UNIVERSITÀ DI PISA

RELAZIONE SOCIAL NETWORK ANALYSIS

BITCOIN OTC

Alunno: *Affolter Riccardo | Grassi Francesco | Messina Valeria*

Docente: *Pedreschi | Rossetti*

Anno Accademico: *2017/2018*

Sommario

DATA COLLECTION

La collezione dei dati, analizzata in questa ricerca, riguarda la sicurezza che intercorre tra i diversi membri della piattaforma Bitcoin Over The Counter (OTC). La collezione dati è stata importata dalla piattaforma "Analisi delle Reti" della Stanford University, SNAP (*Social Network Analytics Platform).* Poiché la rete Bitcoin consente il possesso ed il trasferimento anonimo della moneta virtuale, per i "Bitcoin *users"* è nata l'esigenza di giudicare il livello di affidabilità degli altri utenti, misurabile in una scala che va da -10 (*total distrust*) a +10 (*total trust*), in modo da prevenire transazioni che possano coinvolgere utenti fraudolenti. Il network si configura quindi come una rete *direct* e *weighted* poiché ogni legame tra nodi ha un peso assegnatogli. Dunque, il grafo analizzato rappresenta un *Weighted Signet Network* (WSN), in cui il grafo G è così costituito:

G= (V, E, W) dove l'insieme degli elementi di G rappresentano:

- V è l'insieme degli utenti;
- E ($E \subseteq V \times V$) è l'insieme degli *edge* (legami o links);
- W ($W : E \rightarrow [-10, +10]$) rappresenta il valore dei legami compreso tra -10 a +10 (escluso il valore zero).

Procedendo più nello specifico il nostro dataset è costituito da quattro parametri: Source, Target, Weight e Timestamp. Nel nostro caso, W (u, v) è pensato come l'assegnazione di un grado di "fiducia" che descrive quanto ogni user *S* (source) *"si fida"* dell' user *T* (target). Di seguito si riportano le statistiche dataset Bitcoin Over the Counter, con una breve descrizione dei parametri utilizzati.

Statistiche del dataset	
Nodi (V)	*5.881*
Edges (E)	*35.592*
Range di link pesati (W)	*[-10,+10]*

Source (S)	identificativo del *source,* votante
Target (T)	identificativo del *target,* votato
Weight (W)	peso dell'edge
Time (t)	tempo dell'avvenuta votazione

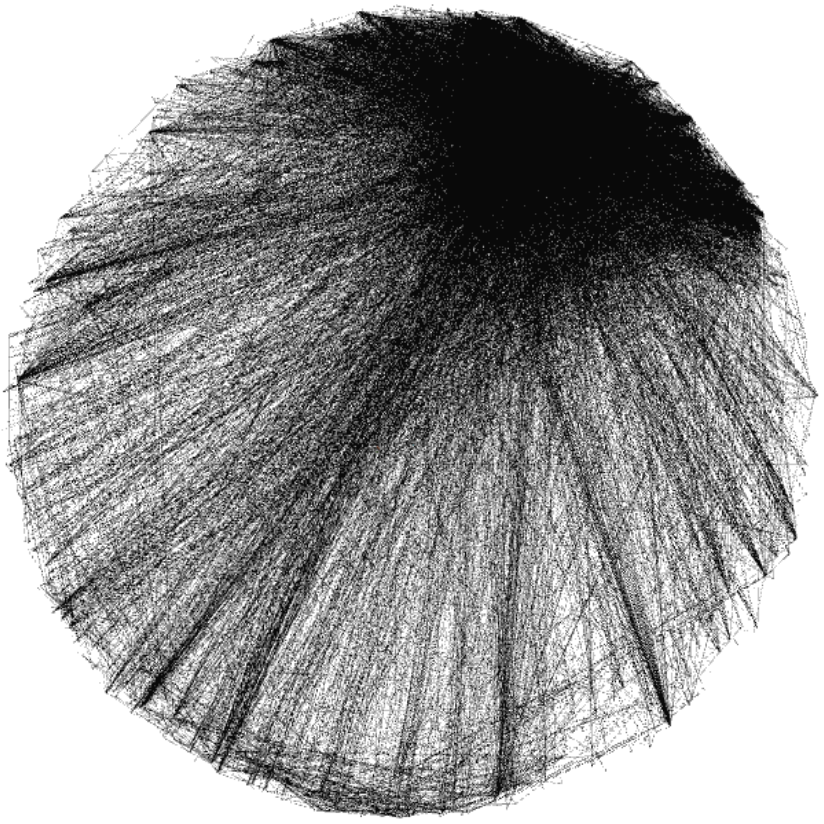


Figura 1: Network Bitcoin-OTC; layout Force Atlas (generato con Gephi)

i

NETWORK ANALYSIS

L'obiettivo di questo capitolo è quello di esporre le misure e le proprietà studiate durante il corso di Social Network Analysis, così da presentare una descrizione completa della rete Bitcoin OTC.
Nei paragrafi seguenti verrà spesso fatto riferimento ai modelli di rete di Barabasi ed Erdos Renyi, che diventeranno il nostro modello di confronto per ricercare somiglianze o meno tra le reti. Per sopperire a eventuali incomprensione abbiamo ritenuto opportuno descrivere brevemente i due modelli di rete sopracitati.

Il modello di **Erdos-Renyi**, denotato anche con G(n,p), descrive un particolare processo di generazione casuale dei grafi: il *random graph*. Un random graph è definito come un insieme di N nodi in cui ogni coppia di nodi è connesso con probabilità *p.*

Il modello di **Barabasi Albert** propone un network *free-scale* basato su due processi fondamentali:

1. un processo di crescita della rete per la continua aggiunta o rimozioni di nodi (processo che nel nostro confronto non è preso in considerazione poiché desunto come fisso nella rete Bitcoin OTC);
2. un processo denominato Preferential Attachment secondo il quale maggiore è il grado di un nodo, maggiore è la probabilità che un altro nodo stabilisca un legame con esso .

Al fine di un corretto confronto della rete Bitcoin OTC con BA e RE, abbiamo ipotizzato per i due modelli un numero di nodi N pari a quello della rete da noi scelta, ovvero 5881. Inoltre, per il modello di Erdos Renyi abbiamo scelto un valore arbitrario di *p* pari a 0.1.

Numero di nodi	5881
Numero di edge	1728756
Average degree	587.9123
Numero di self-loop	0
Densità	0.0999850782006392
Media	587.91225982
Mediana	588.0
Transitivity	0.09997158338357026
Average Clustering Coefficient	0.09997081152531356

Caratteristiche modello ER

Numero di nodi	5881
Numero di edge	17634
Average degree	5.9969
Numero di self-loop	0
Densità	0.0010198876353806274
Media	5.99693929604
Mediana	4.0
Transitivity	0.0041178044007714725
Average Clustering Coefficient	0.00799727492922933

Caratteristiche modello BA

1. Degree distribution analysis

Una rete può essere composta da una struttura semplice o estremamente complessa. Il primo passo, per comprendere ed analizzare la struttura di un network, è quindi quello di studiare come le diversi componenti della rete interagiscono tra loro. Poiché le componenti basilari che costituiscono e caratterizzano ogni network sono i nodi e i loro "legami", la nostra ricerca, in questa prima parte, si soffermerà sull'analisi di questi elementi.

Come introdotto precedentemente ogni struttura del grafo consiste in un insieme finito (e forse mutabile) di nodi, collegati tra loro attraverso delle interazioni che prendono il nome di "link". Questi ultimi possono configurarsi come delle relazioni orientate o meno, originando così un network rispettivamente directed o undirected.

Il nostro network è un grafo orientato (directed), caratterizzato dal fatto che ogni link (S, T) stabilisce un legame in cui l'user S esprime un'opinione riguardo l'utente T. Infatti, soffermando la nostra attenzione su ogni singolo nodo, vedremo come questi sono contraddistinti da alcuni archi in entrata, k_i^{in} (in-degree), ed altri in uscita, k_i^{out} (out-degree). Il grado totale del nodo, k_i , è ottenuto dalla somma degli in-degree e degli out-degree, ovvero:

$k_i = k_i^{in} + k_i^{out}$

L'analisi degli archi può essere affrontata anche su scala globale, prendendo in esame non il singolo nodo, ma l'insieme di tutti i 5881 nodi presenti nel grafo e il totale dei loro relativi edges, pari a 35592.

Utilizzando questi due valori è possibile calcolare una serie di ulteriori statistiche, descritte nelle tre tabelle sottostanti:

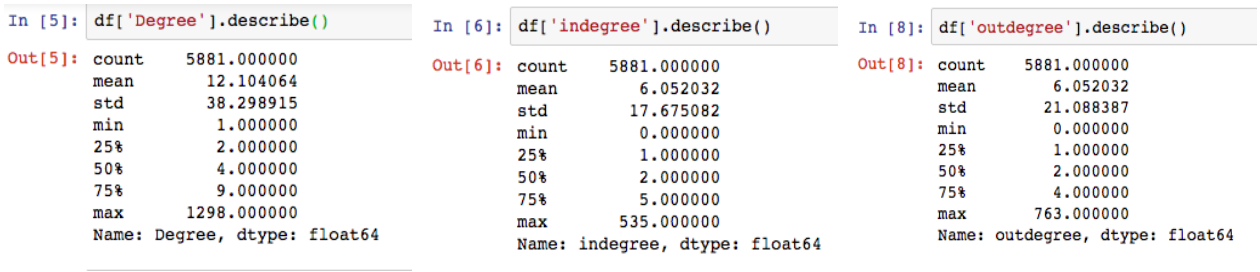


Figura 2: Degree, Indegree ed Outdegree

Si riporta di seguito la descrizione dei principali parametri che compaiono nella *Figura 2*:

-*Count*: numero dei nodi del grafo pari a 5881.

-*Mean*: valore del grado medio (average degree) posseduto da ogni nodo del grafo. Trattandosi di una rete diretta l'Average Degree è ottenuto dal rapporto fra gli *edge* e i nodi. Di seguito la relativa formula:

- AVG D (degree) = 2*E / N = 35592*2 / 5881 = 71184/5881 = 12.10406393
- AVG D (outdegree e indegree) = E/N = 35592 / 5881= 6.052031967

-*Std*: deviazione standard ovvero la misura di dispersione del valore di ogni *node degree* rispetto a quello della media calcolato nel mean.

Nella nostra analisi, data la congruenza tra i risultati ottenuti utilizzando archi orientati e archi non orientati, è stata adottata una rete non orientata nelle varie implementazioni su Python.

E' possibile ottenere informazioni molto importanti sulla struttura del grafo attraverso i soli dati che provengono dai nodi e dai loro gradi. In particolare, procedendo con il conteggio di quanti nodi ha ciascun grado otteniamo la *distribuzione dei gradi* (degree distribution). Quest'ultima misura ci permette di dare una rapida visione della struttura del network distinguendo i vari tipi di rete.

Di seguito si riporta la distribuzione dei gradi della rete Bitcoin OTC, su scala log-log, che ci permette di analizzare la funzione per un elevato valore di degree <k>. Infatti, dalla Tabella 3 è possibile osservare che solo pochi nodi presentano un elevato valore di <k>, questi nodi si chiamano *hubs,* mentre invece la maggior parte di nodi ha un valore di <k> più basso come nella distribuzione della *scale-free*. E' interessante notare come la distribuzione dei

gradi della nostra rete è quasi identica a quella del modello del Preferential Attachment di BA (tabella 4), dunque essa si distingue notevolmente da quella dal modello della Random di Erdos-Renyi.

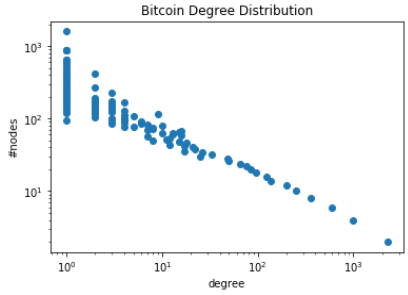


Figura 3: Distribuzione dei gradi in Bitcoin OTC

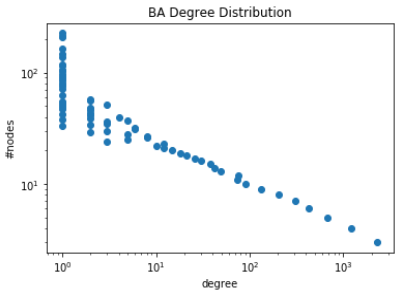


Figura 4: Distribuzione dei gradi in BA

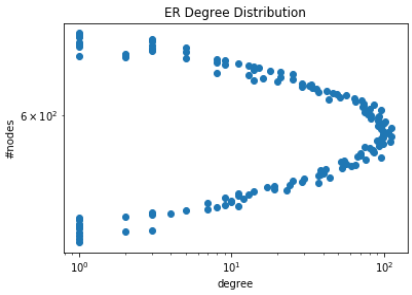


Figura 5: Distribuzione dei gradi in ER

Infine, a conclusione dell'analisi del network, si fa presente che nel grafo descritto non sono presenti self-loop, ovvero non esistono interazioni di un nodo con se stesso. L'assenza di self-loop comporta che la diagonale della matrice di adiacenza del grafo sarà costituita da tutti valori pari a zero: $A_{ii} = 0$, i = 1,..., *N.*

2. Connected components analysis

La nostra rete si presenta come una rete di tipo "assortativa", caratterizzata cioè da un'altra presenza di hubs. Il numero elevato di hubs permette che la rimozione di uno di questi non incida negativamente sul trasferimento dell'informazione tra i nodi.

Nel caso in cui la nostra rete fosse stata di tipo "disassortativa", questo sarebbe stato il caso contrario, cioè l'eliminazione di hub avrebbe provocato un disastro nella rete in quanto non ci sarebbero stati degli hub connessi ad altri hub, ma solamente hub utilizzati da ponte tra i nodi con poche connessioni.

La nostra rete presenta 4 componenti molto differenti tra loro per dimensione: uno di questi è il giant component mentre gli altri tre componenti sono formati da coppie di nodi. I quattro componenti sono visibili dall'ultimo attributo della *Tabella 1* e vengono elencati con identificatore 0, 1, 2, 3.

Id	Modulari...	In-De...	Out-D...	Degree	Component ID
6000	337	0	1	1	3
6002	337	1	0	1	3
3911	82	1	1	2	2
3912	82	1	1	2	2
3762	79	1	1	2	1
3763	79	1	1	2	1
6	90	44	40	84	0
2	90	41	45	86	0
5	90	3	3	6	0

Tabella 1: Connected Components

3. Path analysis

Nei network la distanza fisica è misurata da particolari "percorsi", ognuno dei quali si configura come una sequenza di nodi tali che ogni nodo, appartenente al percorso, è connesso al successivo tramite un link. Questi percorsi prendono il nome di "path" e la loro lunghezza è misurata dai *link* che essi contengono.

Il diametro <dmax> è la distanza minima che intercorre tra i due nodi più lontani. Nel nostro grafo la distanza massima ha valore di 11. Questo dato è significativo in quanto ci permette di determinare che esiste un cammino di 11 collegamenti per passare da un nodo ad un altro nella rete.

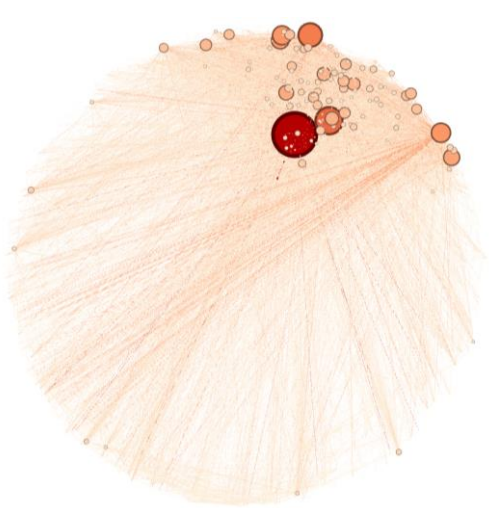


Figura 6: Hubs

Per studiare le proprietà della rete, è molto importante lo studio della lunghezza media del percorso *(Average Path length)* del grafo, ovvero la distanza media fra tutti i percorsi minimi *(shortest path lenght)* di tutte le copie dei nodi. Attraverso le statistiche, analizzate dal Graph Distance Report di Gephi, il risultato dell' Average Path length del network Bitcoin-OTC è di 3.7189130700273005. Tale risultato indica che, in media, per collegare una qualsiasi coppia di nodi è necessario compiere circa 4 step, dunque con pochi passi è possibile raggiungere qualsiasi nodo della rete.

La *Tabella 2* illustra i cambiamenti che la rete subirebbe se fosse soggetta ad un attacco mirato all'eliminazione dei suoi hub. Il criterio per la scelta degli hub da eliminare si basa sul grado del nodo, ovvero, si è partiti dall'eliminazione dell'hub con il grado massimo per poi eliminare i successivi. Già dopo la prima eliminazione è possibile osservare dei cambiamenti significativi nel numero delle componenti, i quali sono aumentati da 4 componenti a 308 (numero che continuerà a crescere, anche se meno velocemente, col

procedere delle eliminazioni). Tale disgregazione esplosiva è indice del fatto che la rete Bitcoin-OTC è una rete scale-free, ovvero, tutti i nodi di grado piccolo preferiscono stabilire un link con un hub. A rafforzare l'anzidetta affermazione il fatto che l'Average Path Length non subisce grosse variazioni, neanche dopo l'eliminazione del quarto hub.

Hub rimossi	Average Degree	Average Weighted Degree	Connected Components	Average Clustering Coefficient	Average Path Length
0	6,052	6,125	4	0,149	3,719
1	5,832	5,804	308	0,142	3,799
2	5,694	5,498	373	0,134	3,836
3	5,574	5,62	409	0,128	3,865
4	5,477	5,718	528	0,125	3,888

Tabella 2: Analisi della rete in seguito alla rimozione di hubs

4. Clustering Coefficient, Density analysis

Il coefficiente di clustering misura quanto i nodi di un grafo tendono ad essere connessi fra loro. Il coefficiente di clustering di un network formalizza la nozione di Average Clustering Coefficient, ottenuto dalla media dei Local Clustering (C_i) fra tutti i nodi del network $i = 1,...,N$. L' Average Clustering Coefficient (C) nel nostro network è di 0.149.

Studiando la distribuzione del coefficiente di clustering è possibile osservare la densità locale di una rete: un'elevata connessione fra i nodi vicini comporterà un elevato coefficiente di clustering in ciascuno dei nodi presi in esame.

Dalla tabella sottostante si legge che 3267 nodi della nostra rete, ovvero il 55%, hanno un valore di Cluster Coefficient valore pari a 0, il che significa che tali nodi hanno una probabilità molto bassa di connessione. Il caso opposto è descritto dai nodi il cui valore è pari a 1, 285 nodi (0.048%), ovvero quei nodi connessi con tutti gli altri del loro component.

```
In [15]: df['clustering'].describe()

Out[15]: count    5881.000000
         mean        0.148747
         std         0.255537
         min         0.000000
         25%         0.000000
         50%         0.000000
         75%         0.200000
         max         1.000000
         Name: clustering, dtype: float64
```

Figura 7

```
Nodi con Cluster Coefficient = 0:  3267

Nodi con Cluster Coefficient <= 0.2:  4470

Nodi con Cluster Coefficient >= 0.2:  1471

Nodi con Cluster Coefficient = 1:  285
```

Figura 8

La densità di un grafo è la misura del rapporto tra gli archi e il massimo numero di archi possibili tra i nodi dello stesso grafo. Il grafo della nostra rete presenta una densità di 0.001, come riscontrata anche in Barabasi Albert. Tale valore, che compare nel risultato del Graph Density Report del software Gephi, può anche essere semplicemente calcolato come rapporto tra il numero di links presenti e il potenziale valore massimo che questi assumerebbero nel grafo.

*density = L / Lmax dove Lmax = (N*N-1)*

Quindi, poiché nel nostro grafo Lmax = 5581*5880= 34580280, possiamo concludere che la densità = 35592/34580280 = 0.00124302058861293.

5. Centrality analysis

L'analisi finora fatta, solleva una domanda: esistono utenti che hanno effettuato o hanno ricevuto molte votazioni? E se questa risposta è affermativa, hanno questi un ruolo centrale? Per rispondere a queste domande, misuriamo determinati valori correlati all'importanza del nodo, ovvero alla sulla centralità nel grafo.

L'analisi della centralità si compone nello studio di tre principali parametri:

- ***degree centrality***: calcola il numero di connessioni di un determinato nodo della rete.
- ***betweenness centrality*** : misura quante volte quel nodo viene attraversato nei vari percorsi minimi dei nodi.
- ***closeness centrality*** : misura la distanza media che intercorre tra un nodo e tutti gli altri nella rete. Meno si è distanti da tutti gli altri nodi più si è centrali e viceversa.

Degree Centrality

La tecnica più comune e semplice per trovare il nodo più centrale è misurare il grado di ogni nodo della rete e confrontare i risultati ottenuti. Il grado di un nodo determina con

quanti altri nodi questo sia connesso. Di conseguenza, i nodi con un alto grado di degree, possono essere considerati come dei potenziali "influencer", poiché capaci di diffondere informazioni più velocemente grazie alle loro numerose connessioni. Per misurare la degree centrality, i nodi sono stati ordinati con un criterio di ordinamento decrescente sui loro rispettivi gradi.

Bitcoin OTC			**Erdos-Renyi**			**Barabasi Albert**		
Nodo	*Gradi*	*DC*	*Nodo*	*Gradi*	*DC*	*Nodo*	*Gradi*	*DC*
35	795	0.135204	**1717**	669	0.113776	**2**	266	0.045238
1810	439	0.074660	**2045**	669	0.113776	**7**	238	0.040476
2642	438	0.074490	**1593**	667	0.113435	**5**	205	0.034864
Mean		6.05	**Mean**		588	**Mean**		5.99

Tabella 3: Confronto Degree Centrality nelle tre reti: BItcoin OTC, ER, BA.

La *Tabella 3* mostra i dati della Degree Centrality della nostra rete confrontati con i gradi delle altre due tipologie di grafo. Il risultato ottenuto, attraverso tale confronto, è che i gradi dei grafi random sono molto simili tra di loro, mentre nel nostro grafo è possibile notare che il nodo 35 è caratterizzato da un grande scostamento di valore rispetto al secondo nodo più centrale. Tale scostamento è indice del fatto che il nodo 35 investe un ruolo fondamentale all'interno della rete Bitcoin OTC.

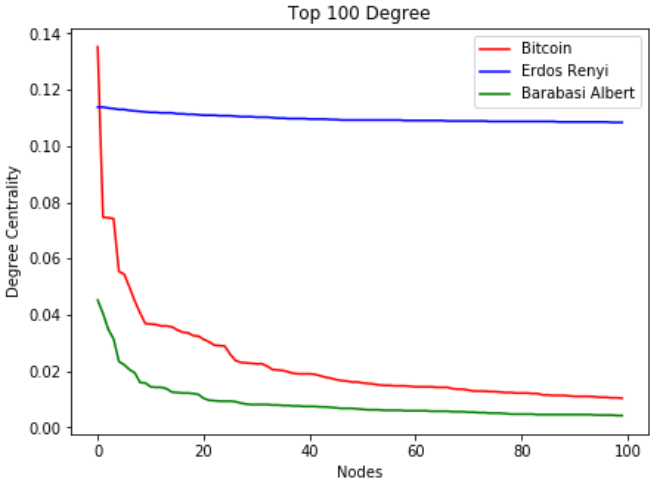


Figura 9

Betweenness Centrality:

Tra tutti i percorsi possibili all'interno di un grafo, il più interessante e significativo è quello minimo, cioè quel cammino per cui il numero di edges,

da utilizzare per andare da un nodo ad un altro, è il più piccolo possibile. Per fare ciò, abbiamo bisogno della centralità, cioè un tipo di metrica che attribuisce una misura di distanza tra nodi o aree delle rete. Attraverso tale misura è possibile, per esempio, osservare il grado di forza di un nodo in un grafo.

Bitcoin OTC		**Erdos-Renyi**		**Barabasi-Albert**	
Nodo	*BC*	*Nodo*	*BC*	*Nodo*	*BC*
35	0.193143	**5730**	0.000196	**3**	0.123629
2125	0.077739	**4149**	0.000195	**1**	0.117080
2642	0.064724	**5459**	0.000195	**2**	0.104766
Mean	0.0004	**Mean**	0.000153	**Mean**	0.0005

Tabella 4: Confronto Betweenness Centrality nelle tre reti: BItcoin OTC, ER, BA.

La betweenness è molto utile quando nella rete sono presenti cluster tra loro isolati, ma comunque collegati attraverso "bridge". Utilizzando la stessa tecnica usata per la Degree Centrality, otteniamo i risultati riportati nella *Tabella 4* (i dati sono stati normalizzati per ridurre i tempi di calcolo). Il confronto di tali risultati con quelli della degree centrality non portano a grosse rilevazioni, se non un'ulteriore conferma, in termini di centralità, del ruolo di importanza del nodo 35 (molto alto rispetto al secondo).

Closeness Centrality:

Un altro tipo di metrica, che potremmo utilizzare come misura, è la vicinanza, cioè quel nodo che, dal punto di vista statistico, è mediamente più vicino agli altri nodi. Detto in altri termini, è il nodo che verrà raggiunto più velocemente nel caso in cui si verificasse un broadcast dell'informazione.

Bitcoin OTC		**Erdos-Renyi**		**Barabasi-Albert**	
Nodo	*CC*	*Nodo*	*CC*	*Nodo*	*CC*
905.0	0.436024	**5730**	0.530016	**3**	0.386486

1.0	0.433031	**3625**	0.529968	**1**	0.384188
35.0	0.427479	**5459**	0.529921	**2**	0.379379
Mean	0.28	**Mean**	0.52	**Mean**	0.24

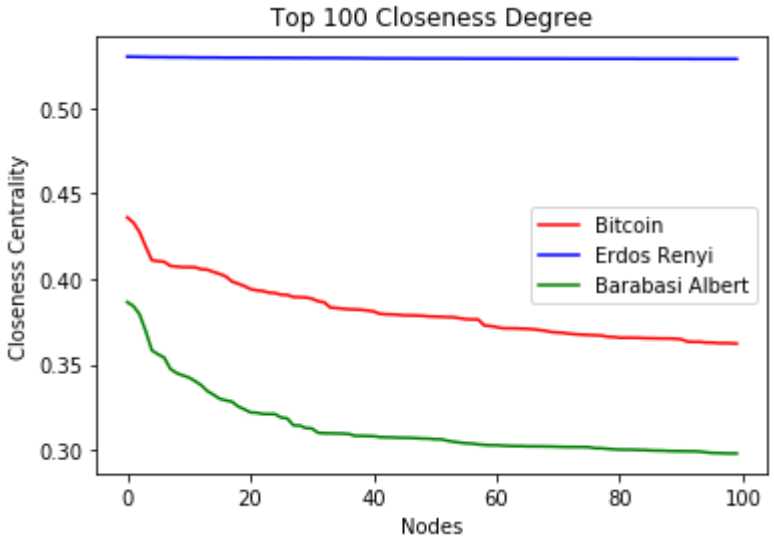


Figura 10

Tabella 5: Confronto Closeness Centrality nelle tre reti: BItcoin OTC, ER, BA.

Dall'immagine in Figura 10 si evince una somiglianza dell'andamento della closeness centrality tra la nostra rete reale con quella Barabasi Albert anche se con un intervallo della closeness centrality più alto. L'hub 35 è tra le prime 3. La funzione della rete random invece, non segue un andamento né crescente né decrescente.

ANALYTICAL TASKS :

1. Community Discovery:

I grafi di una rete possono essere costituiti da diverse *communities*. Ogni community di una rete forma un "sottografo" (*subgraph)*, all'interno del quale i nodi risultano essere densamente connessi fra loro.

Prima di procedere alla discovery abbiamo deciso di eliminare le 3 componenti composte da soli 2 nodi, perchè ritenute superflue nell'analisi della comunità.

NetworkX ha diversi meccanismi per computare le community:

- La funzione ***nx.transitivity*** usa la funzione nx.triangles per ricavare il rapporto tra il numero di triangoli e tutti i possibili triangoli. Nel nostro caso il valore ottenuto è: *0.05923*

  Abbiamo anche calcolato il valore dei triangoli presenti nella rete: *66986*

- Un valore che mi identifica la tradic closure è il clustering coefficient, esso può essere ricavata tramite la funzione ***nx.average_clustering*** che lo ricava per ogni nodo e successivamente ne viene calcolata la media così da calcolare il clustering coefficient medio pari a: *0.17750*

Un tasso di transitività e di coefficiente medio di clustering alto comporta un effetto tipico del small-world. La small-world è un network che si mostra come un network casuale, ma ha un alto tasso di transitività e una lunghezza media del percorso (average path length) breve. Tale caratteristica è molto comune nelle strutture delle reti sociali, perché le reti moderne hanno forti connessioni e perciò una alta transitività, viceversa i grafi random presentano un basso coefficiente di clusterizzazione. Il nostro valore comparato al caso di BA(0.009) è maggiore.

Le communities possono essere identificate da particolari algoritmi capaci di valutare le diversi partizioni della rete. In questo elaborato utilizzeremo tre diversi algoritmi per l'individuazione di community: DEMON e Label Propagation, German-Newman e Louvain. In seguito le abbiamo confrontate usando questi quattro valori:

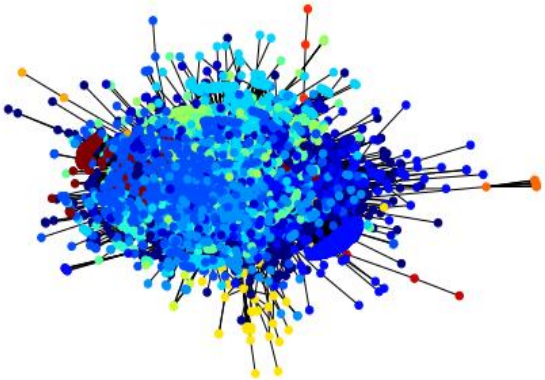
• Internal Edge Density: una partizione X sarà migliore, in termini di IED, di un'altra partizione Y se la peggiore densità interna di link di X sarà migliore della peggiore IED di Y.

• Average Internal Degree: è una funzione il cui obiettivo è quello di mostrare se una partizione X è migliore di un'altra Y. Ciò avviene controllando se il peggior grado medio dei nodi di X è migliore rispetto al peggiore AND di Y.

• Conductance: parliamo di conduttanza quando si fa riferimento alla probabilità che un cammino casuale esca dalla comunità. In tali termini, parleremo di migliore partizione quando il peggior valore di conduttanza di X (che è causa di un elevato indice di conduttanz) detiene un valore di conduttanza più basso della peggiore conduttanza di Y.

• Modularity: è una funzione il cui obiettivo è quello di valutare una qualsiasi partizione considerando la densità interna di ciascuna comunità che appartiene alla stessa partizione. Dentro le comunità la densità è elevata, ma se paragonate dovrebbero contare pochi collegamenti e, di conseguenza, una densità inferiore.

Louvain:



Il metodo Louvain è un metodo semplice, efficiente e di facile implementazione per l'identificazione di comunità in reti di grandi dimensioni, il cui obiettivo è quello di ottimizzare la "modularità" di una

partizione della rete. Tale ottimizzazione viene eseguita in due passaggi: in primo luogo, il metodo cerca comunità "piccole", ottimizzando la modularità localmente; in secondo luogo, aggrega nodi appartenenti ***Figura 11***

alla stessa comunità e costruisce una nuova rete. Questi due passaggi vengono ripetuti iterativamente finché la massima modularità non viene raggiunta. Tale algoritmo di aggregazione definisce la costruzione di poche comunità, ma di buone dimensioni.

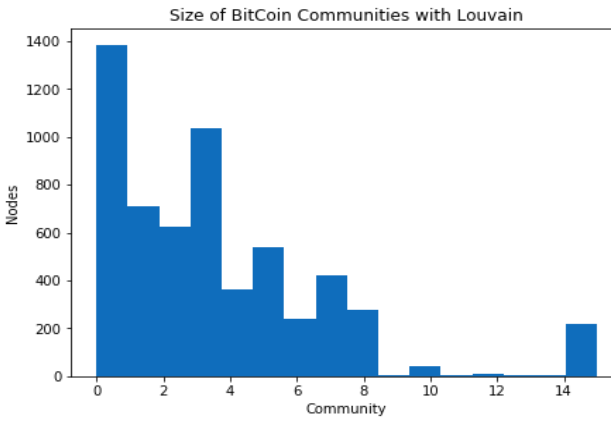


Figura 12

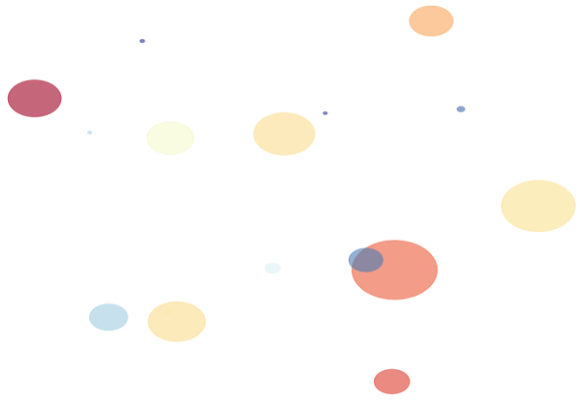


Figura 13

All'interno della rete Bitcoin-OTC sono state individuate 16 comunità, come osserbaile in *Figura 12* e in *Figura 13,* tra le quali due superiori a 1000 nodi. Tale rete mostra una modularity di 0.48828470781886074. Un approccio per visualizzare la grandezza delle community Bitcoin potrebbe essere quello di utilizzare il bubble chart, come mostrato in Figura 6: l'area di ogni cerchio rappresenta la grandezza di ciascuna community.

Valutiamo di seguito le misure ottenute:

AND	IED	Modularity	Conductance
1.333333	0.000941	0.488177	0.564593

Tabella 6

Girvan–Newman:

L'algoritmo di Girvan-Newman ha come obiettivo quello di scoprire le community rimuovendo iterativamente gli edge con betweenness centrality maggiore, suddividendo il grafo in sotto community ad ogni step. Il metodo di Girvan-Newman è stato implementato ma, per motivi di tempi computazionali, non è stato considerato.

K-clique:

Il K-Clique è uno degli algoritmi per l'individuazione delle componenti attraverso l'uso di cricche; le k-cricche sono quelle cricche che contengono k nodi, i quali sono connessi con k-1 nodi che appartengono alla stessa cricca. Tale algoritmo è molto utile per l'individuazione dell'overlap. Applicando tale algoritmo all'interno della nostra rete, abbiamo iterato il metodo implementato nella libreria networkX, con un $k \in 3,4,5,6$. La *Tabella 7* ospita i valori delle nostre metriche (le community emerse possono essere messe a confronto utilizzando per ogni k il valore massimo di AND, IED e Modularity e il valore minimo della Conductance):

K	AND	IED	Modularity	Conductance
3 (75 comunità)	2	0.001	-0.208677	0.992500
4 (36 comunità)	3	0.004	0.175199	0.986379
5 (26 comunità)	5.698	0.011	0.115777	0.383756
6 (13 comunità)	7.728	0.019	0.091634	0.460462

Tabella 7

Label Propagation:

Attraverso tale algoritmo iterativo ogni nodo, inizialmente, possiede un'etichetta unica e, ad ogni step, cambia assumendo quella di un suo nodo vicino con probabilità *p.* Nelle successive iterazioni, ogni nodo utilizzerà l'etichetta più utilizzata nel suo vicinato (il nodo può essere assegnato a più comunità). Abbiamo ottenuto 93 comunità dove i valori di misura sono osservabili dalla *Tabella 8*:

	min	max	avg	std
AND	1.000000	18.500000	1.647299	1.984619
IED	0.000332	0.250000	0.194272	0.065266
Modularity	0.067218			

Conductance	0.013744	0.500000	0.372359	0.101711

Tabella 8

Demon:

L'algoritmo Demon ha come obiettivo quello di estrarre comunità trovate grazie alla propagazione di una determinata proprietà.

Quindi, per ogni nodo n, l'algoritmo compie i seguenti passaggi:

1. Estrarre l'Ego Network di n
2. Rimuovere n dall'Ego Network
3. Eseguire la Label Propagation
4. Inserire n in ogni comunità trovata
5. Aggiornare il set di comunità C

Abbiamo deciso per ogni partizione di comunità di variare il valore di ε, il quale rappresenta la soglia minima per unire le comunità. Se l'ε% di quella più piccola non è incluso nella comunità più grande allora le due comunità verranno unite nell'insieme finale. Ammettono overlapping.

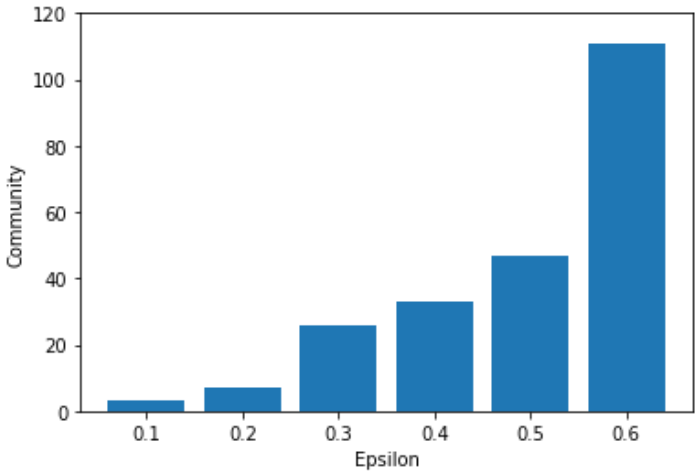


Figura 14

Nel nostro caso il valore di *ε* varia da 0.1 a 0.6, come osservabile in *Figura 14,* con le grandezze delle componenti che variano da 3 a 111.
Al variare di ε, determiniamo la partizione migliore visualizzando le misure dalla *Tabella 9*:

ε	AND	IED	Modularity	Conductance
0.1 (3 comunità)	6.295	0.001305	-0.33051	0.606
0.2 (7 comunità)	12.227	0.001722	-0.3058	0.606
0.3 (26 comunità)	9.256	0.002531	-0.13598	0.638
0.4 (33 comunità)	8.638	0.002202	-0.108635	0.663
0.5 (47 comunità)	7.501	0.002051	-0.235181	0.717

0.6 (105 comunità)	7.489	0.002277	-0.069906	0.799

Tabella 9

La partizioni migliori sono, guardando i dati, la comunità con **ε = 0.2** per conductance e per AND, invece con **ε = 0.6** ho modularità migliore (non ho comunque una buona suddivisione delle reti con valori di modularità bassi). Per Internal Edge Density la partizione migliore ha **ε = 0.3.**

2. Tie Strength:

Nelle reti sociali i collegamenti tra i nodi possono essere divisi in due tipi:

- Strong Ties
- Weak Ties

Per la valutazione delle strong ties, all'interno del nostro grafo, è stato calcolato l'overlap ratio. Definiamo il valore dell'overlap di un edge tra i e j come il rapporto tra il numero di vicini di entrambi e il numero di vicini che sono di almeno o di i o di j. La misura di questa strength è dunque:

$$Oij = \frac{N(i) \cap N(j)}{N(i) \cup N(j)}$$

Per vedere la forza dei legami abbiamo rimosso, in ordine decrescente e crescente, ogni edge che aveva, in ordine, il legame forte o debole secondo il coefficiente.

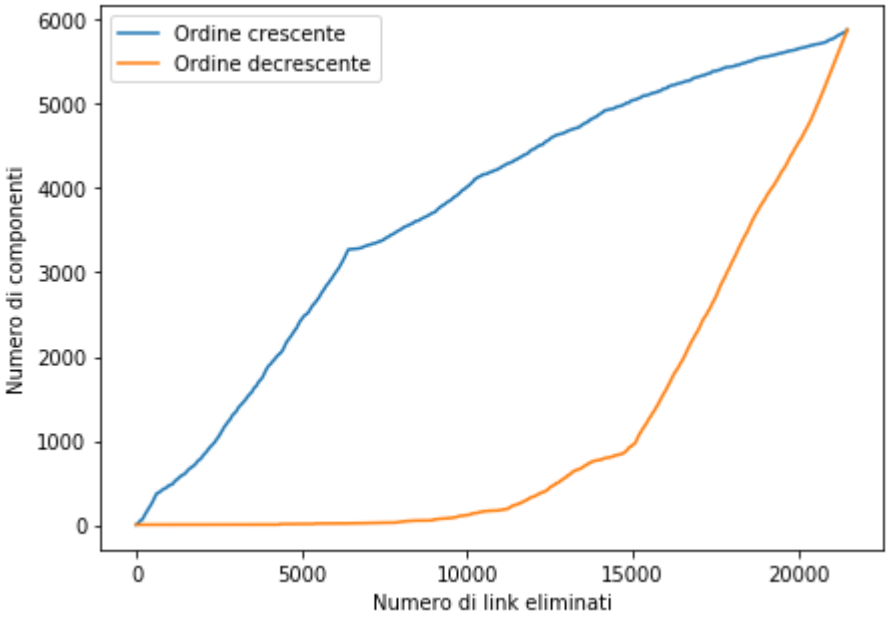


Figura 15

Come mostrato dalla *Figura 15*, la rimozione degli edge con un legame forte tende a bloccare il numero di componenti a 1. Dopo la rimozione del 10 millesimo link, la rete inizia a spezzarsi. Procedendo in ordine inverso, dopo lo stesso numero di link rimossi, la rete mostra un numero di componenti di circa 4000.

3. Spreading:

Le caratteristiche con cui una rete si configura al suo interno possono determinare la diffusione più o meno veloce di "epidemie". Condizione necessaria e sufficiente affinché si verifichi uno *spreading* epidemico è infatti la presenza di uno scenario *a rete* dove i collegamenti fra gli utenti diventano i canali per la diffusione dell'epidemia. In particolare, sono stati teorizzati tre modelli per la diffusione di patologie (SI, SIS, SIR), la cui differenza è caratterizzata dallo stato in cui si trova l'utente coinvolto, ovvero:

-Susceptible (S): individuo sano che non ha ancora contratto il patogeno;

-Infected (I): individuo infetto che ha contratto il patogeno e che può infettare altri nodi;

-Recovered (R): individui guariti che hanno debellato il patogeno e che non sono contagiosi.

I modelli SI, SIS e SIR, descritti di seguito, ci permettono di comprendere in che modo il patogeno viene diffuso nella rete.

- **Modello Susceptible-Infected (SI)**

Per esemplificare questo modello si consideri la diffusione di un'epidemia in una popolazione con N individui, dove S(t) è il numero di individui che sono suscettibili (sani) al tempo t e I(t) è il numero di individui che sono infetti. Si assuma che un individuo abbia <k> contatti e che la probabilità che il patogeno venga trasmesso da un individuo sano a uno suscettibile in una unità di tempo è β. se I(t) individui infetti stanno trasmettendo il patogeno, ognuno con probabilità β, il numero medio di nuove infezioni dI(t) durante un intervallo di tempo dt è:

$$dI\ (t)\ = \beta\langle k\rangle \frac{S(t)\ *\ I(t)}{N}$$

Al fine di verificare come cambia, nel nostro network, la diffusione di patogeni da soggetti *susceptible* a soggetti *infected* al variare di β, sono stati applicati a questa variabile tre diversi valori: β=0.01, β= 0.05 e β=0.1.

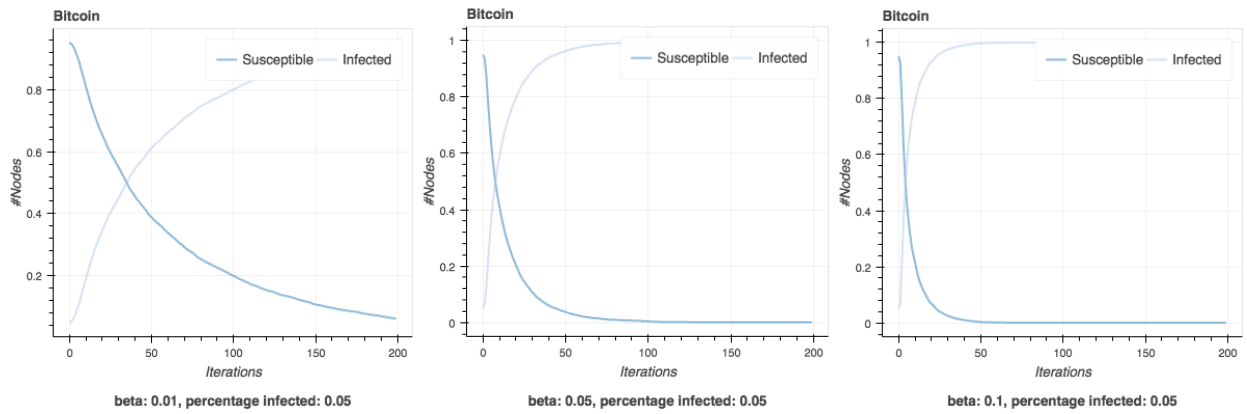


Figura 16: Confronto modello SI al variare di β

Nella tabella 16 si nota come all'aumentare di βsi assiste ad un aumento della velocità di trasmissione dell'infezione, infatti per βpari a 0.1 il numero di iterazioni necessario per contagiare N nodi è inferiore rispetto a quello che si ha per β= 0.01. Per semplicità, utilizzeremo un unico valore per il parametro β, ovvero quello di 0.05 in quanto valore intermedio fra i tre prima considerati. Segue un confronto del modello SI della rete Bitcoin-OTC con quello della rete Barabasi Albert e della rete Erdos-Renyi.

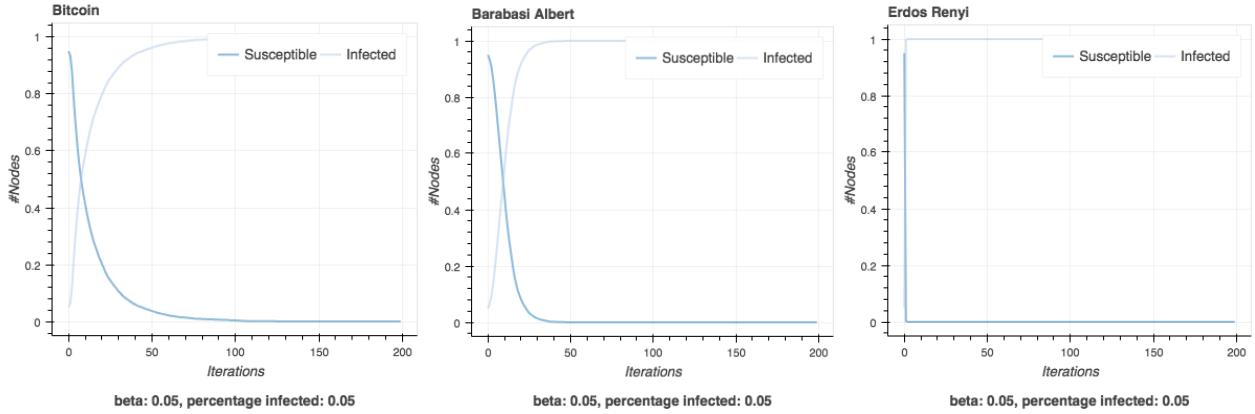


Figura 17

I primi due grafici mostrano nell'istante 0 una composizione di soli individui suscettibili e di nessuno infetto. L'andamento delle curve tendono poi ad una crescita vertiginosa degli individui infetti e, di conseguenza, ad un calo altrettanto vertiginoso degli individui suscettibili. A seguire, si assiste ad un rallentamento delle stesse fino al contagio completo di tutti gli individui. La rete random ha una crescita degli individui influenzati istantanea e ciò è dovuto al fatto che tutti gli individui tendono a comunicare con tutti.

- **Modello Susceptible-Infected-Susceptible (SIS)**

Questo modello, seppur costituito dagli stessi due "stati" affrontati precedentemente, si caratterizza per la possibilità, da parte dei soggetti infetti, di poter guarire dallo stato di malattia. Più precisamente, in questo modello gli individui guariscono secondo un tasso fisso μ, che li rende nuovamente *susceptible.* L'equazione che descrive le dinamiche di questo modello è la seguente:

$$\frac{di}{dt} = \beta < k > i\,(1 - i) - \mu i$$

A differenza del modello SI in cui al tempo t->∞ tutti vengono infettati, nel modello SIS l'epidemia ha due possibili esiti:

- **Endemic State (stato endemico)** ($\mu < \beta\langle k\rangle$): per un basso valore della *recovery rate*, la frazione di infivisui i, segue una curva simile a quella osservata nel modello SI. Questo comporta che in ogni momento, solo una parte della popolazione diventa infetta. In questo stato endemico il numero di nuovi individui infetti eguaglia quello di individui che sconfiggono la malattia, per cui la frazione di popolazione infetta non cambia col tempo.
- **Disease-free State (stato senza malattia)** ($\mu > \beta\langle k\rangle$): per un alto valore della *recovery rate,* il valore *i* diminuisce esponenzialmente col tempo e ciò dipende dal fatto che in questo stato il numero di individui curati per unità di tempo eccede quello di nuovi individui infetti. Quindi col tempo il patogeno scomparirà dalla popolazione.

Di seguito si riportano i grafici del modello SIS applicato sulla rispettivamente sulla rete Bitcoin, BA e random.

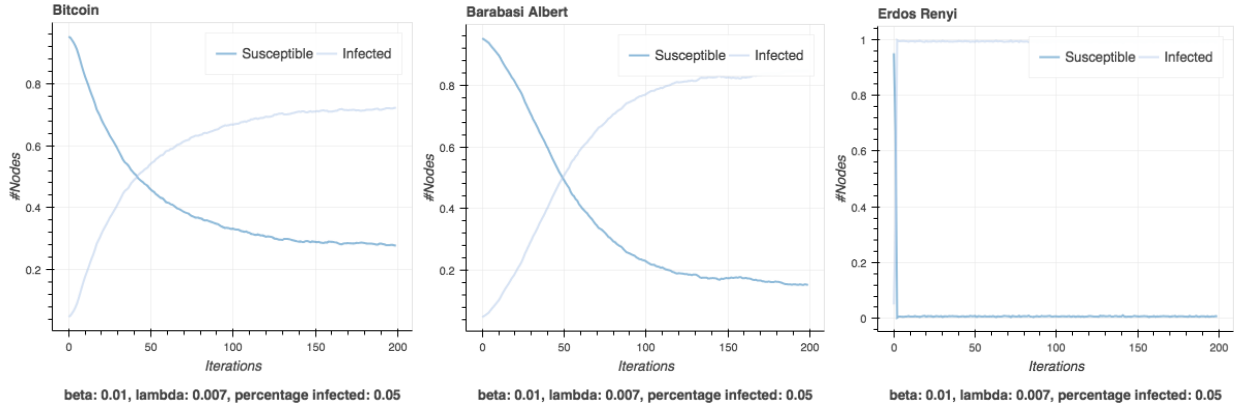


Figura 18

Esattamente come per il modello SI, anche in questo caso le curve dei primi due grafici hanno un andamento molto simile, con la differenza che solo nella rete Bitcoin-OTC le due curve tendono a rallentare molto di più dopo la loro collisione. Anche in questo modello, le curve della rete random hanno una crescita e una decrescita istantanea non appena superata l'istanza 0.

- **Modello Susceptible-Infected-Recovered (SIR)**

In questo modello, gli individui sviluppano un'immunità dopo la guarigione dall'infezione, entrando quindi in una fase ultima chiamata *recovered*, attraverso la quale non possono né essere infettati né infettare altri individui.

Lo schema dell'andamento è: S → I → R.

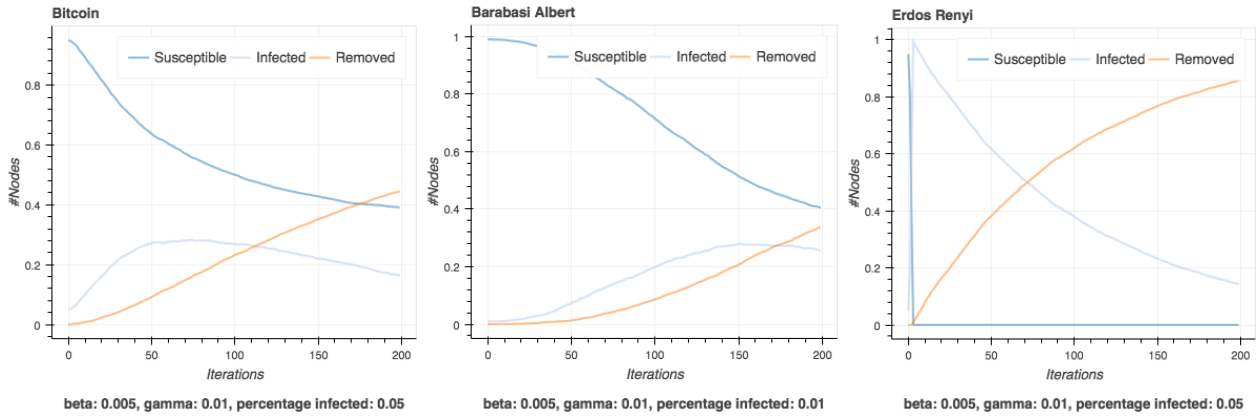


Figura 19

I risultati ottenuti sulla rete Bitcoin-OTC ci permettono di vedere come la quantità di individui che si infettano sia di lenta crescita fino alla cinquantesima iterazione per poi incontrare una fase di calo. Infatti, questa analisi ci permette di prevedere che per t→ ∞ il numero di infetti tenderà a zero e ciò è determinato dalla continua crescita di *recovered.* Tale caratteristica mostra l'ennesima somiglianza con la rete Barabasi Albert, con l'unica eccezione sul numero di iterazione in cui avvengono le fasi di calo (150 nella rete BA). La rete random si comporta in maniera univoca anche in quest'ultimo modello, dove la curva dei soggetti suscettibili si abbatte non appena viene superata l'istanza, così come la curva dei soggetti infetti. Quest'ultima però mostra poi un andamento inverso con la curva dei soggetti rimossi.

Dai grafici riportati si nota come i tre modelli SI, SIS e SIR sono congruenti nella prima parte dello stato dell'epidemia: quando il numero di individui infetti è basso la malattia si diffonde rapidamente avendo una crescita quasi esponenziale. Gli effetti differiscono invece sul *long-term*: nel modello SI tutti i nodi diverranno infetti; nel modello SIS si può raggiungere sia uno stato endemico in cui solo una parte di individui rimarrà per sempre infetta o l'infezione scomparirà; nel modello SIR ogni individuo guarirà alla fine.

CURIOSITY DRIVEN

La scelta di aggiungere quest'ultimo punto nasce da una curiosità sorta riguardo i pesi dei nodi, quindi dai feedback rilasciati dagli utenti della rete presa in analisi.

In particolare, ci siamo domandati quanto incidesse la presenza dei feedback negativi e come sarebbe la rete se questi venissero eliminati.

Per rispondere a questa domanda, abbiamo deciso di utilizzare l'algoritmo Louvain, sia per le partizioni negative che per quelle positive.

Sono stati quindi creati due grafi, dividendo quelli positivi da quelli negativi. Nel grafo positivo (*Figura 20*) le 4 componenti della rete si sono frammentate in 39 (*Figura 21*), mostrando però sono in presenza di una componente (quella blu) che può essere dichiarata giant component (3200 nodi). La presenza di pochi hub, tra i nodi con peso negativo, ha sicuramente influito a far si che la rete rimanesse abbastanza compatta. Infatti, nell'istogramma in Figura …, è possibile vedere come le componenti formatesi siano di dimensioni molto piccole, ad eccezione delle prime 7, che contano il maggior numero di nodi totali.

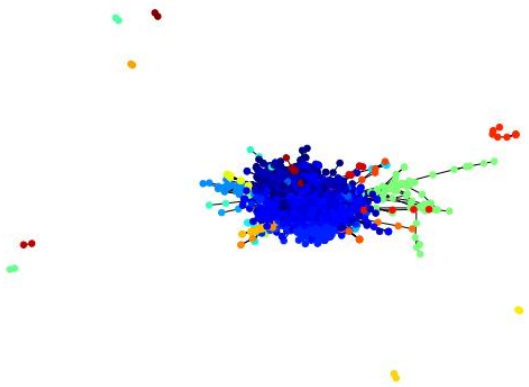


Figura 20

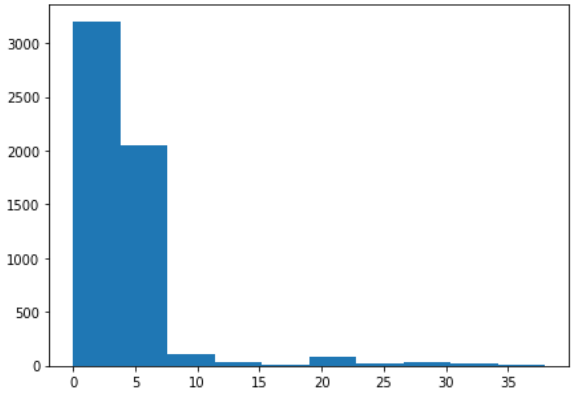


Figura 21

Per quanto riguarda il grafo con soli nodi con peso negativo (*Figura 22*) il numero delle componenti si è frammentato in 87 partizioni (*Figura 23*), diverse delle quali, come mostrato nelle figure … e …, in presenza di pochi nodi al loro interno.

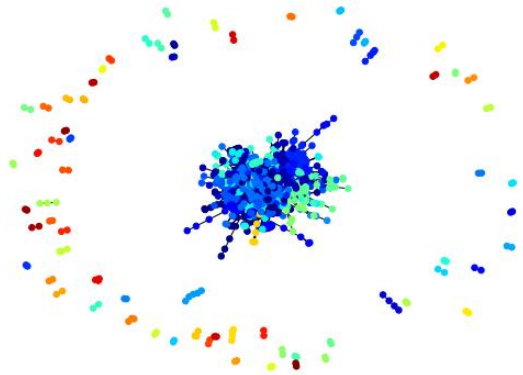


Figura 22

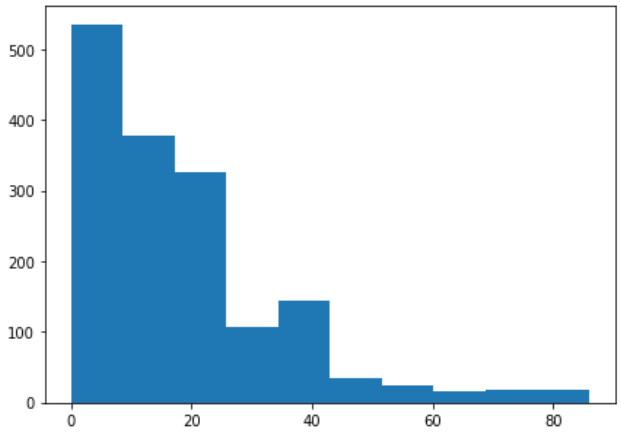


Figura 23

Contrariamente a quanto successo nella prima partizione, l'eliminazione dei nodi con peso positivo ha comportato l'eliminazione di molti hub, tra cui il 35, il cui ruolo di centralità è stato riscontrato sia nella degree centrality, nella betweenness centrality e nella closeness centrality.

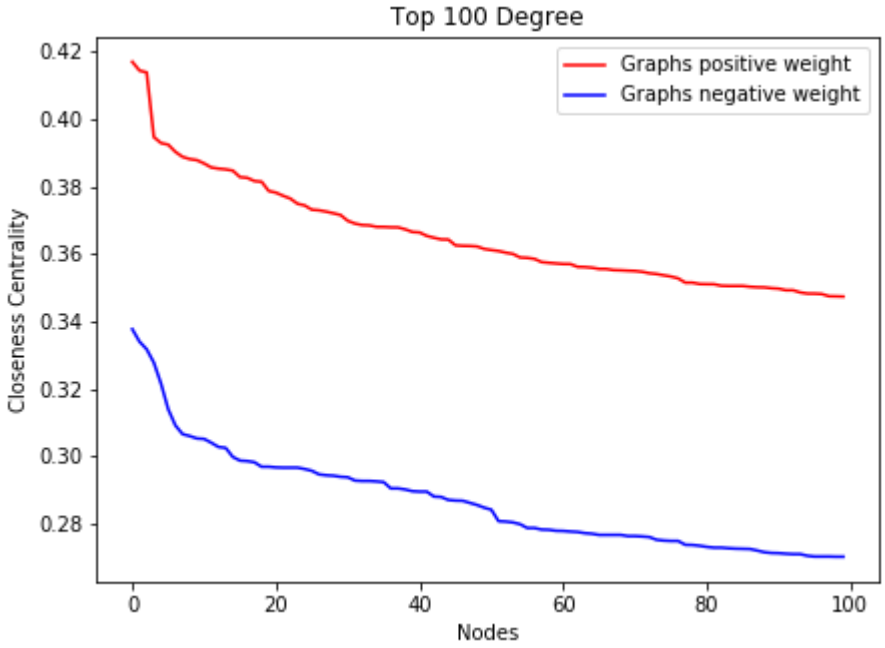


Figura 24

In conclusione, come osservabile in Figura 24, la partizione dei nodi con peso negativo mostra un livello di vicinanza più basso rispetto alla sua controparte.

[i] La cartella di sviluppo può essere trovata qui: https://github.com/Rickyaffo/ARS