highlighted in the matrix column and row, respectively with a bright blue and orange color, so that the user can have a reference point while navigating. Moreover, if the number of nodes in the matrix is low, the text is rotated in order to show it horizontally and enlarged to occupy as much space as possible (limited to the maximum cryptocurrency name length).

4.2.2 inter-interactions:

On the click of a node extra-graph effects are produced with all the other representations(except the twitter one), as explained in the respective paragraph. This causes the tuning of all the page elements respect to the user choice.

4.3 Similarity Heatmap

4.3.1 intra-interactions:

- mouseover on a cell: when the user hover its mouse on a cell of the matrix, it will show a pop-up window where the compared cryptocurrencies names appear together with their similarity value. Moreover, their names will be enlarged and colored in the same way the charts will display their time series; meanwhile, all the other names will be greyed out (as we can see in figure 3(c)). This feature was found necessary to make it possible to show the matrix texts even with an high number of cryptocurrencies , bringing out the selection of nodes and increasing the readability.

4.3.2 inter-interactions:

- click on a cell: when the user clicks on a cell of the matrix the charts shows both cryptocurrencies time series. At the same time, the scatterplot will display the names of the two selected cryptocurrencies while highlighting the nodes. To make the visualizations more interconnected and user friendly, both time series and scatter plot use the same two colors to distinguish between the cryptocurrencies pair selected on the heatmap cell.

5 Positioning

Comparators of cryptocurrencies aren't very common object of study in the accademic research. No many papers were found totally dedicated to the comparison of financial time series. We will discuss in this last section the inspiration that moved us towards the choice we did for the data visualization.

5.1 Elliptic Network Graph

Our graph is a development of the more used circular shaped graph, as explained above, and represents a double concentric elliptic function. An example of application where cryptos are compared in a circular graph can be found in the paper [7] , where each node represents a cryptocurrency and each link a similarity between them. The similarity value is calculated basing on the parameters as the daily return or the fundamental characteristic of the crypto(as the hashing function used or the encryption used)
In our case instead we try to express visually one hundreds nodes, with the intent to give a clear but useful visualization for a node search. One example of an unordered graph regarding cryptocurrencies is found at [3], particularly interesting for us because represents large transactions of cryptos between entities across the years, starting from Shatoshi Nakamoto in the center. As in our case it takes into account the time differences, but they visualize them through the concentric structure in order to expose the time difference represented by the distance between the center and all the other elements. Here we can see how the representation is well fitted

for their aims but not for our purpose, where the node name is an important information which can't be hidden and shown only when the mouse hovers on a node.

5.2 Similarity Heatmap

As we already said, heatmap is one of the most popular visualization when it's necessary to describe a similarity measure. The color coding was the most difficult choice because, in most of the cases, the differences between similarity values are subtle. At the end, we decided to use the color scheme "plasma" that is very similar to the one used in [5] but is more consistent with the chosen colors of other elements. Differently from them, we opted for a continuous colored legend because the information of the distribution of the similarity values can be already observed exploiting the (default) ordering by similarity.
In [11] they examine inter-linkages among seven leading crypto markets. In particular, they used an heatmap in order to visualize correlation levels over a time period, which get stronger as the colour become warmer. The heatmap produced in their work seems to focus the attention of the user only to strong correlation values (blank spot with no digits denotes insignificant correlation). Moreover they use a color coding on the values written inside each cell, leading to a challenging comparison between nearest cells with consequently difficulties in finding possible patterns. We found the solution to this problem using the color coding for each cell relying on the pop-up window that we discussed in section 4. It must be said, however that their type of work is not focused on data visualization and the amount of cryptocurrencies is much smaller.
Another paper that inspired us was [9]: they examine the top 100 cryptocurrencies ranging from 2015 to early 2018, showing in an heatmap the correlation of daily price returns between them. Even if their heatmap is just supporting a more broader work that is not based only on data visualization, it suggested that the approach that we used in our visualization can be effective and at the same time truly informative.

5.3 Line charts

In the signal processing litterature, one of the easiest way to compare and study couple of time series is comparing on the same domain both the trends. It was just a natural choice to lean towards such strategy. Also, the possibility to see the variation in real time through the zooming brush was a top notch possibility given by the d3.js library. In some papers [6] it is used without stating it explicitely how a good approach for visual time series analysis is to analyze them alligning their domains: d3 is giving us an even more powerful choice, and we found it just natural to walk this path.

5.4 Scatterplot

For the dimensionality reduction we decided to approach this visualization using simply a scatterplot, since it is the most common way to graph data with dimension reduced. PCA and MDS are very common techniques in unsupervised learning, and they are usually used to see how the data clusterize in a bidimensional(or sometimes tridimensional) space. The PCA is computed over the top 100 dataset and in particular using "market cap", "circulating supply" and "% change 24h" as attributes. The MDS is computed instead on a dissimilarity matrix based on Pearson Correlation Coefficient between time series of "volume" attribute, shifted to a common origin. What we decided to add with respect to the common scatterplot of PCA or MDS usually encountered in the accademic research field [8], was the interactivity we discussed in the previous sections. D3.js gave us an enormous chance to empower the data viz part, since many paper got implemented in other languages like python or C, where libraries for visualizations are not as powerful as the one opted for javascript.

5.5 Positioning of CryptoComparator

As we have indirectly implied through the whole report, few accademic works exist completely dedicated to comparison of cryptocurrencies oriented to the visual analytics field, based on their financial attributes. The only other benchmark we had access to was composed by other financial tool for cryptocurrency investiment. We want to stress how our tool is not thought a priori for investiments, but just for information retrival from financial data. We used, as comparison, different sites [2] [1] that are commonly used by traders. We truly believe to have built a tool with way more interactions than any other common tool for analysis. Also, the possibility to study the similarity between cryptocurrencies it gives the chance to investigate trasversally a single coin, and not only trying to do prediction on the single one blindly(i.e. through linear regression method or LSTM prediction models). The interactive scatterplot allow to identify and study through the click events possible outliers;

the linecharts allow the user to study specific time windows, for istance, in order to understand depression moments of the coin life. The elliptic graph give the possibility to see which elements are connected in terms of similarity on their volume attribute, something that the scatterplot alone does not allow to do: the clustering of the cryptocurrencies allows to understand only how they got grouped, but our similarity measure(Pearson similarity), does not rely on projection on lower dimensional spaces like it is in PCA for istance. In other words, it gives information on non time-dependant attributes connections like the volume of the 24h(daily), which is something that held an interpretation, in opposition to dimensionality reduction techniques for clustering that allows to see relations between elements, but usually based on single components that lose interpretability for the nature of the methods themselfs. To conclude, the heatmap cover the weaknesses of the graph representation: even though we can see the connection, we don't know how much they are weighted. The matrix gives the exact values to the users that are interested in learning the parametry lying behind.

References

[1] Crypto.com Coin price today, CRO live marketcap, chart, and info.

[2] Cryptocurrency Prices, Portfolio, Forum, Rankings.

[3] Elliptic - The Bitcoin Big Bang.

[4] Ahmed, S. N. *Physics and engineering of radiation detection*, 2nd edition ed. Elsevier, Waltham, MS, 2014.

[5] Burch, M., Kumar, A., Mueller, K., Brand, I., Castelijns, L., Ritchi, F., Rooks, F., Smeth, H., and Timmermans, N. Eye tracking for exploring visual communication differences.

[6] Fulcher, B. D., Little, M. A., and Jones, N. S. Highly comparative time-series analysis: the empirical structure of time series and their methods. *Journal of The Royal Society Interface 10*, 83 (June 2013), 20130048.

[7] Guo, L., Tao, Y., and Härdle, W. K. Understanding Latent Group Structure of Cryptocurrencies Market: A Dynamic Network Perspective. SSRN Scholarly Paper ID 3658206, Social Science Research Network, Rochester, NY, May 2018.

[8] Lever, J., Krzywinski, M., and Altman, N. Principal component analysis. *Nature Methods 14*, 7 (July 2017), 641–642.

[9] Liew, J., Li, R., Budavári, T., and Sharma, A. Cryptocurrency Investing Examined. *The Journal of the British Blockchain Association 2*, 2 (Nov. 2019), 1–12.

[10] Metsalu, T., and Vilo, J. ClustVis: a web tool for visualizing clustering of multivariate data using Principal Component Analysis and heatmap. *Nucleic Acids Research 43*, W1 (July 2015), W566–W570.

[11] Omane-Adjepong, M., and Alagidede, I. P. Multiresolution analysis and spillovers of major cryptocurrency markets. *Research in International Business and Finance 49* (Oct. 2019), 191–206.