

Laboratorio di Web Scraping

RATINGS: scRaping and AnalyzIng biTcolN mininG pools

Progetto di Fine Corso A.A. 2023/24

Lo scopo del progetto è l'implementazione di un insieme di tecniche di deanonimizzazione e di analisi dei **miners** della blockchain di Bitcoin. E' necessario lavorare sia su un DataSet, fornito con il progetto, contenente un sottoinsieme delle transazioni di Bitcoin, che sul sito **WalletExplorer**, per la parte di scraping. Vengono richieste un insieme di analisi generali sulle transazioni contenute nel DataSet di Bitcoin, quindi un insieme di analisi che sfruttano dati ottenuti mediante scraping.

1. Descrizione del DataSet

Viene fornito un DataSet di **Bitcoin** che contiene una selezione delle transazioni incluse nei blocchi compresi tra il blocco Genesis, minato da Satoshi Nakamoto in data **03-01-2009, 17:15:05** e il blocco di altezza **214562**, minato in data **31-12-2012, 11:52:37**. Il DataSet è stato ottenuto tramite una serie di trasformazioni effettuate sui dati pubblici reperiti dalla blockchain di **Bitcoin**, con lo scopo di diminuire la dimensione. In particolare:

- alcuni campi della transazione (versione del protocollo, time lock, ...etc) non sono stati considerati
- gli hash delle transazioni, gli indirizzi contenuti negli output delle transazioni, e gli script sono stati sostituiti con identificatori univoci interi. La corrispondenza tra gli indirizzi della blockchain e gli identificatori univoci del DataSet è stata memorizzata in un ulteriore file di mapping. La corrispondenza tra script e rispettivi identificatori è fornita in Tab. 1.

Il DataSet consiste di **4 files CSV**

- **transactions.csv**, che contiene una riga per ogni transazione del DataSet, con i campi:
 - timestamp**: timestamp del blocco che contiene la transazione. Corrisponde al tempo **UNIX** del miner che ha inserito la transazione nel blocco minato, e indica il momento in cui il blocco è stato minato
 - blockId**: identificatore del blocco che contiene la transazione. Indica l'altezza di tale blocco, ovvero la sua distanza dal blocco genesis di **Bitcoin**
 - txId**: identificatore unico della transazione corrispondente all'hash del contenuto della transazione
 - isCoinbase**: indica se la transazione è una **Coinbase**, ovvero una transazione che trasferisce la ricompensa al miner che ha risolto la **PoW** (0 false, 1 true)
 - fee**: eventuale commissione volontaria contenuta nella transazione, attribuita al miner che la inserisce in un blocco. Può essere zero.

inputs.csv, che contiene una riga per ogni campo di input di ogni transazione del DataSet, con i campi:

- txId**: identificatore della transazione all'interno della quale si trova questo input
 - prevTxId**: identificatore della transazione che ha creato l'output attualmente speso da questo input
 - prevTxpos**: posizione dell'output attualmente speso come input, all'interno della transazione che lo ha creato (diversa da quella che contiene questo input)
- **outputs.csv**, che contiene una riga per ogni campo di output di ogni transazione del DataSet, con i campi:
 - txId**: identificatore della transazione all'interno della quale si trova questo output
 - position**: posizione di questo output all'interno della transazione che lo ha creato

addressId: indirizzo a cui viene inviato questo output, è un identificatore univoco che viene mappato nell'indirizzo reale (hash) tramite il file **mapping.csv**

amount: valore trasferito da questo output

scripttype: codice che identifica lo script contenuto in questo output. Gli script possono essere di diversi tipi (la Tabella 1 mostra i tipi di script definiti da **Bitcoin** e il rispettivo codice contenuto nel DataSet). Tuttavia, dato che il DataSet contiene solo transazioni generate nei primi 4 anni di vita di **Bitcoin**, solo i primi 4 script della tabella sono significativi per questo DataSet. Se lo script è di tipo 0 significa che lo script non è standard e spesso non ha un address associato.

- **mapping.csv**, file di mapping degli indirizzi, campi:

addressId: identificatore unico di ogni indirizzo contenuto in almeno un output delle transazioni del DataSet.

hash: hash corrispondente all'indirizzo. E' l'hash del corrispondente indirizzo contenuto nella blockchain di Bitcoin.

Nel caso di output con script di tipo 0 che non contengono address, nel file di mapping si trova un identificatore univoco rappresentato da una # seguita da un numero che rappresenta quell'output e solo quello, associato con l'identificatore utilizzato per quell'output nel DataSet. .

La struttura del DataSet è mostrata in Fig.1.

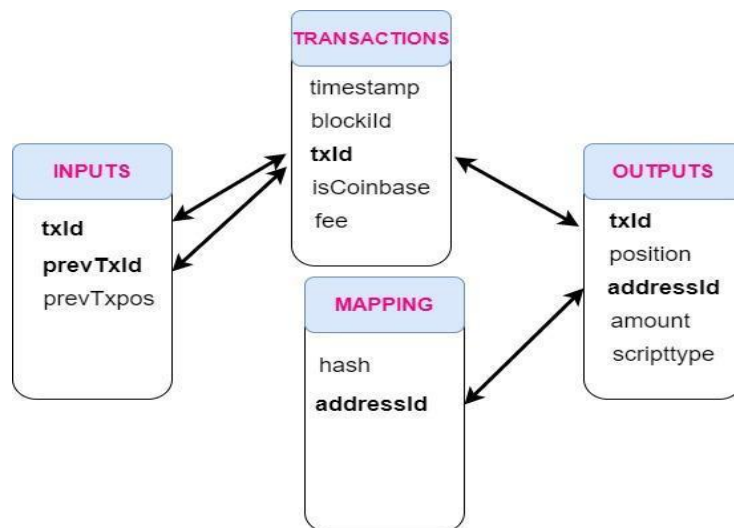


Figura 1: Struttura del DataSet

Script Code	0	1	2	3	4	5	6	7
Script Type	Unknown	P2PK	P2KH	P2SH	RETURN	EMPTY	P2WPKH	P2WSH
Script Size	-	153 bytes	180 bytes	291 bytes	-	-	-	-

Tabella 1: **scripttype** di codifica degli script e le loro dimensioni

Il DataSet è disponibile su Drive al link:

<https://drive.google.com/file/d/1RWP19B0MbfDL43DAEhPwcVkb8nLoIbhX/view?usp=sharing>

2. Analisi generali dei dati della blockchain

E' richiesto di implementare le seguenti analisi sul segmento iniziale di transazioni di Bitcoin contenuto nel DataSet

- studiare l'andamento delle fee contenute nelle transazioni rispetto alla congestione della blockchain, trascurando le transazioni Coinbase, che forniscono le ricompense ai miner e quindi non contengono fee. Lo scopo è verificare se all'aumentare della congestione della chain aumentino anche le fee (come accade anche con il meccanismo del gas sulla blockchain di Ethereum). La congestione della rete in un certo intervallo di tempo viene misurata come somma della dimensione delle transazioni presenti quell'intervallo. La dimensione in byte di una transazione può essere calcolata mediante la seguente formula:

$$size(transaction) = size(input) * n_inputs + size(output) * n_outputs + size(script)$$

ovvero è ottenuta sommando la dimensione media in byte di ogni input moltiplicata per il numero di input della transazione, la dimensione media in byte di ogni output moltiplicata per il numero di output e la dimensione dello script della transazione. La dimensione media di ogni script è mostrata in Tabella 1, mentre le dimensioni medie di ogni input e di ogni output valgono, rispettivamente, 40 bytes e 9 bytes.

- analizzare i tipi di script contenuti nel dataset, evidenziando se e come sia cambiato nel corso dei primi 3 anni della vita di Bitcoin il loro utilizzo.

3. Analisi delle Mining Pool

Bitcoin block explorer with address grouping and wallet labeling

Enter address, txid, firstbits (first address characters), first txid characters, X PUB/Y PUB/Z PUB, internal wallet id, or service name:

ng by X PUB is much improved! Now it supports all X PUB formats, it scans all derivation paths, and all address types, it is much faster and it works even for very large wallets. "Transaction view" for an X PUB is

Top wallets

Exchanges:	Pools:	Services/others:	Gambling:	Old/historic:
Huobi.com (2) Bittrex.com Luno.com Poloniex.com Kraken.com (old) BTC-e.com (output) (old) BitZlato.com Bitstamp.net (old) LocalBitcoins.com (old) MercadoBitcoin.com.br Cryptsy.com Binance.com (old) Bitcoin.de (old) Cex.io	BTCCPool SlushPool.com (old) (old2) GHash.io AntPool.com (old) (old2) Eligius.st Bitfury.com EclipseMC.com (old) (old2) (old3) KnCMiner.com Bitfury.org BW.com Kano.is (old) Telco214	CoinPayments.net Xapo.com Cubits.com Cryptonator.com (old) BitPay.com (old) (old2) (old3) BitcoEX.com HaoBTC.com Cryptopay.me (old) AlphaBayMarket (old) NucleusMarket BitcoinFog BitcoinWallet.com CoinJar.com HolyTransaction.com	SatoshiDice.com (original) LuckyB.it (chatbot) BitZillions.com 999Dice.com CloudBet.com CoinGaming.io PrimeDice.com (old) (old2) (old3) (old4) SatoshiMines.com NitrogenSports.eu SecondsTrade.com PocketDice.io FortuneJack.com Rollin.io BitZino.com	AgoraMarket BitcoinDice.tn SilkRoadMarketplace DeepBit.net SilkRoad2Market EvolutionMarket Instawallet.org UpDown.BT AbraxasMarket MintPal.com SealsWithClubs.eu PandoraOpenMarket MiddleEarthMarketplace BtcDice.com

Fig 2: WalletExplorer: pagina iniziale

In questa parte di RATINGS si dovrà implementare un web scraper il cui scopo è provare a deanonimizzare gli indirizzi contenuti nelle Coinbase contenute nel DataSet, con lo scopo di analizzare il comportamento delle mining pool attive nel periodo considerato.

A questo scopo si consiglia di leggere inizialmente il paper[1] (allegato al progetto). Il paper evidenzia come, nel periodo considerato nel DataSet, fossero attivi solo i seguenti mining pools: **DeepBit**, **Eligius**, **BTC Guild**, **BitMinter**. Ovviamente, durante tale periodo, erano attivi anche utenti singoli che partecipavano al processo di mining, ad esempio lo stesso Satoshi Nakamoto. Anche tali utenti possono aver generato delle Coinbase.

Il processo di deanonimizzazione dovrà utilizzare **WalletExplorer**: si tratta di un servizio che collega un insieme di indirizzi **Bitcoin** a servizi noti (ad esempio piattaforme di exchange, servizi di gambling, marketplaces, etc). In Fig. 2 è mostrata la pagina iniziale di Wallet Explorer (<https://www.walletexplorer.com/>) in cui si può notare come sia possibile immettere un indirizzo presente sulla blockchain di Bitcoin per ottenere, se presente, la specifica del servizio/wallet corrispondente. Tuttavia inserire l'indirizzo di ogni Coinbase richiederebbe un massiccio impiego del processo di scraping, perché potenzialmente potrebbe essere necessario deanonimizzare almeno **214562** indirizzi, uno o più per ogni blocco e quindi per ogni Coinbase presente nel DataSet. Il numero di richieste al server risulterebbe quindi decisamente molto elevato, anche se possibilmente riducibile ad esempio mediante caching degli indirizzi.

Come soluzione alternativa, come mostrato in Fig.3, ricercando ad esempio la mining pool Eligius e cliccando successivamente sul nome della MiningPool, è possibile ottenere l'insieme di tutti gli indirizzi contenuti nel wallet della MiningPool, così come le transazioni effettuate da/verso quegli indirizzi. E' quindi possibile, tramite scraping reperire tutti gli indirizzi associati alle quattro mining pool attive durante il periodo considerato (non è consentito scaricare il file .csv contenente gli indirizzi)

WalletExplorer.com: smart Bitcoin block explorer

Wallet **Eligius.st** ([link to service](#), [show transactions](#))

Page 1 / 11 [Next...](#) [Last](#) (total addresses: 1,033) [Download as CSV](#)

address	balance	incoming txs	last used in block
3P14159f73E4gFr7JterCCQh9QjTjiZrG	0.0021	9	521596
1sA9gdGpKL1FH28DnGdTm2qtWzfnXvAdy	0.00001641	5	769730
18d3HV2bm94UyY4a9DrPfoZ17sXuiDQq2B	0.00000811	10375	598825
1A6eLsi6cmm2xKDSGThd8C2kfzDBXD8YTww	0.00000666	12	770101
1E1igiusfEjs1pCaGjEERExE9gYcrFwow7	0.000006	135	770105
1KDg1KUZ1N3cgcxZ3ghYrETtex7VwhDg5H	0.00000002	3	461183
1CAovbtbRuL3BZ3BhHCGRPdtx5CGU9LVfa	0.00000002	3	460136
1EzLoptmbs3ZDPZDG89bvskPA8Yxuwak58	0.00000001	3	475104
1P6CLUVGYhrKo36XNsf5FaFqAJXGvB6TP2	0.00000001	2	460150
1JkT6EQMotCz6BFqPWFd3kFDQzU1qa119N	0.	690	769730
16kNka7WUg8QAPFy8dJrv7UUSu2FAG2pkW	0.	461	426601
1362ww2sMAagXXmFrif5s7vsA8ymCGEBua	0.	444	426601
14cbTxT4nN1AFQzEdFURb9co7TBA7MYqaA	0.	320	426601
1Dpd5GEv2S64E4q1q3YgQFKUfMPsBkAzvv	0.	298	426601
1QATWksNFGUJCWBRN4g6hGM178Lovm7Wh	0.	221	754534
16g5S427XsTG8torPRb7r3PXRh1zcuYCo	0.	216	426601
1HrqGugizGWdfZgZJy2BehkCdLcCBdhMnV	0.	180	769730
1Di1z5MYHuD3rZqdg1hPDWcUt7a8xqghJb	0.	174	409652
1CdcYVP4T4hjHwt353pEnGHrgeDLvuvZL	0.	165	265452
1GEJfZRPk2BL5Sx3r6gwtuFxCUvq3OytN	0.	124	217440
1NRzE2C8yZx4zgX2PK3C6a5E5PqWhWgiF3	0.	77	426875
15hob4bTJct25JwM5HF35abHyE7pY6rvme	0.	72	426628
17HorBQ3H8H9SSNCEqD5bhtupkLZnNhGL	0.	66	426601
1RNUbH2wo2PmrEQiuX5ascLEXmtcFpool	0.	63	213516
134dV6U7gQ6wCFbfHUz2CMh6Dth72oGpgH	0.	53	754532
164a8VU1G69VWwCqLhhuA3au6Vw468mkbD7	0.	49	470761

Fig. 3 Indirizzi associati ad un wallet

Si richiede quindi di:

- reperire, mediante scraping, tutti gli indirizzi associati alle 4 mining pool considerate ed utilizzare gli indirizzi scaricati per deanonimizzare gli indirizzi utilizzati nelle Coinbase presenti nel DataSet. Per quanto riguarda le Coinbase che presentano indirizzi non appartenenti a nessuna delle 4 mining pool, provare a deanonimizzare tramite WalletExplorer i 4 miners che hanno prodotto più transazioni Coinbase (riferiti come top 4 miners), e raggruppare tutti gli altri in una categoria "Others"
- analizzare le Coinbase deanonimizzate e produrre le seguenti statistiche:
 - numero di blocchi minati da ciascuna delle 4 mining pool, sia globalmente, che mostrando l'andamento temporale dei blocchi minati, per intervalli temporali di due mesi (ed eventualmente quelli dei top 4 miners) ;
 - distribuzione delle reward totali ricevute da ogni mining pool, sia globalmente che mostrandone l'andamento temporale, sempre per intervalli di due mesi;
- considerare infine la Coinbase di Eligius mostrata in Fig.4. Questa transazione può essere reperita semplicemente digitando il suo hash nell'explorer. Come si può vedere in figura è possibile individuare la transazione successiva che spende i bitcoin di questa Coinbase seguendo la freccia in basso a destra in figura. Ripetendo il procedimento ricorsivamente più volte è possibile "seguire il flusso" dei bitcoin (una tecnica utilizzata in una tecnica di analisi chiamata taint analysis). Si chiede di tracciare il percorso dei bitcoin creati e di creare, mediante NetworkX, un grafo che descriva tale percorso. Si considerino al massimo k passi di tale percorso.



Fig. 4: Una Coinbase di Eligius

Per ogni analisi richiesta, scegliere adeguatamente il meccanismo di plotting di **Matplotlib** e di **Seaborn** più adeguato.

4. Modalità di svolgimento e di consegna del progetto

Il progetto deve essere eseguito individualmente.

E' possibile scaricare il DataSet di riferimento da Drive, link:

<https://drive.google.com/file/d/1RWP19B0MbFDL43DAEhPwcVkb8nLoIbhX/view?usp=sharing>

Il riferimento è a **Google Drive** fornito da Unipi, per cui l'accesso dovrebbe essere consentito con credenziali Unipi. In caso di difficoltà nell'accesso, inviare una mail a laura.ricci@unipi.it.

Il materiale da consegnare comprende:

- codice dell'applicazione (Notebook **.ipynb** e/o script **Python**) e una breve relazione, contenuta nel Notebook **.ipynb**. Si prega di sottomettere tutto il materiale in formato **.pdf**.
- il codice deve essere sviluppato in **Python 3.0** e per la parte di scraping si devono utilizzare le librerie **BeautifulSoup** e (eventualmente) **Selenium**

Relazione e codice sorgente devono essere consegnati su Moodle in un unico archivio compresso in formato zip. Nel caso le dimensioni del DataSet si rivelassero troppo elevate per le risorse computazionali che lo studente ha a propria disposizione, potete mandate una mail a laura.ricci@unipi.it, per ricevere un DataSet ulteriormente ridotto.

Riferimenti

[1] *The evolution of mining pools and miners' behaviors in the Bitcoin blockchain*, Natkamon Tovanich, Nicolas Soulié, Nicolas Heulot, Petra Isenberg, HAL Id: hal-03610424